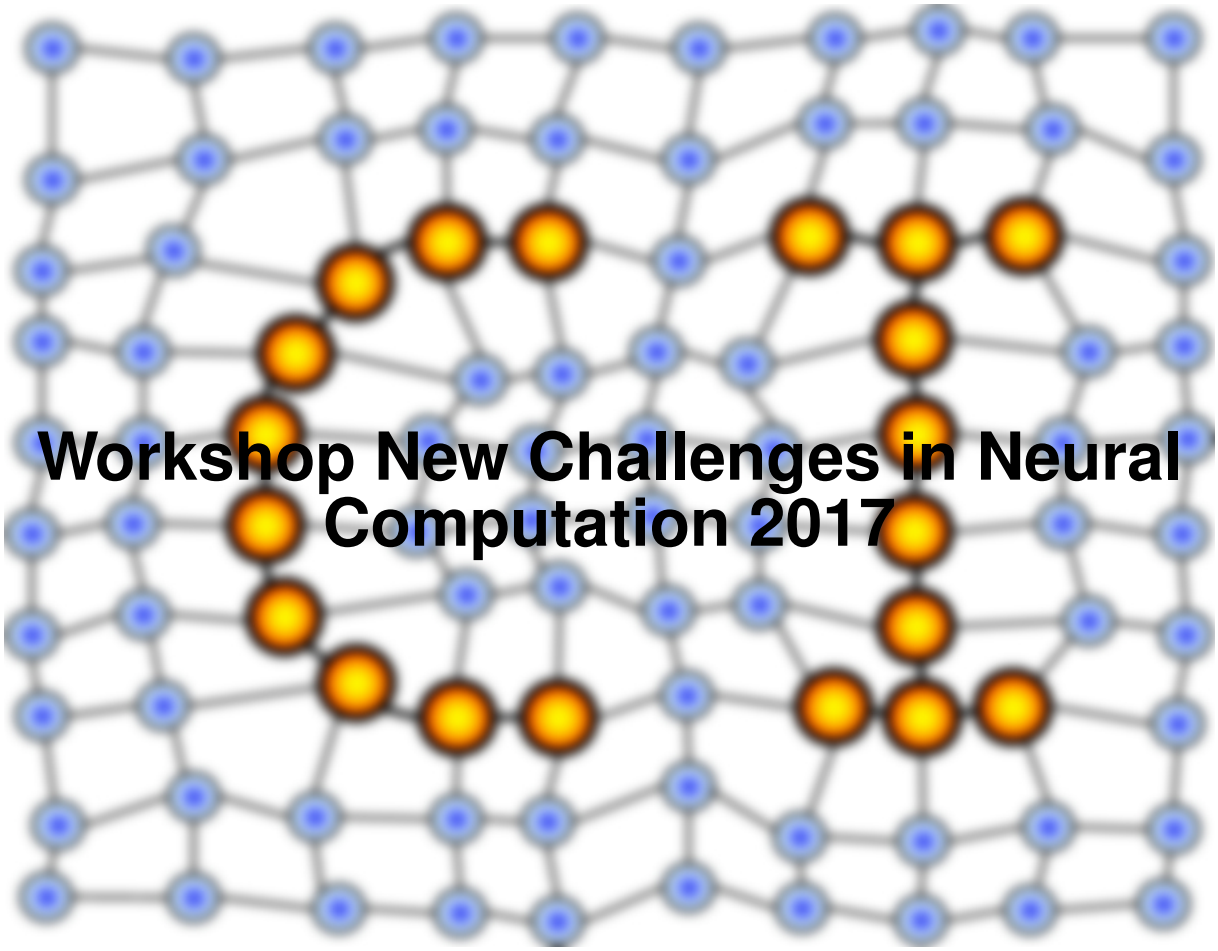
MACHINE LEARNING REPORTS



Workshop New Challenges in Neural Computation 2017

Report 03/2017
Submitted: 06.09.2017
Published: 12.09.2017

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

Contents

Keynote talk: Towards end-to-end learning and optimization

Frank Hutter, University of Freiburg, Germany

Abstract:
Deep learning has recently helped AI systems to achieve human-level performance in several domains, including speech recognition, object classification, and playing several types of games. The major benefit of deep learning is that it enables end-to-end learning of representations of the data on several levels of abstraction. However, the overall network architecture and the learning algorithms' sensitive hyperparameters still need to be set manually by human experts. In this talk, I will discuss extensions of Bayesian optimization for handling this problem effectively, thereby paving the way to fully automated end-to-end learning. I will focus on speeding up Bayesian optimization by reasoning over data subsets and initial learning curves, sometimes resulting in 100-fold speedups in finding good hyperparameter settings. I will also show competition-winning practical systems for automated machine learning (AutoML) and briefly show related applications to the end-to-end optimization of algorithms for solving hard combinatorial problems.

Keynote talk: Hierarchical Sensing

Thomas Martinetz, University of Lübeck, Germany

Abstract:
Natural signals like images often have an inherently sparse structure. Compressive Sensing exploits this structure and is able to measure such signals with only a very few measurements. However, it requires to solve a complex optimization problem. In this talk I present a measurement scheme which is adaptive and hierarchical. This scheme obtains the signal coefficients directly with exactly the same number of measurements as Compressive Sensing, but without having to solve an optimization problem. It comes out that it starts to sample images coarsely and then goes deeper into those regions which carry information.

Restricted Tangent Metrics for Robust Learning Vector Quantization of Data with Local Drifts

T. Villmann[1] and S. Saralajew[2]

[1] Computational Intelligence Group, Univ. Applied Sciences Mittweida, DE

[2] Dr. Ing. h.c. F. Porsche AG Weissach, Germany

Abstract. In this contribution we consider restricted tangent metrics for local approximations of prototype manifolds to deal with variations and transformations in data like drifts or rotations as frequently observed also in transfer learning. .

Learning Vector Quantization (LVQ,[1]) for classification learning is based on the idea of class distribution representing prototypes. These prototypes usually are seen as vectors in a vector space with a given dissimilarity measure, frequently the Euclidean distance. Advanced variants of LVQ like generalized LVQ (GLVQ,[2]) or robust soft LVQ (RSLVQ,[3]) use a stochastic gradient scheme to optimize a cost function reflecting the classification error. Thereby, the differentiability of the dissimilarity measure is a necessary assumption. In general, these prototype-based classifiers are known to be noise tolerant.

Difficulties arise if systematic variations like drifts, shifts or rotations occur [4]. An attempt to overcome those problems is the application of data dissimilarities being invariant regarding specific transformation [5]. One of the most popular metrics is the tangent metric [6].

Recently, the GLVQ concept was extended to deal with variations and transformations in the data. According to this approach the prototypes are now references for affine sub-spaces of the data space [7,8]. More precisely, the prototypes constitute affine approximations of a continuous prototype manifold, which model the manifold [9]. This approximation is obtained by a Taylor expansion of the manifold model whereby the respective dissimilarity measure is the previously mentioned tangent metric. Particularly, the so-called one-sided tangent metric proposed in [10] remains differentiable and, therefore, is well suited for the application in gradient based learning schemes.

Yet, the Taylor approximation considered so far is taken as a global approximation, which violates the mathematical assumption of local validity of Taylor expansions. Therefore, we now restrict the affine subspaces to be only local patches valid only in local regions, i.e. we consider a local representation of data variability. The resulting adaptive *restricted tangent metric* leads to a minimization problem, which can be solved analytically [11]. Further, this local tangent metric is still differentiable such that it can be immediately plugged into gradient based models like GLVQ or RSLVQ.

During the workshop we will demonstrate the working principles of the method and illustrate them for two toy examples. Further, the close relation of this approach to transfer learning strategies for GLVQ will be explained. Finally, open questions like numerical stability, robustness and complexity are addressed.

References

[1] Teuvo Kohonen. Learning Vector Quantization. *Neural Networks*, 1(Supplement 1):303, 1988.

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

[3] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.

[4] C. Prahm, B. Paassen, A. Schulz, B. Hammer, and O. Aszmann. Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In J. Ibanez, J. Gonzales-Vargas, J.M. Azorin, M.Akay, and J.L. Pons, editors, *Proceedings of the 3rd International Conference on NeuroRehabilitation (ICNR2016)*, volume 15 of *Biosystems and Biorobotics*, pages 153–157. Springer, 2016.

[5] D. Keysers, W. Macherey, H. Ney, and J. Dahmen. Adaptation in statistical pattern recognition using tangent vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):269–274, 2004.

[6] P. Simard, Y. LeCun, and J.S. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 50–58. Morgan-Kaufmann, 1993.

[7] T. Hastie, P. Simard, and E. Säckinger. Learning prototype models for tangent distance. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 999–1006. MIT Press, 1995.

[8] S. Saralajew and T. Villmann. Adaptive tangent metrics in generalized learning vector quantization for transformation and distortion invariant classification learning. In *Proceedings of the International Joint Conference on Neural networks (IJCNN) , Vancover*, pages 2672–2679. IEEE Computer Society Press, 2016.

[9] S. Saralajew and T. Villmann. Transfer learning in classification based on manifold models and its relation to tangent metric learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) , Anchorage*, pages 1756–1765. IEEE Computer Society Press, 2017.

[10] D. Keysers, J. Dahmen, T. Theiner, and H. Ney. Experiments with an extended tangent distance. In A. Sanfeliu, J. J. Villanueva, M. Vanrell, R. Alquékzar, J. Crowley, and Y. Shirai, editors, *Proceedings of the 15th International Conference on Pattern Recognition, Barcelona*, volume 2, pages 38–42. IEEE Press, Los Alamitos, California, 2000.

[11] S. Saralajew and T. Villmann. Restricted tangent metrics for local data dissimilarities - mathematical treatment of the corresponding constrained optimization problem. *Machine Learning Reports*, 11(MLR-01-2017):submitted, 2017. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2017.pdf.

Condition Invariant Visual Localization Using Slow Feature Analysis

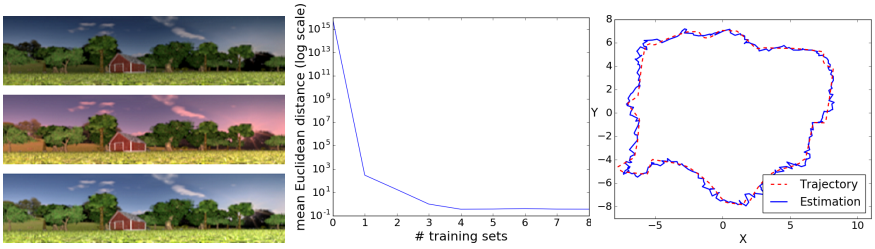
M. Haris[1], B. Metka[1], M. Franzius[2], and U. Bauer-Wersing[1]

[1] Frankfurt University of Applied Sciences, Frankfurt, Germany
[2] Honda Research Institute Europe GmbH, Offenbach, Germany

Abstract. In outdoor scenarios varying environmental conditions like seasonal, weather and lighting effects have a strong impact on the appearance which often prevents successful localization. A spatial representation of the environment can be learned by applying unsupervised Slow Feature Analysis (SFA) directly to images captured by a mobile robot. However, effects that change on a slower or equal timescale than the robot's position during learning will be encoded in the resulting representations and thus affect spatial coding. In this work we use recordings from a simulator along the same trajectory, each in a different condition, which allows to change the perceived image statistics for improved condition invariance. Experiments demonstrate an improvement of spatial coding even for few training sets.

Despite the impressive advancements in visual localization and mapping [1] methods, outdoor long-term localization in changing environments is still a challenging problem that is approached using condition invariant image descriptors or learning-based appearance change prediction [2]. In this work, we approach long-term robustness using the unsupervised learning capabilities of SFA [3] which extracts slowly varying or invariant features from quickly varying input signals. Applied to a temporal sequence of images it encodes high level information, like the position of objects, which is embedded in the image data and changes slowly compared to the raw pixel values. A hierarchical SFA-model trained with views from a virtual rat can mimic place or head direction cells [4], which represent spatial information in the brain of rodents, depending on the movement statistics during training. To learn orientation invariant representations of the position the model was extended using an omnidirectional mirror to increase the amount of perceived rotational movement [5]. In [6], loop closures in the trajectory are used to re-insert images in the training sequence to learn an invariance w.r.t. slowly varying environmental changes during training. Here, we extend this approach to long-term recordings from the same trajectory. Using a simulator, we can easily determine position correspondences between different recordings to create a training sequence where we add images from past conditions for the successive places along the trajectory.

Experiments are conducted in a simulated outdoor environment covering an area of 15×15 meter. We capture 10 image sets along the same trajectory, each consisting of 279 panoramic images of size 600×60 pixels. For every set, a change



(a) **(b)** **(c)**

Fig. 1: (a) The same place in different conditions. (b) Results for an increasing no. of training sets. (c) Estimated trajectory using 9 training sets.

of the environmental condition is simulated by a random variation of lighting parameters (see Fig.1a). The parameters include energy, the y-coordinate of light source and the intensity of the red channel. Based on position correspondences, we reorder the training sequence such that the environmental condition varies faster than the position of the robot. The model is trained with an increasing number of up to 9 data sets and the performance is tested on the successive set by computing a regression function from the SFA-outputs to ground truth positions (x, y). We repeated the same procedure with 10 random permutations of the image sets. The mean localization performance of the experiments is given in Fig.1b. Using the representations learned in a single condition has a prohibitively large error in different conditions but it decreases quickly for more data sets and amounts to a mean Euclidean distance of 0.35 meter using 9 sets. The ground truth trajectory and an estimated trajectory is shown in Fig.1c. We have shown that an agent can autonomously learn an increasingly invariant representation of the environment. To test it in a real world scenario, we are currently recording data. Further, we plan to combine it with the simulated rotation in order to provide orientation invariance as well. It would be interesting to investigate if the learned condition invariant features of the lower layers generalize to a completely different environment.

References

1. Mur-Artal, R., Montiel, J.M.M., Tards, J.D.: ORB-SLAM: A versatile and accurate monocular slam system. IEEE Transactions on Robotics 31(5), 1147–1163 (2015)
2. Lowry, S., Snderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P., Milford, M.J.: Visual place recognition: A survey. IEEE Transactions on Robotics 32(1), 1–19 (2016)
3. Wiskott, L., Sejnowski, T.: Slow feature analysis: Unsupervised learning of invariances. Neural Computation 14(4), 715–770 (2002)
4. Franzius, M., Sprekeler, H., Wiskott, L.: Slowness and sparseness lead to place, head-direction, and spatial-view cells. PLoS Computational Biology 3(8), 1–18 (2007)
5. Metka, B., Franzius, M., Bauer-Wersing, U.: Outdoor self-localization of a mobile robot using slow feature analysis. In: ICONIP. pp. 249–256 (2013)
6. Metka, B., Franzius, M., Bauer-Wersing, U.: Improving robustness of slow feature analysis based localization using loop closure events. In: ICANN. pp. 489–496 (2016)

Unsupervised Transfer Learning for Time Series via Self-Predictive Modelling - First Results

Witali Aswolinskiy and Barbara Hammer

Bielefeld University, Germany

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

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

1 Introduction

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

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

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

data is minimal. Still, this solution requires sufficient supervised information to find the transformation. Here, we attempt to use only the temporal structure of the source and target data to find such a transformation.

More precisely, we propose to build a self-predictive reservoir model to capture the spatio-temporal relationships in the source data. We then use this model as a surrogate for the actual learning task to find a transformation from target to source space. In the following, we will formalize this methodology and present examples, where it enables a transfer without any given labeling. We will also showcase an example, where a transfer failed due to a limited correlation of the temporal dynamics and the supervised transfer learning task.

2 Unsupervised Transfer Learning via Self-Predictive Modeling

2.1 A hypothetical example

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

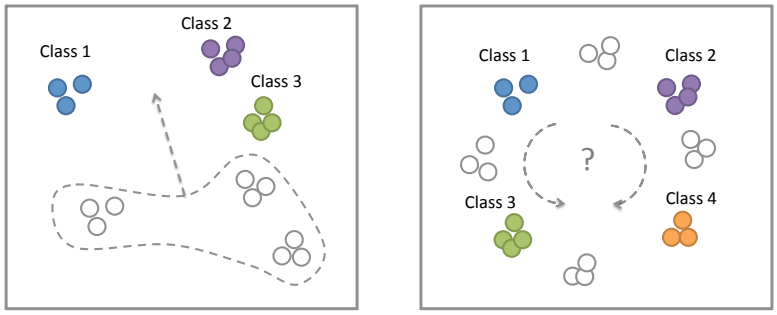


Fig. 1: Hypothetical classification examples. The filled circles represent the labeled source data and the empty circles the unlabeled target data. In the case on the left, a translation will map the target data onto the source data. In the case on the right, there are several possible rotations to do so.

The example on the right visualizes a case, where the structure of the data does not provide sufficient information to transform the target data back so

that the classes can be correctly estimated. Because of the quadratic structure, four different rotations are possible, but only one of them will map the classes correctly.

As these examples show, in order to find the correct transformation from the target space back to the source space, the data must have a very distinctive structure and the classes must also have structurally distinctive properties. Next, we present a general framework for domain adaption using the structure of the data.

2.2 Unsupervised transfer of structured data via self-predictive models

In Fig. 2 we sketch a framework for unsupervised transfer via structural self-prediction. Additionally to the supervised learner (regressor, classifier, etc.), we train a self-predictive model on the source data using its structure to define a training goal, e.g. to predict the next steps in time series or nearby pixels in images. No external information is used to train the predictive model. Then, we try to find a transformation from the target to the source space, so that the error of the predictive model applied to the transformed target data is minimal. A small error of a precise predictive model should indicate a good mapping. After finding such a mapping, we can transfer the target data into the source space and apply the source-trained learner to it.

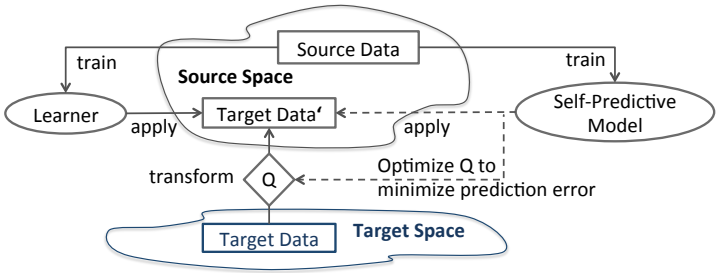


Fig. 2: Framework for unsupervised transfer via self-predictive modeling. The transformation Q is learned by minimizing the prediction error of the self-predictive source model on the transformed target data.

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

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

Given time series $\boldsymbol{u}_\mathcal{S}$ and $\boldsymbol{u}_\mathcal{T}$ from source space $\mathcal{S}$ and target space $\mathcal{T}$, respectively, we want to find a linear transformation $Q : \mathcal{T} \to \mathcal{S}$ such that the temporal dynamics of the transformed target data match those of the source domain. The approach is visualized in Fig. 3 and has two phases: learning a self-predictive model on the source time series and learning the linear transformation from source to target space.

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

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

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

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

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

Learning the linear transfer function Having determined the one-step-ahead-prediction dynamics P for the source domain, we now learn the linear transfer function $Q(\boldsymbol{u}_\mathcal{T}) = \boldsymbol{Q}\boldsymbol{u}_\mathcal{T} = \boldsymbol{u}'_\mathcal{S}$ on the target data, such that the source dynamics apply: $P(\boldsymbol{Q}\boldsymbol{u}_\mathcal{T}(k))\boldsymbol{Q}^{-1} \approx \boldsymbol{u}_\mathcal{T}(k+1)$. For this purpose, we evolve the linear transformation matrix $\boldsymbol{Q}$ with the CMA-ES [3] optimization technique by minimizing the mean squared error:

$$\boldsymbol{Q} = \arg\min_{\boldsymbol{Q}}(\|P(\boldsymbol{Q}\boldsymbol{u}_\mathcal{T}(k))\boldsymbol{Q}^{-1} - \boldsymbol{u}_\mathcal{T}(k+1)\|^2) \tag{3}$$

An alternative formulation avoiding matrix inversion is to minimize $\|P(\boldsymbol{Q}\boldsymbol{u}_\mathcal{T}(k)) - \boldsymbol{u}_\mathcal{T}(k+1)\boldsymbol{Q}\|^2$. However, this has the degenerate solution $\boldsymbol{Q}=0$.

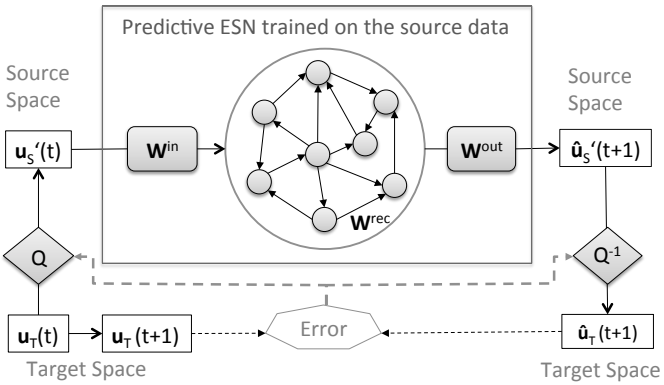


Fig. 3: Approach for unsupervised transfer of the target data through the linear transformation Q by using a self-predictive ESN trained on the source data.

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

3 Experiments

For our experiments, we use ESNs not only for the self-predictive model, but also for the actual learning task. Since any other learning method suitable for time series would work as well, we omit the description of the learner training.

3.1 Sine wave regression

As first example we consider a synthetic, two-dimensional dataset consisting of two sine waves $u_1 = sin(0.1x), u_2 = sin(0.25x)$ with the learning target $y = u_1(t-2) + u_2(t+2)$. Fig. 4 shows the original data on the left and the same data after a random linear transformation on the right (regression goal y remains the same).

For both the learner and the self-predictive model, a reservoir with 50 neurons was used. Fig. 5 shows the prediction error and the transfer error (the regression error of the learner on the transformed target data) during evolution of the transformation matrix. After approximately 80 iterations of CMA-ES, the target data is successfully mapped back into the source space.

3.2 Time series classification - success

Here, we apply the approach to time series classification of multivariate time series. The character trajectories dataset [11] obtained from [6] contains pen tip trajectories (x,y,force) with lengths from 109 to 205 recorded during writing

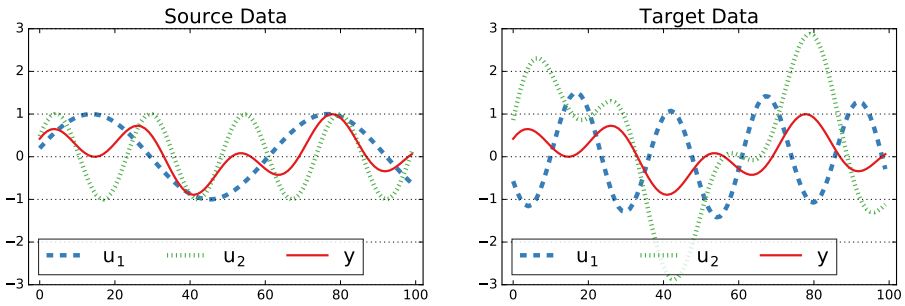


Fig. 4: Synthetic sine regression dataset with source data on the left and target data on the right.

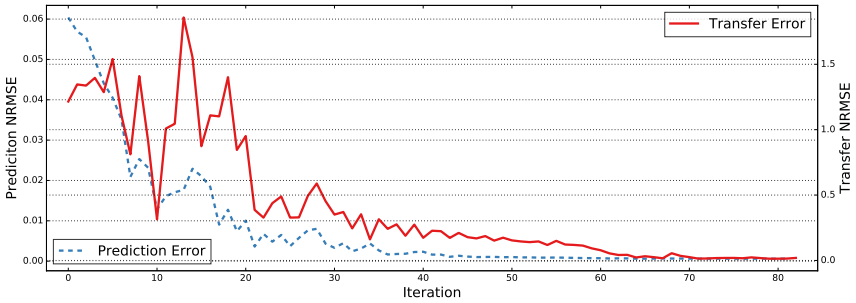


Fig. 5: Progress during optimization of the transformation matrix. Shown is the prediction and the regression error on the target data after transformation.

of twenty characters by a single subject. 300 sequences are used for training (source data) and 500 for testing (target data). Again, we simulate a systematic change in the data by transforming the test sequences through multiplication with a random matrix. We then try to discover the transformation back into the source space using a self-predictive ESN with 300 neurons trained on the original training sequences.

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

3.3 Time series classification - failure

The uWave [7] dataset contains three-dimensional (x,y,z) sequences of length 315 containing eight different gestures recorded from eight subjects. 200 sequences were used for training and 500 for testing. Fig. 6 (bottom) shows the evolution of the prediction and target classification error. Here, the transformation reduces the prediction error, but increases the classification error - the approach did not work for this dataset.

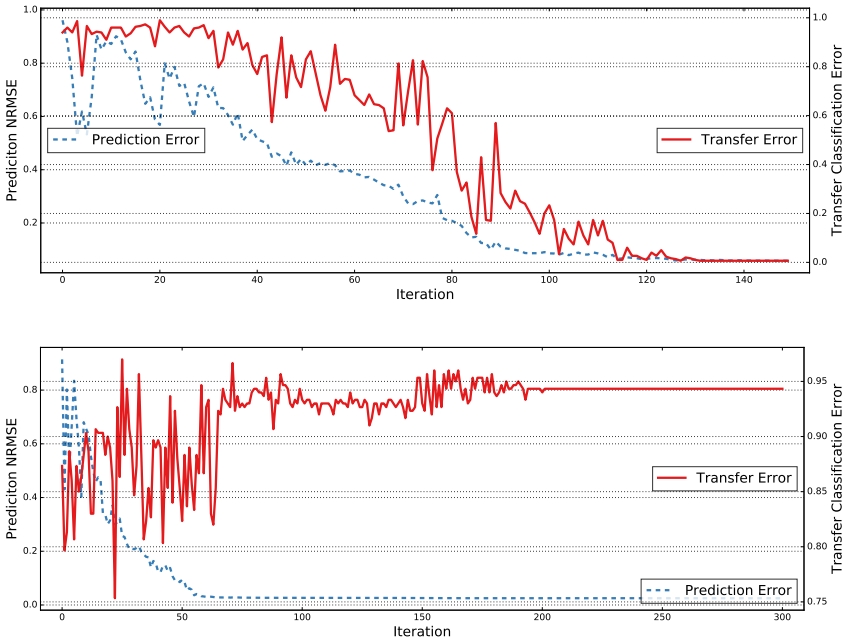


Fig. 6: Evolution of the optimization of the transformation matrix for classification of the character trajectories (top) and uWave dataset (bottom). Shown is the prediction and the classification error on the target test data after transformation.

The hypothetical example in 2.1 showed that a transformation from the target to source space, which aligns the data distributions in the respective spaces, may still be wrong semantically. The failed transfer of the uWave target data may be an example of this problem. Despite a very small prediction error, the classification results were wrong. We hypothesize that the dynamical invariant in the time series is not relevant for the classification in this case, hence a transformation based on the preservation of the temporal dynamics cannot be used as a surrogate for the learner.

Two other possible reasons for a failed unsupervised transfer are:

- Inaccurate predictive model. If the predictive model has a high prediction error on the source data (due to noisy data, badly chosen basis functions, etc.), there will be many transformations resulting in similar prediction errors on the transformed target data. Such an inexact mapping might lead to misclassifications by the learner.
- Local error minima in the transformation space. Let us again consider the triangle-data in the hypothetical example visualized in Fig. 1. The linear

transformation from the the target to the source space is a simple translation. However, a translation together with half a rotation would result in only a slightly higher prediction, but a complete classification failure. Intermediate rotation angles might lead to better classification, but would have higher prediction errors. Thus, finding a global optimum might be more important here than in other applications.

4 Conclusion

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

References

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

Differential Privacy for Learning Vector Quantization

J. Brinkrolf, K. Berger, and B. Hammer

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

Abstract. Digital information is collected daily in growing volumes. Mutual benefits drive the demand for the exchange and publication of data among parties. However, it is often unclear how to handle these data properly in the case that the data contains sensitive information. Simple anonymization of the data, for example, does not ensure privacy since the information can easily be linked to information which is freely available on the web and which might reveal the true identity of the involved persons [12]. Differential privacy has become a powerful principle for privacy-preserving data analysis tasks in the last few years, since it entails a formal privacy guarantee for such settings. This is obtained by a separation of the utility of the database and the risk of an individual to lose her privacy. In this contribution, we briefly review the approach of statistical disclosure control which is offered by differential privacy. We introduce the Laplace mechanism and a stochastic gradient descent methodology which guarantee differential privacy [1]. Then, we show how these paradigms can be incorporated into a popular machine learning algorithm, namely prototype-based classification trained by learning vector quantization (LVQ). We demonstrate the results of privacy preserving LVQ based on a popular benchmark example.

Keywords: privacy preserving data analysis, differential privacy, learning vector quantization

1 Introduction

The necessity to preserve a person's privacy in data bases and according requirements for the privacy and security of this technology has been debated since more than twenty years [4]. While encryption can secure data bases whenever private information is revealed to only the user herself, the setting becomes more problematic whenever important information of the data base is offered to the public. This is the case if summary information or trends which have been inferred from the data base are offered to the public, and it constitutes a key challenge if personal data are used to train a machine learning model, which is later rolled out to the public. While summary statistics or machine learning models deliver accumulated information and general models, it cannot be ruled

out a priori that private information can be inferred from those, provided the model is coupled with according auxiliary data.

In this context, the notion of *differential privacy (DP)* has been proposed as a formalism which provably limits the possibility to retrieve private information from published models [7]. Basically, DP formalizes the intuition that the amount of individual information which can be retrieved from such models is strictly limited per query. This way, it can formally guarantee essential properties such as immunity of the formalism to auxiliary information and privacy of individual information as well as specific groups. Additionally, it possesses convenient mathematical properties such as understandable behavior under composition of DP mechanisms and closure under post-processing; as a consequence, DP algorithms can be designed based on essential building blocks, and complex programs can be designed this way. Quite a few general formalisms of DP have been proposed, including, e.g., differential privacy for general purpose optimization mechanisms such as genetic algorithms or gradient descent [5,11,15,1]. In this contribution, we will rely on a DP formalism which robustifies gradient descent.

Learning vector quantization (LVQ) constitutes a very popular and intuitive machine learning technology, which represents data in terms of prototypical representatives. This enables its intuitive interpretation as well as its integration into life-long learning scenarios [3]. Its model form, however, carries a high risk of revealing sensitive information since prototypes display typical feature values which directly stem from training data. Since LVQ constitutes a very popular model in the context of highly sensitive domains such as biomedical applications [2], there is a strong need of differential privacy in this domain.

In this contribution, we propose an adaptation of LVQ to a provably private version based on the notion of differential privacy. For this purpose, we rely on a popular variant of LVQ which phrases learning as cost optimization [13]. This enables us to combine the method with a differentially private stochastic gradient descent [1]. We demonstrate the efficiency and effectiveness of the method in experiments.

2 Background

In the following, we briefly introduce generalized learning vector quantization (GLVQ) as one popular and intuitive class of machine learning models for classification [13]. Since it constitutes a prototype based classification mechanism which is based on prototypical representatives within the vector space of input signals, it runs the risk of revealing sensitive information about data which have been used for training. Hence we aim for a variant which guarantees differential privacy (DP), as we will introduce later. For this purpose, we shortly recapitulate the notion of differential privacy as well as a few popular DP strategies.

2.1 GLVQ

We are interested in classification scenarios in $\mathbb{R}^d$ with k classes which are enumerated as $\{1, \ldots, k\}$. Prototype-based classifiers are defined as follows: a set

W of w prototypes with $(\mathbf{w}_j, c(\mathbf{w}_j)) \in \mathbb{R}^d \times \{1, \ldots, k\},\ j \in \{1, \ldots, w\}$ is specified which enables a good classification and representation of the data and its underlying classes. A new data point $\mathbf{x}$ is classified by the winner takes all scheme:

$$\mathbf{x} \mapsto c(\mathbf{x}) := c(\mathbf{w}_l) \quad \text{with } l = \arg\min_{\mathbf{w}_j \in W} d(\mathbf{w}_j, \mathbf{x}),$$

where d is a distance measure, e.g., the standard Euclidean distance.

Here, we will focus on the generalized LVQ by Sato and Yamada [13] which is derived from an explicit cost function. The LVQ rule is phrased as cost minimization with the cost function

$$E = \sum_i \Phi \left(\frac{d^+(\mathbf{x}_i) - d^-(\mathbf{x}_i)}{d^+(\mathbf{x}_i) + d^-(\mathbf{x}_i)} \right).$$

The function Φ is a monotonic increasing one, e.g., the logistic function. $d^+(\mathbf{x}_i)$ is the distance of $\mathbf{x}_i$ to the closest prototype of the correct class and $d^-(\mathbf{x}_i)$ is the closest distance of $\mathbf{x}_i$ to another prototype of a different class than $\mathbf{x}_i$. Standard GLVQ uses the squared Euclidean metric $d(\mathbf{w}_j, \mathbf{x}) = (\mathbf{x} - \mathbf{w}_j)^T (\mathbf{x} - \mathbf{w}_j)$.

2.2 Differential Privacy

Differential privacy [7,6,8] constitutes a strong standard for privacy guarantees for algorithms on aggregate databases. Informally, it requires that the output of a data analysis mechanism remains approximately the same if any data point in the input database is added or removed. This guarantees that a single entry cannot substantially affect the revealed outcome, hence it is impossible to retrieve sensitive individual information from the latter. Now, we define differential privacy first and introduce specific differential private mechanisms later.

Definition 1 (Differential Privacy [7]). *Assume ε, $\delta > 0$ are given. We are interested in the privacy of an operation $\mathcal{A}$ such as a machine learning algorithm, which maps a given set of training data D to a model or summary statistics revealed to the user. These outcomes might be subject to manipulation or attacks, which are unknown. To take this into account, the space of possible models is modeled as a probability space where measurable events can take place. A randomized function $\mathcal{A}$ gives* **(ε, δ)-differential privacy** *iff for all pairs of adjacent datasets D and D', and all events S*

$$\mathrm{P}[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \cdot \mathrm{P}[\mathcal{A}(D') \in S] + \delta.$$

Here P *refers to the probability induced by the algorithm $\mathcal{A}$. Thereby, two datasets D and D' are* **adjacent** *if and only if D can be obtained from D' by the deletion of one training example (or vice versa).*

This notion of differential privacy ensures the privacy of any single data point which can be used for training, because adding or removing any single data point

results in e^{ε}-multiplicative-bounded changes in the probability distribution of the output of the algorithm only.

Differential privacy is compositional in the sense that combining m multiple mechanisms $\mathcal{A}$ that satisfy differential privacy for $\varepsilon_1, \ldots, \varepsilon_m$ results in a mechanism that satisfies ε-differential privacy for $\varepsilon = \sum_i \varepsilon_i$. We will call ε the privacy loss of the algorithm.

2.3 Differentially Private Mechanisms

There are several approaches which satisfy ε-differential privacy, including the *Laplace Mechanism* [7]. The latter deals with algorithms or functions $f : \mathcal{D} \mapsto \mathbb{R}^k$ from the domain of all datasets to vectorial outputs. It adds symmetric and scaled noise to each dimension of the output. The magnitude of the required noise depends on the so-called *sensitivity* of f. The latter refers to the maximum difference of the outputs of two adjacent datasets. Formal, the sensitivity of f is defined as

$$\Delta f = \max_{\text{adj } D,\, D'} \parallel f(D) - f(D') \parallel_1$$

measured in the L_1 norm. Given a function f the Laplace mechanism is defined as

$$\mathcal{A}_f(D) = f(D) + (Y_1, \ldots, Y_k)^T$$

for a given database D, where Y_i are i.i.d. random variables drawn from the Laplace distribution $\text{Lap}\left(\frac{\Delta f}{\varepsilon}\right)$. The latter is defined by the probability density function $\text{P}[\text{Lap}(\beta) = x] = \frac{1}{2\beta} e^{-|x|/\beta}$. It can be shown that the resulting mechanism $\mathcal{A}_f$ is $(\varepsilon, 0)$-differential private. The Laplace mechanism constitutes a very convenient way to turn a given database query into a differentially private one. However, it has only limited applicability if f is given by a learning algorithm, since the sensitivity of the latter might be complicated to bound. Therefore, more direct methods which directly rely on typical machine learning mechanisms have been proposed. A very popular one adds differential privacy to gradient techniques.

2.4 Differential Private Scaled Gradient Descent

GLVQ, as introduced above, is often trained by means of a gradient descent mechanism. Hence, we can make use of the DP mechanism as introduced by Abadi et al. [1]. Essentially, the mechanism proposes a variant which makes the operations as common for a scaled gradient method private. In the following, we just give an outline of the algorithm as proposed by Abadi et al. and we refer to the original paper for more details and proofs.

Assume an objective function $\mathcal{L}(\theta)$ with parameters θ is given which is optimized to reveal the model parameters θ. The idea of the proposed formalism is to compute the gradient $\nabla_\theta \mathcal{L}(\theta, \mathbf{x}_i)$ of the loss function for each data point which is taken from a random subset with size L of the training set. Then, the

gradients are clipped whenever its L_2 is greater than a threshold C. The results are averaged, and noise is added such that this noise guarantees privacy protection. Finally, a noisy gradient descent according to these directions is taken. Algorithm 1 provides an outline of this mechanism.

Algorithm 1 Differential private SGD (Outline) [1]

Inputs: examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, \mathbf{x}_i)$.
Parameters: learning rate η_t, noise scale σ, batch size L, gradient norm bound C, number of epochs T.
for $t \in \{1, \ldots, T\}$ **do**
  Take L_t random samples with sampling probability L/N
  Compute gradient
  For each $i \in L_t$, compute $g_t(\mathbf{x}_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, \mathbf{x}_i)$
  Clip gradient
  $\overline{g}_t(\mathbf{x}_i) \leftarrow g_t(\mathbf{x}_i)/\max\left(1, \frac{\|g_t(\mathbf{x}_i\|_2}{C}\right)$
  Add noise
  $\hat{g}_t \leftarrow \frac{1}{L}\left(\sum_i \overline{g}_t(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})\right)$
  Descent
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \hat{g}_t$
Output: θ_T

This algorithm reflects popular mini-batch optimization techniques as are popular for the optimization of non-convex cost functions in machine learning. Unlike the pure version, added noise guarantees the algorithms' differential privacy. It has been shown by Abadi et al. that the resulting algorithm is (ε, δ)-differential private for any $\delta > 0$, provided $\sigma \in \Omega(q\sqrt{T\log(1/\delta)})/\varepsilon$ and $q = L/N$ is the sampling probability for one data point in the batch.

3 Differential Private GLVQ

In the following we describe how this training scheme can be used to optimize the cost function of the GLVQ model. The result will be a differentially private LVQ. Note, that we need to guarantee the differential privacy of all operations, including the prototype initialization and gradient update.

Initialization: For simplicity, we assume that we use one prototype per class (more general schemes are possible, e.g., relying on a differentially private version of neural gas, but would require more work). In standard GLVQ, we initialize each prototype by the class means. These can be calculated based on the sum of all data points of each class and the number of class members. These operations can be enhanced to DP versions based on the Laplace mechnaism as follows: In standard GLVQ, prototypes are initialized as $\mathbf{w}_j = \frac{1}{N_j}\sum_{i:c(\mathbf{x}_i)=j} \mathbf{x}_i$ for all classes.

The cardinalities of the classes are given by the function $f : \mathcal{D} \mapsto \mathbb{N}^k$, $f(D) = (N_1, N_2, \ldots, N_k)$. This function has a sensitivity $\Delta f = 1$ because adding or removing one data point in the dataset changes only the output of one N_i by one. In the literature, these functions are also known as a histogram queries [8].

The sums of all points in each class is given by the function $g : \mathcal{D} \mapsto \mathbb{R}^{k \cdot d}$, $g(D) = \left(\sum_{i:c(\mathbf{x}_i)=1} \mathbf{x}_i, \ldots, \sum_{i:c(\mathbf{x}_i)=k} \mathbf{x}_i \right)$. Without loss of generality, we assume that the data points are normalized so that $\mathcal{D} \subset [-1, 1]^d$. Then the sensitivity of the function is $\Delta g = d$. One adjacent dataset can change the output at least by one in each dimension in the L_1 norm because the classes are disjoint sets.

For a given privacy loss ε_1 we obtain all N_i and all sums with the Laplace Mechanism in a differentially private way. If we use the noise scales $\beta_f = \frac{2}{\varepsilon_1}$ for the function f and $\beta_g = \frac{2d}{\varepsilon_1}$ for g we achieve a ε_1-differential private mechanism altogether due to standard arguments for composition.

Gradient descent: For the gradient descent we rely on the algorithm as described above in chapter 2.4. Let L be the batch size, C the gradient norm bound, $q = L/N$, E the number of epochs and $T = \frac{E}{q}$ the runs of the Algorithm 1. For a given ε_2 and δ we can calculate the noise scale by $\sigma = \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon_2}$ [1]. Hence, the total privacy loss of the whole training is $\varepsilon = \varepsilon_1 + \varepsilon_2$. We obtain an (ε, δ)-differential private algorithm.

4 Results

We test our approach with the real world MNIST dataset of handwritten digits [10]. For the benchmark test, we do a 5-fold cross validation and use 50 epochs for the SGD. The total privacy loss is split into $\varepsilon_1 = 0.2\varepsilon$ for the initialization step and $\varepsilon_2 = 0.8\varepsilon$ for the gradient descent. The other parameters for the algorithm are $\delta = 10^{-5}$, $q = 0.01$ and $C = 0.4$. We compare our approach with the non-private version of GLVQ. There, the optimum is found by the BFGS algorithm, a quasi-Newton method for solving nonlinear optimization problems [9]. It represents the minimum error which we can reach based on GLVQ.

In addition to the algorithm as introduced above, we test the algorithm PrivGen [15]. The latter is a general-purpose differentially private model fitting framework which is based on genetic algorithms. For a given dataset D and a fitting-score function $f(D, \theta)$ that measures how well the parameter θ fits the dataset D, the PrivGen algorithm initializes a candidate set of possible parameters θ and iteratively refines them by mimicking the process of natural evolution. The best candidate is chosen by the exponential mechanism [11] and a new population is generated by mutation. We define the fitness function as the objective of the GLVQ problem, we use the same initialization as for our approach, and we use the default parameter according to the paper [15].

In Fig. 1 the GLVQ costs and the error rate of the test sets are plotted against the privacy loss for our approach, (non-private) BFGS optimization of GLVQ,

and PrivGen. For a privacy loss greater than 0.75 the error of our approach is as small as the one of classical GLVQ. PrivGen requires a privacy loss of 2.5 for a test error less than 20%.

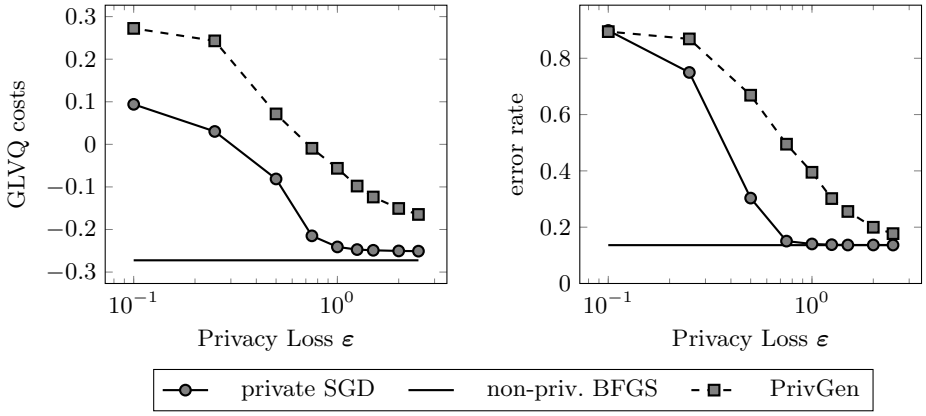


Fig. 1: GLVQ costs and test errors for our approach, for the non-private GLVQ version with BFGS optimization and for the different private fitting framework PrivGen on the MNIST dataset.

Fig. 2 depicts the impact of the norm bound C and the number of epochs E on the classification errors of the test sets. It also shows the test errors if we do not initialize the prototype before the differential private SGD as described in section 3 but use random points for each class as initialization.

In the left plot of Fig. 2 one can see that for very small C the test error rates rises. This is because an extremal clipping can cause the fact that the direction is completely different from the true gradient. On the other hand, a bigger value requires more noise. Analog results are obtained if the number of epochs is varied. Also here, a sweet spot can be observed for an increased number of epochs. It is clearly demonstrated that an appropriate initialization is important for a good solution.

5 Conclusions

We have introduced an approach to obtain a differential private version of GLVQ. We changed the initialization step and used a differential private SGD for optimization. In the results, we showed that for the real-world dataset MNIST a privacy loss $\varepsilon > 0.75$ suffices to achieve a differential private model that is as good as the non-private version with BFGS optimization. The method is robust to the choice of its hyperparameters.

This promising result opens the way towards a GLVQ variant which can publicly be released, e.g., in the medical domain albeit it has been trained based

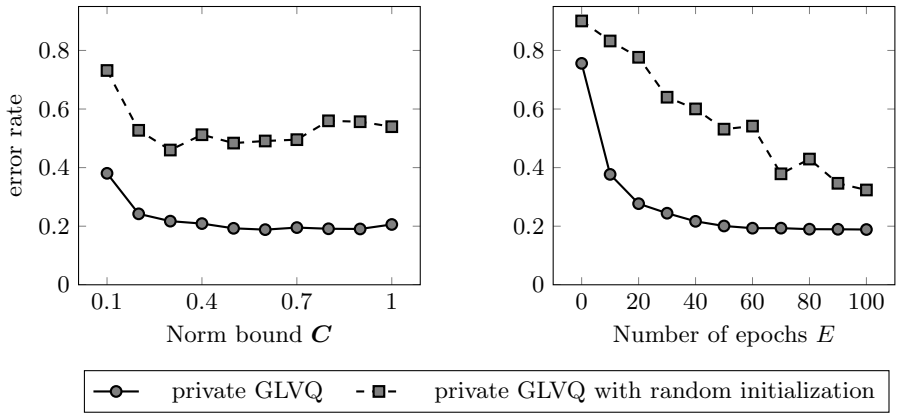


Fig. 2: Test errors for varied clipping bounds C and numbers of epochs E. The solid line is for our approach and the dashed one shows the error rates when no initialization is used and the prototype are just some random points. The parameters are: $\varepsilon = 1$, $\delta = 10^{-5}$, $\varepsilon_1 = 0.2\varepsilon$, $E = 50$, $C = 0.4$.

on sensitive data. So far, we have presented a differentail private version of standard GLVQ. It has been shown in practice that metric learning constitutes an essential step in metric-based models to achieve state of the art results, and methods such as generalized matrix LVQ are very popular demonstrations of this fact [14]. It is a matter of future work to extend the proposed DP formalism to these variants.

ACKNOWLEDGMENTS

Funding from DFG by the CITEC center of excellence (EXC277) and from the BMBF by the leading edge cluster “it’s owl” (intelligent technical systems OstWestfalenLippe) is gratefully acknowledged.

References

1. M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318, 2016.
2. M. Biehl. Biomedical applications of prototype based classifiers and relevance learning. In *AlCoB: 4th Int. Conference on Algorithms for Computational Biology*, pages 3–23, 2017.
3. M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.
4. J. R. Dowell. An overview of privacy and security requirements for data bases. In *Proceedings of the 15th Annual Southeast Regional Conference*, ACM-SE 15, pages 528–536, New York, NY, USA, 1977. ACM.

5. C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.
6. C. Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95, 2011.
7. C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.
8. C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
9. R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.
10. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.
11. F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103. IEEE Computer Society, 2007.
12. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pages 111–125, 2008.
13. A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 423–429. MIT Press, 1995.
14. P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.
15. J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. Privgene: differentially private model fitting using genetic algorithms. In K. A. Ross, D. Srivastava, and D. Papadias, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 665–676. ACM, 2013.

RoNiSCo: Robotic Night Shift Companion

Falko Lischke*, Frank Bahrmann, Sven Hellbach, and Hans-Joachim Böhme

HTW Dresden, Friedrich-List-Platz 1, 01069 Dresden, Germany

Abstract. This publication presents a comprehensive solution for an autonomous mobile robot platform that help caretakers in a stationary retirement home during the night shift. Consolidating algorithms and approaches from almost all research fields of robotics made it possible to create a complex system able to navigate freely in a learned environment, to recognize incidents like elderly getting lost and to contact a caretaker via a smartphone or computer application if that incident has to be reported.

1 Introduction & Related Work

We live in a modern world full of technology. In our daily life, technology reminds us of meetings and birthdays. Mobile devices plan the shortest path to a desired location while we walk, cycle or drive around. However, in several fields of work this modern comfortable support does not exist yet. This work investigates how it is possible to increase the quality of life in such a case: stationary retirement homes. Attention is not only paid to working fields of the medical staff but the elderly people as well.

In this publication a control architecture for a mobile robot platform (Scitos G5 by MetraLabs GmbH, Fig. 1a) is introduced. This robot is capable to patrol an indoor stationary retirement home environment looking for elderly people who lost their way. If such an incident happens, the staff is informed via an application on a mobile device. The availability of modern navigation algorithms in both mapping [15, 6, 2] as well as reactive motion planning [4, 10] enables the design of complex and sophisticated systems. The patrol service was requested multiple times in our talks with caregivers in 2 different facilities as well as listed in studies provided by [9]. Additionally, in a Japanese facility a robot was successfully deployed to patrol through rooms and monitor patients [16]. In the field of ambient assisted living, Cesta et al. in [5] developed a system which companionizes elderlies in their own homes. The robot in this work can switch between the role of active interactor via a PDA (*P*ersonal*D*igital*A*ssistant) and a silent observer. This enables the robot to manage appointments, to remind of the regular intake of medication or classify possible emergencies to call for help. Mobiserv [12] and CompanionAble [8] are other projects where robotic systems assist elderlies in their own homes but they do not offer a night shift function. Besides the intake of medication the field tests in [7, 11] showed the high acceptance and necessity of introducing mobile robotics into health care.

On the basis of the mentioned prior research, this publication presents a solution for a night shift patrol robot. Starting with an introduction of the setting of

* This work was supported in part by SAB grant number 100231931.

a) 

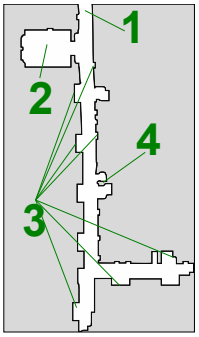
b) 

Fig. 1. a) A MetraLabs Scitos G5 robot equipped with forward and backward LaserRangeFinders and 2D and 3D camera sensors. b) Floor plan of the stationary retirement home learned by the robot - map size approx. 100x60 m (Legend: 1 - stairway/elevator, 2 - dining hall/community hall, 3 - examples of doorways to living quarters, 4 - ready room).

the environment in Section 2 and requirements in Section 3, the implementation of the control architecture is described in Section 4 with details for the robot's part in Section 4.1 and for client's part in Section 4.2. In Section 5 we report first test results. Section 6 presents a conclusion and outlook for further work.

2 Setting

The retirement home in which the proposed system is deployed consists of multiple 4 floor buildings. The floor plan of one building is shown in Fig. 1b. This map has been learned using robot's laser sensors. Adjustments can be made adaptively by use of fuzzy-based methods [1]. Due to the shortage of professional caretakers, there is only one staff member responsible for two floors. Consequently, the caretaker would need some time to notice if anything undesirable would have happened. Even if there were caretakers on all floors, the way the building was constructed would hinder the personnel to monitor the complete floor. In Fig. 1b all bulges (tagged with 3) are examples of doorways to the living quarters of the residents. The bottom right part as well as the dining hall (tagged with 2) pose a high risk. Both are blocked from the immediate view regardless of whether the staff enters the floor or is already in the ready room (labeled with 4).

The choice of a robot platform instead of a passive camera monitoring system is two-folded: Primarily, it arose from the human robot interaction possibilities. If an emergency occurs, the robot is able to distract the resident until help arrives as well as to support the nursing personnel with equipment in case of a first aid situation. The second reason is regulations that forbid to lock the building's entrance doors from the inside (e.g. in case of a fire emergency). Most residents of a stationary retirement home suffer from dementia. One of the symptoms is a distracted daily routine, which for instance can lead to the desire to go shopping in the middle of the night. Due to the fact, that doors are only able to be opened from the inside, in several occasions residents locked themselves out. A robot platform can bind those people by simply talking to them.

3 Requirements

As described in the previous section, the main purpose of the proposed system is to provide information to the care personnel about residents wandering the

hallways. This information is gathered and processed by a mobile robot platform. Using a robot makes it necessary to provide even more information about the robot itself, e.g. battery status, its current location, and its current task. To provide this information, the caregivers, who have to be available everywhere in the building, need to be equipped with an adequate mobile device. We decided to implement the system on a smartphone because we assume the personnel is already acquainted to its usage, it is easy to replace and due to its compactness.

Having a mobile robot allows for additional use cases like calling the robot to the caretaker's own position (e.g. to fetch a first aid kit) or to send the robot to a specific spot where something might have happened. For this, the caretaker needs to have the possibility to control the robot's position.

However, sending the robot on more complex missions (e. g. defining the waypoints for the patrol) might be a challenging task when performed on the screen of a smartphone. Hence, in an addition to the smartphone, we decided to integrate a base station with a fixed position in form of a laptop. So the mobile device is for fast and easy access, while the laptop is for configuration.

Allowing the communication between the devices makes a wireless network setup inevitable. A disadvantage of a wireless network is that clients can lose their network connection if the signal strength is too low. In such a case (or if the battery of the mobile device drained unnoticed), it is necessary to inform caretakers about the disconnected device. Furthermore, the network connection needs to be protected against illegitimate access. This guarantees both the protection of the collected data and the access limitation over the robot's control functionality.

Finally, following the principle of data reduction and data economy, only relevant data should be provided by the robot to improve application handling and clarity of displayed information, as well as to lower the risk of data abuse. It is useful to adapt displayed data automatically depending on the robot's situation to give caregivers as few information as necessary, which allows a fast evaluation of the situation around the robot in case of an emergency. Additionally, resulting from the usage of different sensors, e.g. cameras, which the robot needs to analyze its environment, data protection issues are of high concern.

4 Realization

In the following, we introduce a control system to support staff of stationary retirement homes. Since the system is equipped with different sensors like cameras, it is necessary to consider surveillance issues. We follow data privacy laws to increase acceptance of the robot platform. The system does not have the purpose to observe caregivers nor residents. The control system is designed such that it does not inflict the feeling of imprisonment or observation to the elderly people. The mobile robot and other used devices shall be unobtrusive parts of night shifts.

Figure 2 displays a network architecture in our scenario. Network connections are realized using WiFi connections. A router and several repeaters are placed

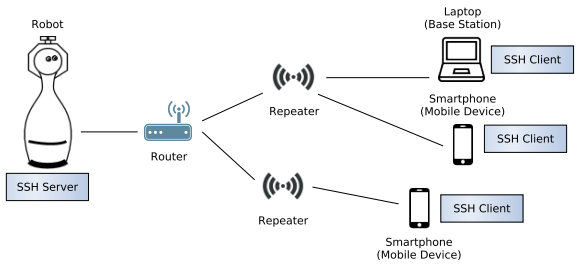


Fig. 2. Exemplary overview of the network architecture and end devices. A router spans a wireless network, which needs to be amplified by repeaters. The robot as a server as well as the clients, e.g. base station and mobile device, connect to the same network.

on the floor to span a wireless network. Our robot connects to the network and serves as application server. Stationary client devices like a laptop are called base stations. We use one base station placed in caregiver's ready room. If caregivers leave the room where a base station is placed they can use mobile client devices. In the network there is one robot and as many clients as necessary.

Communication between server and clients is protected by an SSH (*S*ecure *SH*ell) connection. The SSH server is running on the robot. All other client devices act as SSH clients. The user gives the login credentials to the client application, which tries to establish a SSH connection. After a successful login, an SSH tunnel is used for data exchange, thereby protecting the transmitted data from man in the middle attacks.

An overview of data exchange between server application and client application is displayed in Fig. 3. More details of both robot and client part are specified in subsequent Sections 4.1 and 4.2 respectively. At first the client application logs in to the server application with name and IP address. After that the client receives the current floor plan as image file. Depending on a so called incident status there exist different data messages. An incident status represents the robot's state and the environment's state. There are three different incident statuses:

- Green: Neither robot problems nor detected people.
- Yellow: Robot problem, colliding with an obstacle or investigating an uncertain person hypothesis. Section 4.1 contains a description what the term "person hypothesis" means.
- Red: Verified person hypothesis.

Incident green and yellow implement the same functionality. The server periodically sends a data message containing battery charge, position and incident status. Both statuses differ in the message's incident status part, which is either green or yellow. The client's acknowledge message is named heartbeat, which signals if the client is still connected. If the server misses multiple heartbeats of a registered client it sends a message containing name and IP address of the disconnected client to all other clients. This information is important to inform caretakers about undesirable disconnected clients.

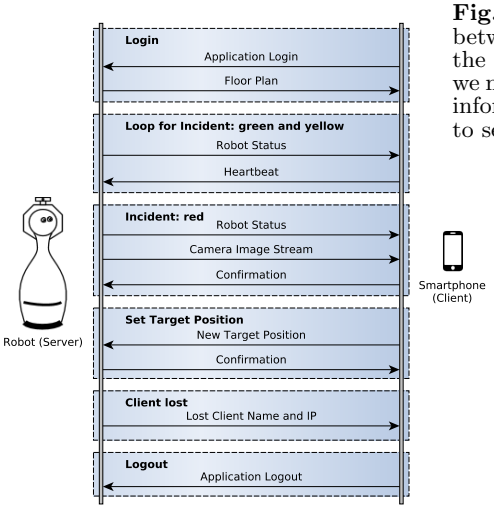


Fig. 3. Main groups of exchanged data between server and a single client. Besides the standard login and logout procedure, we need different categories of incidents to inform the personnel and the possibility to send control commands to the robot.

Incident red interrupts the periodical sending of status messages. The server also sends a status message containing battery charge, position and red incident status. In addition the server transmits a camera image stream. This image stream is provided by an omnidirectional RGB camera on the head of our mobile robot platform. A confirmation message to the server ends image streaming. This message is caused by the caregiver if she/he confirms the red incident on the device. Subsequently, the robot will continue its behavior in its previous incident state.

A mentioned requirement is the control of the robot's target positions by caretakers. The personnel is able to set a new target position for the robot using the client application. This new target position is transmitted to the server. If it is a valid target position the client will receive a confirmation message and the robot navigates to the given location.

Before a client logs out and closes its SSH session it sends a log-out message to the server application. This removes the client's IP address from its list of connected clients.

4.1 Robot Part

In the proposed scenario the robot consists on several subsystems (e.g. navigation and people detection). Depending on the robot's state, those subsystems are switched into different modes. For example, the navigation module can only be turned on or off. If it is turned on, the module generates movement commands based on a reactive local navigation algorithm [4]. The robot therefore tries to reach the predefined waypoints. If the module is turned off, the robot simply stops. The people detection module has 3 states: on, off, and hypothesis only. If the people detection module is switched on, the robot stores all uninvestigated person hypotheses in a temporary data structure. In the single hypothesis only mode, the robot focuses all sensors on the active hypothesis to investigate the actual incident.

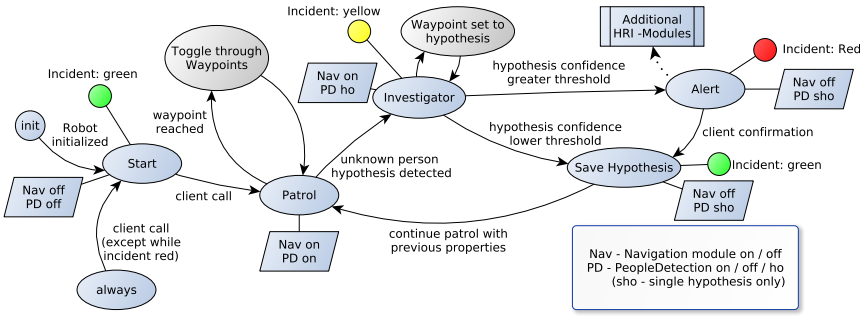


Fig. 4. Overview of the robot's states and the transitions between them. Ellipsoid nodes are states the robot can be in, the attached rhomboids denote statuses of the used subsystems. Transitions labeled with client are activated by a mobile device. The circular status (incident) nodes are colored consistently to represent the systems status until stated otherwise.

Figure 4 shows the implemented state machine. The robot starts after its initialization process in the Start node. Switching to this node can be achieved by a client's call at any given point except if a red incident occurred. In the start state, the robot simply waits for the command to begin patrolling. It can be used by the staff to disable the robot while they are working in the robot's vicinity. If a client starts the robot's mission, it will switch in the patrol mode and drive from waypoint to waypoint, scanning the map for possible persons. The underlying scanning algorithm is based on background subtraction of the current laser range finder data with a previously learned map. Due to the relatively simple representation of the world, ambiguous sensor readings are very likely. The robot collects those locations like stated earlier and switches to the Investigator mode. Now, the closest hypothesis found acts as a new temporary waypoint, causing the robot to drive to that location. This provides the possibility to use sensors with limited range capabilities like RGB (face detection [17]) or depth cameras and increases the density with which the laser range finder is able to scan the environment. Adaptive methods for each sensor can increase the individual recognition accuracy of people. All that information is combined using a sensor fusion approach proposed in [3, 14]. This fusion approach merges all sensor cues (each represented by a Gaussian distribution) to a resulting single Gaussian distribution. From that final distribution, the confidence value is calculated, which is utilized to decide whether a person hypothesis is accurate. This value is highly dependent on the amount and kind of sensors used. For the proposed case with laser range finder and RGB-D cameras the confidence threshold was determined empirically. Further studies have to be applied to investigate robust values. Instead of setting a threshold based on investigations, adaptive methods can also be used to adjust it. If the robot arrives at the closest point near the active location and the confidence value is below a threshold (mainly caused by tables, chairs or wall structures), the location is permanently stored as ambiguous - the robot learns to avoid false alarms. Afterwards, the mission is continued either with the next possible location or with the most recent waypoint. However,

if the confidence value is greater than the defined threshold, an alert is triggered, which sends a message with alarm to the clients. This certain people hypothesis causes a red incident. The robot now waits for the clients to confirm that alarm while sending an image stream to the personnel, allowing them to correctly assess the situation. During that time, additional Human-Robot-Interaction modules could be used to calm down or distract the lost resident until a caretaker arrives. As soon as the incident is resolved, the robot continues its mission.

4.2 Client Part

Different devices can be used as clients to show information provided by the robot. We use smartphones as mobile clients and regular laptops as base stations. With regard to a smooth introduction of the system, the KISS principle ("Keep it simple and stupid") was chosen to design graphical user interfaces. Intuitive application handling like a possibility to set a new target position and information understanding like incident reports should reach the KISS principle.

An important aspect to guarantee usability is to limit the amount of possible user interaction to the necessary minimum. Hence, we avoid confusion during handling by implementing shallow menu structures which ensures reachability with only a few interactions. There are clearly named buttons and option menu entries to control the application. Every display shows only few but relevant information. Besides textual information such as battery charge and state message, there is a map image displaying the robot's position. Figure 5a and 5b show status display and control display for status incident green. Sending the robot to a new target position, e.g. ready room, can be done with only one button and a command confirmation. Robot movement is always observable on the map where an icon marks the robot. The icon color depends on the robot's state.

We use a traffic light principle for the robot state. In addition to three textual state description possibilities there are three colors. Every icon color is associated with an incident status. A green icon implies that there is no considerable situation to report. If there is a less prioritized incident, e.g. assumption of a person under uncertainty, a yellow icon will be used. Figure 5c shows status screen with changed state message for incident yellow. Person incidents get highest ranking and show a red icon. Such incidents are certain people detections or detections of fallen people.

As it is important to inform caretakers immediately about red incidents there are additional security arrangements. Base stations as well as smartphone clients receive a camera stream of the current scene around the robot and a location description. A red incident message is showed in Fig. 5d. Mobile devices use additional functions like acoustic alert and vibration to indicate a red incident. Caretakers have to confirm the incident red message for each client.

We prevent undesirable use and setting changes of the smartphones using the following arrangements.

– It is conceivable that a user turns off WiFi to disconnect from the robot and doesn't receive incident messages. If WiFi is turned off the application will switch it on and reconnect to network.

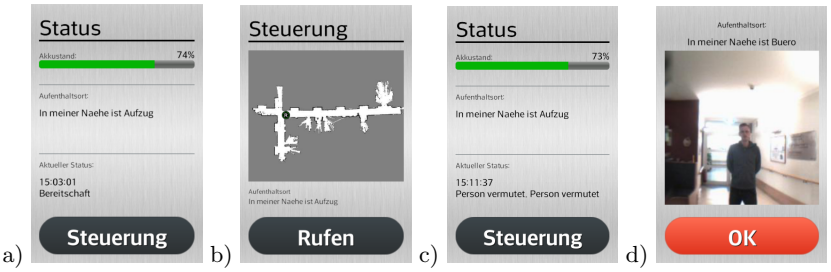


Fig. 5. a) A status display for incident green shows battery charge, textual position description and status. b) A control display shows floor plan and an incident state dependent icon marks the robot position. Beneath the map is a textual position description. Caretakers can set a new target position on the map and send the robot using the button. The smartphone's menu button allows the return to status display. c) Status display shows status message for incident state yellow. In this case there is an uncertain people hypothesis. d) Incident display showing robot location and camera stream. The detected person is centered in the image. Alert sound and vibration of mobile devices stops if the red button is pressed.

- Our application returns to the screen if it has been switched to background.
- We disable muting the smartphone. A red status incident sets the volume to a configured value and plays a message sound.

Another arrangement for all clients is an application execution after boot up of the device. Staff members do not need to take care of starting the application, thus the risk of inaccurate device usage is minimized.

5 Preliminary Test Results

Prior to the tests in the night shift, during daytime, the robot acted as card player and infotainment system to give the elderlies the chance to become accustomed to the robot's presence. After that, we executed tests with a limited number of caretakers and elderly people during night shifts. The circumstance, that only a limited number of test persons were available is mitigated by Nielsen and Landauer's proposal [13], that only a few test persons are sufficient to get a first estimation of a system's usability.

Successive interviews with involved residents and caretakers showed an appreciation of the robot. Caretakers approved the robot because they had time to do important work instead of patroling on the floors themselves. Residents enjoyed the presence of the patroling robot and its appearance. The robot was a diversion in their routine and a eye-catcher in the retirement home. Despite the limited number of test persons, our test results are a first baseline to assume a high acceptance of our patrol system. Suggestions coming from the care personnel overlap with our already planned extensions (confer Section 6).

Caretakers performed their work while the robot patrolled the floor. Throughout the duration of the tests, the amount of hypotheses that had to be investigated diminished as expected. This means that the system learned false-positive hypotheses correctly during the test phase, resulting in fewer mistaken red incidents over time. Wandering residents were recognized correctly.

6 Conclusion and Outlook

This publication presents a system to support caretakers in stationary retirement home during night shift. A mobile robot can patrol on the corridors instead of a caretaker. If an incident occurs all connected client devices will receive a message containing incident detail so they can react. While the robot fulfills the patrol task, caretakers have time to do other tasks.

Further work will extend the current system with person-oriented behavior and security features. This will improve security for residents as well as support for caretakers. An important step is to perform more user studies to consider user requests. Results will show user acceptance, weak points during work, impact on the caretakers time management and number of incidents during night shift.

Different adaptive approaches can be used to reduce the number of false red incidents. Based on a distinction between personnel and residents, the system can learn which residents are allowed by staff to leave their room at night to fetch water, for example. Due to this adaptive behaviour, only those who are not allowed to be on the corridor will be reported over time.

Currently, we use a straight forward way to detect persons in the robot surroundings. This is far from reliable. However it serves for proof of concept. To make our approach more robust it is also possible to distinguish people from movable objects standing on the floor using results of background subtraction. Differences between the current map and the stored map can be used to classify it as a movable object or person. Object representations can be learned adaptively.

Reactions of the current system are limited to sending a message to all clients if the robot detects a red status incident. This applies for people detection during patrol on corridors too. Stationary retirement home's residents mainly suffer from various forms of dementia. They tend to forget where they are and possibly leave the retirement home until a caretaker is on location. At present the robot is unable to prevent residents from leaving. A simple possibility to do that is to try a conversation. Small-talk is meant to stop the wandering elderly and gain time for arrival of a caretaker. An initial dialogue system can be trained for communication with residents suffer from dementia. The longer the wandering resident is prevented from continuing to walk, the better the dialogue system is. Personal information of the resident allows a more personalized dialogue, which should bind the resident longer and give the staff more time to arrive.

Wandering is a potential risk for elderly people. In addition to a small-talk function to stop them, we plan to extend the module for people detection. This extension includes fall prevention and fall detection. Both are important functions in an environment where the fall risk is increased. In the case of recognizing a fallen person the robot informs caretakers by sending a state red incident message to all client devices. Additionally, the robot has the ability to speak with the fallen person to inform her or him that help will arrive shortly. A dialogue system as well as fall prevention and fall detection require components of soft computing to achieve adaptive behavior.

All mentioned additions promise to increase support potential for a time-consuming task during night shift for staff and additional security for residents.

The suggested night shift system will serve as a test bed for multiple methods in the context of adaptive dialogue strategies, adaptive strategies for estimation of interest for communication with the robot, fall prevention and fall detection, robot navigation, robot motion planning, mapping, people detection and recognition.

References

1. F. Bahrmann, S. Hellbach, and H.-J. Böhme. A fuzzy-based adaptive environment model for indoor robot localization. In *Telehealth and Assistive Technology / 847: Intelligent Systems and Robotics*. ACTA Press, 2016.
2. F. Bahrmann, S. Hellbach, S. Keil, and H.-J. Böhme. *Understanding Dynamic Environments with Fuzzy Perception*, pages 553–562. Springer International Publishing, Cham, 2014.
3. N. Bellotto and H. Hu. Vision and laser data fusion for tracking people with a mobile robot. In *IEEE ROBIO*, pages 7–12. IEEE, 2006.
4. H. Berti, A. Sappa, and O. Agamennoni. Improved dynamic window approach by using lyapunov stability criteria. *Latin American applied research*, 38(4):289, 2008.
5. A. Cesta, G. Cortellessa, F. Pecora, and R. Rasconi. Supporting interaction in the robocare intelligent assistive environment. In *AAAI Spring Symposium: Interaction Challenges for Intelligent Assistants*, pages 18–25, 2007.
6. E. Einhorn and H.-M. Gross. Generic 2d/3d slam with ndt maps for lifelong application. In *ECMR*, pages 240–247. IEEE, 2013.
7. B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein. Robotic home assistant care-o-bot® 3-product vision and innovation platform. In *IEEE ARSO Workshop*, pages 139–144. IEEE, 2009.
8. H.-M. Gross, C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, T. Langner, C. Martin, and M. Merten. I'll keep an eye on you: home robot companion for elderly people with cognitive impairment. In *IEEE SMC*, pages 2481–2488. IEEE, 2011.
9. D. Hebesberger, T. Körtner, J. Pripfl, and M. Hanheide. What do staff in eldercare want a robot for? An assessment of potential tasks and user requirements for a long-term deployment . In *IROS Workshop on "Bridging user needs to deployed applications of service robots"*, Hamburg, 2015.
10. S. Hellbach, F. Bahrmann, M. Donner, M. Himstedt, M. Klingner, J. Fonfara, P. Poschmann, R. Schmidt, and H.-J. Böhme. Learning as an essential ingredient for a tour guide robot. *Workshop New Challenges in Neural Computation 2013*, page 53, 2013.
11. T. Jacobs and B. Graf. Practical evaluation of service robots for support and routine tasks in an elderly care facility. In *IEEE ARSO Workshop*, pages 46–49. IEEE, 2012.
12. M. Nani, P. Caleb-Solly, S. Dogramadgi, C. Fear, and H. van den Heuvel. Mobiserv: An integrated intelligent home environment for the provision of health, nutrition and mobility services to the elderly. In *4th Companion Robotics Workshop in Brussels, Brussels, 30th September*, 2010.
13. J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *Proc. of INTERACT '93 and CHI '93*, CHI '93, pages 206–213, New York, NY, USA, 1993. ACM.
14. P. Poschmann, S. Hellbach, and H.-J. Böhme. Multi-modal people tracking for an awareness behavior of an interactive tour-guide robot. In *Intelligent Robotics and Applications*, pages 666–675. Springer, 2012.
15. J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal. Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *IEEE ICRA*, pages 2233–2238. IEEE, 2013.
16. K. Sato, M. Ishii, and H. Madokoro. Testing and evaluation of a patrol robot system for hospitals. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 86(12):14–26, 2003.
17. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, volume 1, pages I–511. IEEE, 2001.

Effects of Domain Awareness in Generalizing over Cameras in Road Detection

Sebastian Schrom[1,2] and Stephan Hasler[2]

[1] Technical University of Darmstadt
[2] Honda Research Institute Europe GmbH

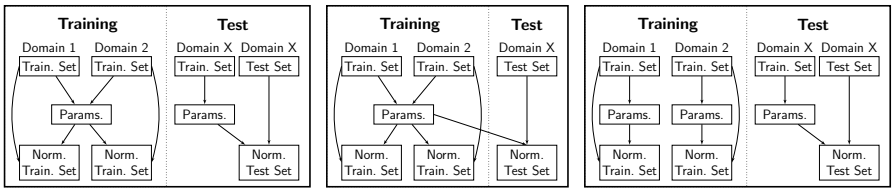
Abstract. In this paper we investigate the effects of domain awareness when using a pre-trained covolutional neural network (CNN). Usually, the only adaptation when using such a CNN for the same task is to normalize the input data by an individual RGB mean from own data. We show that it plays a major role whether the test domain was included during training and if training was aware of domains. For this we investigate generalization over few cameras in a road detection task as a domain transfer scenario. We train CNNs for all combinations of used cameras and test each camera individually. We apply RGB mean subtraction in three different cases of domain awareness during training and test. Our results reveal a harmful effect if the test domain was included during the network training, but not considered as an individual domain.

1 Introduction

In recent years deep neural networks showed impressive results on several large-scale object recognition benchmarks [7, 13]. However, when such pre-trained networks are applied on images of a different benchmark the observed performance can be substantially reduced. This problem is referred to as domain transfer. A standard way to minimize the effects of domain transfer is to adapt some parameters of the pre-trained network to the target domain, e.g. usually the input data is normalized using the RGB mean of the target domain.

Most domain transfer studies focus on the transfer from a single (or multiple) source domain(s) to a single new target domain only [10, 16]. This is the most difficult case which we will call the *new*-case of generalization. Often these results are interpreted in relation to the simplest case, where source and target domain are the same [14, 15]. We call this the *same*-case. Only rarely the case is considered that the target domain was one of multiple source domains (e.g. [6]). From our point of view this case needs to be considered more, since downloadable networks are usually trained on huge image collections from the internet which potentially contain many different domains. So e.g. the test data might be from a camera that was also used to acquire some images of the training data. In our work we will focus on this *part*-case of generalization and show that domain adaptation can be harmful if training was not aware of domains.

Usually the entire training set is treated as a single domain and the parameters used to normalize the input are generated from the entire training set. Later,



(a) *Test*-aware (b) *Never*-aware (c) *Always*-aware

Fig. 1. Three different cases of domain awareness.

during test the parameters are re-estimated for the target domain. This is the standard workflow for using a pre-trained network and we call this case of domain awareness the *test*-aware case (see Fig. 1a). Since we saw a weak performance for the *test*-aware case in the *part*-case of generalization, we concluded this might be due to inconsistent normalization for the same domain during training and test. To evaluate this further we additionally tested two consistent cases of domain awareness. The *never*-aware case and the *always*-aware case. The *never*-aware case (see Fig. 1b) is unaware of domains during both, training and test. Here, the normalization parameters are also calculated over the entire training set but later the same parameter set is used to normalize data of the test domain as well. However, this way of handling domain transfer is not recommended since there is no real adaptation to the test domain and a weak performance for the *new*-case of generalization can be expected. The second consistent awareness case we investigated is the *always*-aware case (see Fig. 1c). It is aware of domains while training and test and computes during training the parameters individually for each domain. However, considering all domains during training is usually not feasible because there might be many which might also overlap.

In our work we extensively evaluate all generalization cases, the *same*-case the *part*-case and the *new*-case with all three cases of domain awareness, the *test*-aware case, the *never*-aware case and the *always*-aware case. For this we have generated a controlled setting to focus on the effects of domain awareness:
1) We have chosen the task of segmenting an image into *road-like-area* and *non-road-like-area*. Usually this is done with architectures that model large receptive fields to take global image relations into account like in [3] and [1]. But we focus on classification of small image patches instead, which allows us the use of a simple CNN architecture. This is sufficient for our goal to show the general effects of domain awareness.
2) We have a controlled dataset where the domains are three cameras that strongly differ in their sensitivity of color channels. Usually other datasets like the frequently used Office dataset proposed in [12] are less controlled as they are partly a result of a web search and might hide several unknown domains inside. Hence, results on these datasets are caused by a mixture of effects, which hinders a qualitative interpretation.
3) We use the the standard input normalization method of RGB mean subtrac-

tion [7, 13], which is usually done before training a CNN to improve convergence speed [9], as a simple unsupervised way of domain adaptation. This means we interpret the "Params." from Fig. 1 as the RGB mean. In general, more sophisticated approaches of domain adaptation range from nonlinear transformations [8] to complex domain confusion losses within CNNs [16]. Nevertheless, there are methods almost as simple as our method as e.g. batch normalization [4], which was originally designed for improving convergence speed by normalizing mean and standard deviation of the batch-wise output of several layers. In [10] they showed that this technique can be successfully used for domain adaptation by re-estimating the normalization parameters for the target data.

The remainder of the paper is structured as follows. In Sec. 2, we describe the used image data and the neural network architecture of our experiments. The results for the different cases of domain awareness will be extensively compared for the different generalization cases in Sec. 3. Finally, we give a conclusion and a short outlook on future work in Sec. 4.

2 Data & Methods

To test the generalization over the three used cameras, we train an individual CNN for each single camera, for each possible pair, and for all cameras together. Each of these seven networks is tested against each individual camera. This means our tests include the three previously presented generalization cases, the *same*-case, the *part*-case and the *new*-case. We repeat this extensive evaluation for the different cases of domain awareness. Before comparing these results in Sec. 3 we will give further details on the used data and methods in the following.

2.1 Data

For our experiments we distinguished between images of three RGB cameras, each looking to the front of the ego car. Our first data set is the publicly available KITTI Road Benchmark [2]. The images from the other two datasets were acquired in-house using a BlackMagic[1] (BMAG) and an ELESYS[2] camera. The images of each camera stem from different recording sessions, containing different routes, illuminations and weather conditions. This increases the chance that the prominent domains in our data are the different cameras and not some hidden condition of the environment. This notion is confirmed in Table 1 when looking at the channel-wise means over pixels of training patches, where we see a strong difference of individual channels over cameras Each image comes with a manual annotation of one or more *road-like-area* polygons. In general, image patches whose center is in the *road-like-area* are used as positive examples and all other patches as negative examples. We limit the extraction of patches with a size of 37×37 pixels to a region that corresponds to a rectangular corridor in the metric Bird's-Eye-View[3][11] space. With this we mainly ensure that the

[1] http://www.blackmagicdesign.com/de/products/blackmagicpocketcinemacamera
[2] Special in-house made, no external reference available
[3] The mounting position and angles are known for each camera

Table 1. Data statistics of the different cameras.

		KITTI	BMAG	ELESYS
Red channel mean		96.5	115.3	83.3
Green channel mean		97.0	136.0	89.6
Blue channel mean		92.2	132.0	55.5
Training	#Images	289	113	166
	#Samples	2,312,000	5,085,000	2,822,000
	Pos./neg.	1/1	1/1	1/1
Test	#Images	290	111	166
	#Samples	42,165,099	85,041,497	49,325,077
	Pos./neg.	∼1/1	∼2/1	∼2/1

Fig. 2. Example image of BMAG camera. The colored area in the perspective view (left) denotes pixels inside the metric Bird's-Eye-View corridor (right).

negative patches focus on regions close to the road, while e.g. removing everything above the horizon, like the sky. In Fig. 2 we show an image of the BMAG camera where the region that corresponds to the Bird's-Eye-View corridor in our work is highlighted.

We split the available images for each camera into a training and a test set of similar size and extract patches with a size corresponding to the input dimension of our neural network. To speed up training we substantially reduced the number of available patches by randomly choosing roughly 5% of them, and we also balanced the number of positive and negative samples. During test we classify all negative and positive patches inside the Bird's-Eye-View corridor. The numbers of used training and test patches are given in Table 1. It can be seen that the number of training samples is imbalanced over different cameras. However, we experimented with more complex balancing strategies but could not find any substantial change in reported results.

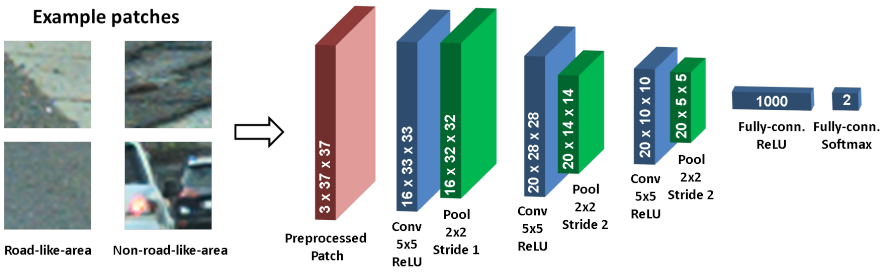


Fig. 3. CNN architecture. The dimension of each layer is shown in white color.

2.2 Network Architecture & Training

For our experiments we use RGB mean subtraction as a simple form of domain adaptation. This method is relatively cheap, as it is a preprocessing step whose few parameters can be estimated from a collection of images without any ground-truth information. More specifically, we used a global channel-wise RGB mean, i.e. we subtracted from each color channel a fixed value for normalization. This corresponds to the notion that the domains that we like to generalize over are defined by different sensitivities of the color channels among the cameras. We use a CNN to predict for a patch of 37×37 pixels whether it belongs to *road-like-area* or *non-road-like-area*. The architecture of the network is shown in Fig. 3 together with some example patches. We implemented the CNN in Caffe [5] and optimized it with Stochastic Gradient Descent using the cross-entropy classification loss. We used a batch size of 50 and trained our architecture for 20 epochs. The initial learning rate was set to 0.001 and constantly halved after ten percent of the maximum iterations. Furthermore, we used a momentum parameter of 0.9, a weight decay factor of 0.0005, and 50% dropout on the output of the first fully-connected layer.

3 Results

During test we interpret the value of the *road-like-area* neuron in the final network layer as a confidence score and measure performance using Average Precision (AP). However, the absolute performance is not in focus, but the change in performance for the different generalization and domain awareness cases. Please note that AP results for the KITTI Road Benchmark [2] are reported for the Bird's-Eye-View space, while we measure performance in image space.

The results of the ***test*-aware** case (see Fig. 1a) are presented in the corresponding column in Table 2, where the upper rows show the results for all experiments and the three lower rows give the averaged results for the three generalization cases. The performance is consistently high whenever the test camera is the only training camera, i.e. in the *same*-case. In the *part*-case, the

Table 2. Results of different cases of domain awareness for different combinations of training and test cameras (AP in %). Blue represents the *same*-case, gray the *part*-case and green the *new*-case. Numbers in gray indicate unchanged results in comparison to previous cases of domain awareness. The averaged results are shown at the bottom.

Used cameras		**Domain awareness**		
Training	**Test**	*Test*	*Never*	*Always*
KITTI	KITTI	93.94	93.94	93.94
KITTI + BMAG	KITTI	89.66	93.76	93.64
KITTI + ELESYS	KITTI	90.12	94.04	93.93
KITTI + BMAG + ELESYS	KITTI	88.03	93.72	93.82
BMAG	KITTI	85.99	58.57	85.99
ELESYS	KITTI	90.10	86.52	90.10
BMAG + ELESYS	KITTI	78.55	59.99	90.04
BMAG	BMAG	96.29	96.29	96.29
BMAG + KITTI	BMAG	95.65	96.27	96.25
BMAG + ELESYS	BMAG	93.12	96.31	96.24
BMAG + KITTI + ELESYS	BMAG	94.80	96.21	96.28
KITTI	BMAG	93.28	73.68	93.28
ELESYS	BMAG	93.70	89.88	93.70
KITTI + ELESYS	BMAG	80.65	80.08	93.46
ELESYS	ELESYS	95.97	95.97	95.97
ELESYS + KITTI	ELESYS	91.33	96.02	95.97
ELESYS + BMAG	ELESYS	86.68	95.77	95.91
ELESYS + KITTI + BMAG	ELESYS	80.25	95.80	95.88
KITTI	ELESYS	86.67	77.12	86.67
BMAG	ELESYS	84.11	74.35	84.11
KITTI + BMAG	ELESYS	80.66	75.46	80.02
Averages	*Same*	95.40	95.40	95.40
	Part	89.96	95.32	95.32
	New	85.97	75.07	88.60

performance drops to 89.96 AP on average, which is surprising, since the network saw similar data already while training and is expected to generalize over such data. A possible explanation for this effect is the inconsistent handling of the same camera during training and test. During test the mean of the considered camera is used, while during training the mean over this camera and one or two other cameras is computed. We think that the CNN learned to specialize too much on the combination of camera data and preprocessing and later is basically surprised to see a familiar camera but with changed characteristics of input channels. For the *new*-case we get the worst performance with 85.97 AP on average, which could be expected, since the cameras differ besides the RGB means also in factors like image noise or resolution.

In the ***never*-aware** case (see Fig. 1b) the mean that was used while training is also used for normalizing the test images. This strategy does not influence the *same*-case. As expected, since there is no real domain adaptation, it strongly affects the *new*-case where the average performance is only 75.07 AP. A visual analysis revealed that for the *new*-case the predicted confidence for *road-like-area* is either quite high or low over the test set. This is a potential indicator that the different characteristics of the input channels of the new camera drive the trained network outside its working range. Interestingly, which is one major result of this work, we see a significant enhancement of performance for the *part*-case. This shows that consistent treatment of domains while training and test is for this case even better than domain adaptation.

In the ***always*-aware** case the data of each camera is normalized independently during training and test (see Fig. 1c). With this we also avoid the inconsistency in the *part*-case, as now each camera uses its own mean during training and test. Correspondingly, we see in Table 2 that the performance for this case stays at the high level of the *never*-aware case. Furthermore, in the *new*-case on KITTI we also see a strong improvement when trained on BMAG+ELESYS, and also for BMAG when trained on KITTI+ELESYS. A reason for this can be that the awareness of domains during training helps the classifier to focus resources on a more general and consistent representation of *road-like-area*.

4 Conclusion

In this paper we investigated the effects of domain awareness in a controlled setting that was based on a simple road segmentation task with a CNN and RGB mean normalization as a simple domain adaptation method. We could reveal that the standard workflow of re-estimating the normalization parameters for the test domain can have a harmful effect if the test domain was already involved among other domains during training. Here, the performance is significantly reduced if a domain is not treated consistently during training and test. However, due to insufficient knowledge about domains when using a pre-trained network it is often not clear whether the test domain was involved and how it was treated if it was one of several. To summarize, the major outcome of our work is that for practice it is important to find out whether the training data of a pre-trained network contained already the target domain, and if so, if the optimization process was aware of different domains and then acting according to that. We think that our rather controlled domain setting together with the simple adaptation strategy strongly helped to discover some of the reported effects, but in general, we expect that our results also hold for less controlled settings. To test this will be the primary direction for future work.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)

2. Fritsch, J., Kühnl, T., Geiger, A.: A new performance measure and evaluation benchmark for road detection algorithms. In: ITSC. pp. 1693–1700 (2013)
3. Fritsch, J., Kühnl, T., Kummert, F.: Monocular road terrain detection by combining visual and spatial information. IEEE Transactions on Intelligent Transportation Systems 15(4), 1586–1596 (2014)
4. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015)
5. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM-MM. pp. 675–678 (2014)
6. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: ECCV. pp. 158–171 (2012)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
8. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR. pp. 1785–1792 (2011)
9. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade, pp. 9–48. Springer (2012)
10. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779 (2017)
11. Mallot, H.A., Bülthoff, H.H., Little, J., Bohrer, S.: Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics 64(3), 177–185 (1991)
12. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. pp. 213–226. Springer (2010)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
14. Tommasi, T., Patricia, N., Caputo, B., Tuytelaars, T.: A deeper look at dataset bias. In: GCPR. pp. 504–516. Springer (2015)
15. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR. pp. 1521–1528 (2011)
16. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)

Strategies for Improving Camera to Map Alignment

J. Silberbauer[1], B. Flade[2], S. Hasler[2], M. Probst[2], J. Eggert[2]

[1] Technical University Darmstadt
[2] Honda Research Institute Europe GmbH

Abstract. Accurate localization of the ego vehicle is a key requirement for contemporary and feature advanced driver assistance systems (ADAS). Map relative localization can be achieved by aligning monocular front camera images to low precision map data. An initial approach uses an iterative strategy to match histogram of oriented gradients (HoG) features. We propose two new strategies for improving this approach: First, we replace the original feature extraction step by HoG computation on a semantic segmentation of the camera image. Secondly, we train an end-to-end convolutional neural network (CNN) to directly predict the correct alignment. We evaluate the approach on a data set recorded on rural roads in German cities on which both approaches significantly improve alignment accuracy. Furthermore we show that an end-to-end learning approach can successfully be used in this context allowing the alignment to be performed in a single forward pass. In this context we also present current challenges like obtaining accurate ground truth data.

1 Introduction

Future ADAS attempt to offer advanced functionality such as risk prediction or lane level navigation. Therefore knowledge of the ego vehicles position and orientation relative to the map is required. Standard absolute localization approaches relying on GNSS fail since they do not account for errors related to the map i.e. an offset to real world coordinates. To circumvent this problem, some approaches for map relative localization have been investigated, i.e. [1]. While offering highly accurate localization such strategies mostly require maps build with expensive equipment. As an attractive alternative Cao et al. [2] show that monocular camera images and low precision maps can be used for inexpensive self-localization in the context of ADAS. This paper presents strategies for improving this original approach.

2 Improvement Strategies

The algorithm from Cao et al. aims at correcting the coarse position estimation provided by GNSS. To that end a perspective view of the map (a candidate image) is created. To determine a position correction multiple candidate images

(a) (b) (c) (d)

Fig. 1: Candidates for the camera image in (a) from the KITTI data set [4].

with different offsets from the GNSS position are generated (Figure 1). The corrected position is the initial GNSS position plus the offset minimizing the cosine distance between HoG features [3] extracted from both the candidate and the camera image. We propose two improvements to the original approach which we evaluate focusing on lateral translations. As a first strategy we want to reduce the effects of dominant non-road edges in the camera image by partly replacing the feature extraction step of the original algorithm. This is done by using a pre-trained segmentation model presented in [5] to predict the road area in the camera image before applying HoG. As a second approach we will eliminate the iterative nature of the algorithm by directly learning the correct alignment. For that we use a CNN that takes candidate and camera image as input. The output is the lateral translation d that aligns candidate to camera image.

3 Experimental Setting and Results

We evaluate all three approaches on a data set of 540 positions from two KITTI streams [4] split into 80% train and 20% test subsets and use the KITTI positioning data as ground truth. For each image in the training set 200 candidate images are rendered with offsets $d_r \in [-2m, 2m]$. The goal of the evaluation is to predict these offsets. For both HoG based approaches we determine the offset by optimizing across 31 lateral translations ($d_h \in [-3m, 3m]$) around the starting position. The CNN approach predicts the offset directly. We consider the mean absolute (mae) and root mean squared error (rmse) in lateral translation with respect to ground truth. Figure 2 shows the predictions for all three approaches on the test set. The HoG baseline algorithm results in an mae/rmse of 0.79m/1.37m, the HoG approach with segmentation produces 0.33m/0.55m and the CNN results in 0.2m/0.3m.

4 Conclusion and Outlook

The presented results demonstrate that both approaches significantly improve alignment accuracy. The better performance of the CNN is likely due to the fact that the data set has a low diversity of driving situations suggesting that the model is over fitting to situations contained therein. More specifically this means that the model is likely to perform worse on other data then the current data set. However, this remains to be tested and the current results still provide indications for the general feasibility of the approach. A challenge is that current ground truth data is not always accurate which causes the model to learn

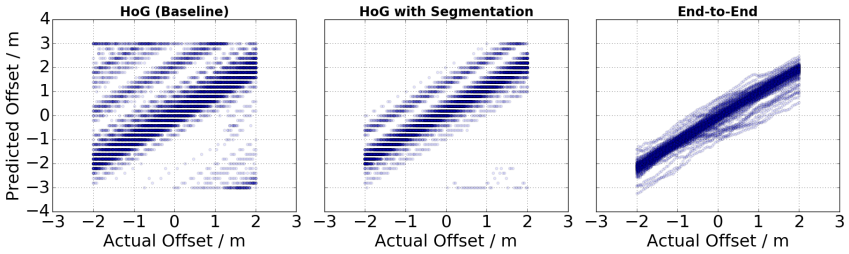


Fig. 2: Performance of the three algorithms on the test set.

a visually imprecise alignment. In future work we will test the approach on a data set containing more diverse driving situations. However, obtaining accurate ground truth is a major challenge which might require manual generation. Furthermore we will extend the evaluation beyond a single lateral degree of freedom where we would like to evaluate the algorithm's run time as well. In that case we expect alignment to be dramatically slower for the HoG approaches needing to optimize offsets in multiple dimensions i.e. longitudinal, lateral and altitude. In contrast to that the CNN approach would still be able to predict the correct offset in a single forward pass. This has been demonstrated in other domains [6], but still remains to be proven here which is a challenging problem because visual alignment might be ambiguous i.e. regarding longitudinal translation on a straight road.

Acknowledgments

This work has been supported by the European Union's Horizon 2020 project *INLANE*, under the grant agreement number 687458.

References

1. Yan Lu, Jiawei Huang, Yi-Ting Chen, and Bernd Heisele. Monocular localization in urban environments using road markings. In *IV, 2017 IEEE*, pages 468–474. IEEE, 2017.
2. G. Cao, F. Damerow, B. Flade, M. Helmling, and J. Eggert. Camera to map alignment for accurate low-cost lane-level scene interpretation. In *2016 IEEE 19th ITSC*, pages 498–504, Nov 2016.
3. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE CVPR*, volume 1, pages 886–893 vol. 1, June 2005.
4. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013.
5. M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.
6. S. Miao, Z. J. Wang, and R. Liao. A CNN regression approach for real-time 2D/3D registration. *IEEE Transactions on Medical Imaging*, 35(5):1352–1363, May 2016.

Grassmann Manifolds for Prototype Based Classification Learning

Thomas Villmann

Computational Intelligence Group,
University of Applied Sciences Mittweida, Germany

Abstract. Robust classification of data by machine learning tools should be able to deal with variations of objects like different representations, data noise or drifts. In this paper we discuss the differential-geometric concept of Grassmann manifolds for variation-tolerant classification learning by means of learning vector quantization (LVQ). Particularly, we review the mathematical foundations of Grassmann manifolds and show how to apply them in the LVQ framework.

1 Introduction

Pattern recognition and classification learning frequently has to deal variations of objects, which lead to respective alterations in the describing feature vector $\mathbf{x} \in \mathbb{R}^n$. Examples are different illuminations or rotations in images or genome variations for different species in genome data analysis. Otherwise, simple data noise or data drift can also lead to sever variations in feature vectors. Those data can be seen as sample vectors belonging to a data space describing the object together with its variations.

Prototype based classification learning aims to distribute reference vectors in the data space to detect the class distributions and to represent the data according to the nearest prototype principle [1]. For this purpose, the most important ingredient is the choice of an appropriate dissimilarity measure to judge the (dis-)similarities between data and prototypes [2]. This becomes essentially crucial in the light of the above mentioned versatilities in data.

Several approaches exist to tackle these problems adequately. Frequently, particular preprocessing tools are applied like filtering, data clustering, dimensionality reduction or compression to name just a few. For sequence data $\mathbf{y} = \{y_1, \ldots, y_{D+N}\}$ the time history can be used to identify the correct pattern also in the presence of noise or other data distortions. For example, if it is assumed that a hidden (linear) dynamical system is generating the time series, the Hankel matrix

$$\mathbf{H}(\mathbf{y}) = \begin{pmatrix} y_1 & y_2 & y_3 & \cdots & y_D \\ y_2 & y_3 & y_4 & \cdots & y_{D+1} \\ y_3 & y_4 & y_5 & \cdots & y_{D+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_N & y_{N+1} & y_{N+2} & \cdots & y_{D+N} \end{pmatrix} \tag{1}$$

can be considered as an appropriate data feature, because the regarding linear subspace $\mathfrak{H}(\mathbf{H})$ spanned by this matrix shows several invariance properties for time series analysis like drifts or shifts within the system [3]. One prominent distance for Hankel matrices is the Frobenius-norm

$$\|\mathbf{H}\|_F = \sqrt{\operatorname{trace}(\mathbf{P})} \tag{2}$$

with the product $\mathbf{P} = \mathbf{H}^*\mathbf{H}$. Yet, to keep the capacity of the underlying linear subspace $\mathfrak{H}(\mathbf{H})$ regarding the previously mentioned affine transformations, frequently dynamic subspace angles based dissimilarities are used for matrix comparisons instead of the distance based on the Frobenius-norm [4].[1]

The geometric approach of tangential spaces assumes that all feature vectors of the data regarding a certain object belong to a manifold. Instead of feature vector comparisons one can investigate the similarities between those manifolds by means of tangent metrics [5]. Yet, the determination of the tangent metric requires the precise estimation of the tangent spaces of the data manifolds [6], which is usually a non-trivial task [7].

A promising alternative to these approaches is provided by application of the Grassmann manifold framework equipped with the respective Riemann geometry [8]. Here, several data vectors describing the same object are firstly collected into a single matrix. Each matrix constitutes a point in the manifold, which then can be compared in terms of distances regarding the underlying Riemann geometry.

In this paper we explain how the latter Grassmann manifold approach can be adopted for prototype based classification learning. Particularly, we consider the family of learning vector quantization networks (LVQ), originally proposed in [9]. Here we will focus on more modern variants like the generalized learning vector quantization (GLVQ,[10]) with gradient descent learning as well as their median and relational counterparts [11,12].

For this purpose, first we briefly review the GLVQ approaches. Thereafter, we explain the concept of Grassmann manifolds and respective geometries for robust data analysis. Particularly, we relate different metric concepts for Grassmann manifolds to the GLVQ variants.

2 Generalized Learning Vector Quantization

We start briefly describing the generalized learning vector quantization approach (GLVQ) as one of the most prominent, easy to interpret and robust classifiers. We assume data classes $1, \ldots, C$ and data $\mathbf{x} \in X \subseteq \mathbb{R}^n$. The aim of GLVQ is

[1] Let $\mathfrak{B}_\mathbf{X}$ and $\mathfrak{B}_\mathbf{Y}$ be orthonormal bases for the subspaces $\mathfrak{H}(\mathbf{X})$ and $\mathfrak{H}(\mathbf{Y})$ with the cardinalities $m(\mathbf{X})$ and $m(\mathbf{Y})$, respectively. Then the subspace angels $\theta_1, \ldots, \theta_m$ with $m = \min(m(\mathbf{X}), m(\mathbf{Y}))$ are recursively defined as

$$\theta_l = \max_{\mathfrak{x}_l \in \mathfrak{B}_\mathbf{X}} \max_{\mathfrak{y}_l \in \mathfrak{B}_\mathbf{Y}} \arccos\left(\left|\langle \mathfrak{x}_l, \mathfrak{y}_l \rangle_E\right|\right)$$

subject to the ortho-normalization restrictions $\langle \mathfrak{x}_j, \mathfrak{x}_l \rangle_E = \delta_{jl}$ and $\langle \mathfrak{y}_j, \mathfrak{y}_l \rangle_E = \delta_{jl}$ for $j = 1, \ldots, l-1$.

to distribute a set $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$ of prototype vectors such that we can assign a class label $c(\mathbf{x})$ to each data point $\mathbf{x} \in X$. Thereby, each prototype $\mathbf{w}_j$ is equipped with a class labels $c(\mathbf{w}_j)$ such that at least one prototype is responsible for each class. Then the class assignment $c(\mathbf{x}) = c\left(\mathbf{w}_{s(\mathbf{x})}\right)$ for a data sample $\mathbf{x}$ is realized by means of a winner-take-all competition (WTAC)

$$s(\mathbf{x}) = \text{argmin}_{j=1\ldots M}\left(d(\mathbf{x}, \mathbf{w}_j)\right) \tag{3}$$

where d is a given dissimilarity measure [2], frequently the squared Euclidean metric. We denote $\mathbf{w}_{s(\mathbf{x})}$ as the winner prototype of the competition. GLVQ takes as cost function

$$E_{GLVQ}(W) = \sum_{k=1}^{N} \varphi\left(\mu(\mathbf{x}_k, W)\right) \tag{4}$$

to be minimized during learning of labeled training data. It approximates the overall classification error [13]. Thereby, $\varphi(z)$ is a monotonically increasing function frequently chosen as the identity function $\text{id}(z) = z$ or the sigmoid function $\phi(z, \theta) = \frac{1}{1+\exp\left(\frac{z}{\theta}\right)}$. The function

$$\mu(\mathbf{x}, W) = \frac{d(\mathbf{x}, \mathbf{w}^+) - d(\mathbf{x}, \mathbf{w}^-)}{d(\mathbf{x}, \mathbf{w}^+) + d(\mathbf{x}, \mathbf{w}^-)} \tag{5}$$

is the so-called classifier function. Here, $\mathbf{w}^+$ is the best matching prototype regarding a training vector $\mathbf{x}$ with label $c(\mathbf{x})$ with the same class label whereas $\mathbf{w}^-$ denotes the best matching prototype of all prototypes of the other classes. Thus $\mu(\mathbf{x}, W) \in [-1, 1]$ takes negative values if $\mathbf{x}$ is correctly classified.

2.1 Stochastic Gradient Descent Learning in GLVQ

Learning in GLVQ often takes place as stochastic gradient descent learning (SGDL) with respect to the prototype vectors for E_{GLVQ} according to

$$\Delta\mathbf{w}^{\pm} \propto -\xi\left(\mathbf{x}, \mathbf{w}^{\pm}\right) \cdot \frac{\partial\mu}{\partial d^{\pm}(\mathbf{x})} \frac{\partial d^{\pm}(\mathbf{x})}{\partial \mathbf{w}^{\pm}} \tag{6}$$

requiring the differentiability of the dissimilarity measure d. The scaling factor

$$\xi\left(\mathbf{x}, \mathbf{w}^{\pm}\right) = \frac{\partial E(\mathbf{x}_k)}{\partial\varphi} \cdot \frac{\partial\varphi}{\partial\mu} \tag{7}$$

is obtained applying the chain rule for differentiation with the short hand notation $d^{\pm}(\mathbf{x}) = d(\mathbf{x}, \mathbf{w}^{\pm})$.

2.2 Median and Relational GLVQ

Median and relational GLVQ only require a given dissimilarity matrix $\mathbf{D}$ containing the dissimilarity values $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ between data. The matrix $\mathbf{D}$ is

said to be Euclidean embeddable if there exist a mapping $\tilde{\mathbf{x}} = \psi(\mathbf{x})$ such that $d_{ij} = d_E(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ is valid, whereby $d_E(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ is the Euclidean distance between $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$. A sufficient condition for the Euclidicity of $\mathbf{D}$ is that the corresponding similarity matrix $\mathbf{S}$ is positive semi-definite [14].

Median-GLVQ optimizes the GLVQ cost function (4) by an Expectation-Maximization approach [11]. Here the prototypes are restricted to be data objects. Yet, the optimization still works under weak assumptions regarding the matrix $\mathbf{D}$, particularly the Euclidicity may be violated. If the Euclidicity of $\mathbf{D}$ is valid, Relational GLVQ can be applied. In this approach the prototypes are assumed to be linear combination of the data, i.e. we have $\mathbf{w}_l = \sum_j \alpha_{lj}\mathbf{x}_j$ and the prototype update is realized as SGDL with respect to the coefficients α_{lj} [12].

3 Grassmann Manifolds, Riemann Metrics and Related Data Dissimilarities

The main assumption for the use of the Grassmann manifold concept is that the variations of an object/class c are reflected in the feature vectors $\mathbf{x}_j \in \mathbb{R}^n$ assigned to this class/object. Therefore, k randomly selected feature vectors all belonging to a certain class are collected into a so-called k-frame constituting a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ with $0 < k \leq n$ in general. However, in machine learning usually $k \ll n$ is chosen. The resulting matrix $\mathbf{X}$ generates a linear subspace $\mathfrak{H}(\mathbf{X})$ with dimensionality $m \leq k$.

The Grassmann manifold $\mathcal{G}_k^n$ equipped with the Riemann geometry is the space of all k-dimensional linear subspaces (hyperplanes) $\mathfrak{H}$ [8], i.e. a matrix $\mathbf{X}$ determines via $\mathfrak{H}(\mathbf{X})$ a certain point in the Grassmann manifold $\mathcal{G}_k^n$, see Fig. (1). Now we are able to compare object representations $\mathbf{X}$ and $\mathbf{Y}$ by a distance between their linear subspaces $\mathfrak{H}(\mathbf{X})$ and $\mathfrak{H}(\mathbf{Y})$ in the Grassmann manifold $\mathcal{G}_k^n$. For this purpose, several dissimilarity measures are known, most of them based on subspace angles (SSA) $\theta_1, \ldots, \theta_m$ between the subspaces with $m = \min(rank(\mathbf{X}), rank(\mathbf{Y}))$ [15]. Most common dissimilarities are the *geodesic distance* along the geodesic path in the manifold (see Fig.(1))

$$d_g(\mathfrak{H}(\mathbf{X}), \mathfrak{H}(\mathbf{Y})) = \sqrt{\sum_{j=1}^{k} \theta_j^2} \tag{8}$$

and the *chordal distance*

$$d_c(\mathfrak{H}(\mathbf{X}), \mathfrak{H}(\mathbf{Y})) = \sqrt{\sum_{j=1}^{k} \sin^2(\theta_j)} \tag{9}$$

as described [16]. Frequently, the simple approximation

$$d_g(\mathfrak{H}(\mathbf{X}), \mathfrak{H}(\mathbf{Y})) = \theta_1 \tag{10}$$

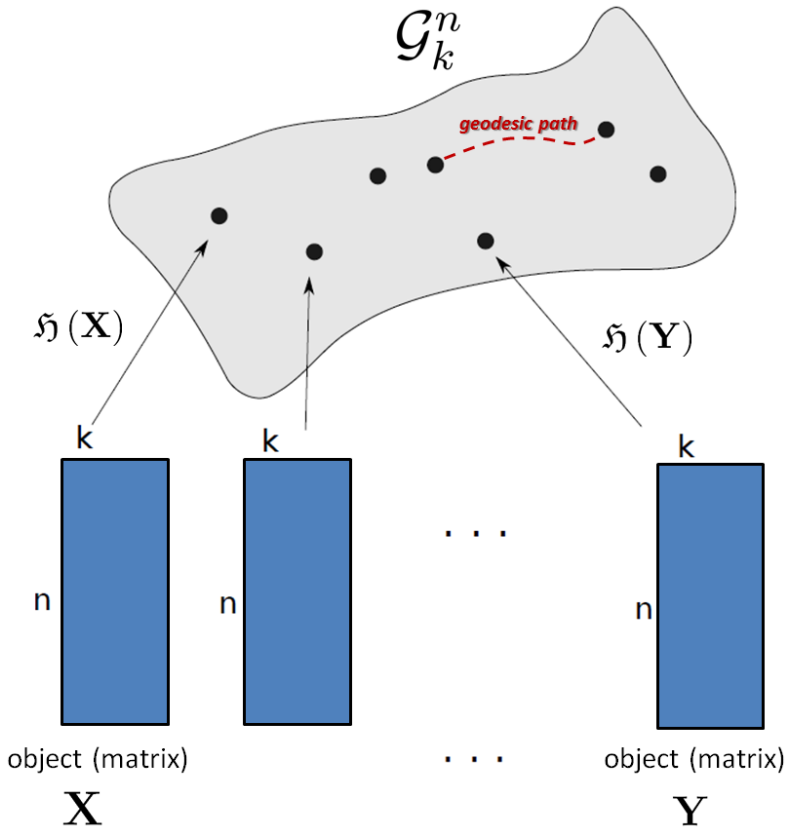


Fig. 1. Illustration of a Grassmann manifold $\mathcal{G}_k^n$. Several feature vectors of a class/object are collected in matrices of size $n \times k$ (usually with $k \ll n$), each of them generating linear subspaces $\mathfrak{H}$. Then, these matrices constitute points at the Grassmann manifold. Distances between points are measured in terms of manifold distances. The geodesic distance $d_g\left(\mathfrak{H}\left(\mathbf{X}\right), \mathfrak{H}\left(\mathbf{Y}\right)\right)$ from (8) is the path length along the geodesic path within the manifold.

is used instead of (8) or (9) due to the lower computational complexity [17]. However, as pointed out in [4], SSA dissimilarities demand a precise estimation of the subspaces $\mathfrak{H}(\mathbf{X})$ and $\mathfrak{H}(\mathbf{Y})$ to avoid misleading results or, equivalently, they assume representative matrices $\mathbf{X}$ and $\mathbf{Y}$. Yet, this requirement is contradictory to the assumption of noisy data. To overcome such problems, we can adopt robust dissimilarity measures developed for Hankel matrix comparisons keeping in mind that Hankel matrices represent linear dynamical systems by invariant subspaces. One prominent SSA surrogate dissimilarity measure is

$$\sigma_F(\mathbf{X},\mathbf{Y}) = 4 - \left\|\hat{\mathbf{X}} + \hat{\mathbf{Y}}\right\|_F^2 \tag{11}$$

suggested in [18], with $\hat{\mathbf{X}} = \frac{\mathbf{X}\mathbf{X}^T}{\|\mathbf{X}\mathbf{X}^T\|_F}$ and $\mathbf{Y}_2 = \frac{\mathbf{Y}\mathbf{Y}^T}{\|\mathbf{Y}\mathbf{Y}^T\|_F}$. Clearly, $\sigma_F(\mathbf{X},\mathbf{Y})$ is not longer a mathematical distance [19], but it remains to be a semi-metric [14,2]. Yet, in [20,21] it is argued that the similarity measure $s_F(\mathbf{X},\mathbf{Y}) = \left\|\hat{\mathbf{X}}^*\hat{\mathbf{Y}}\right\|_F^2$ is more robust than $\sigma_F(\mathbf{X},\mathbf{Y})$. It fulfills the inequality $0 \leq s_F(\mathbf{X},\mathbf{Y}) = \left|\left\langle\hat{\mathbf{X}},\hat{\mathbf{Y}}\right\rangle_F\right| \leq \left\|\hat{\mathbf{X}}\right\|_F \cdot \left\|\hat{\mathbf{Y}}\right\|_F$ according to the Cauchy-Schwarz-inequality where $\langle\mathbf{A},\mathbf{B}\rangle_F = \sum_i\sum_j \overline{a}_{ij}b_{ij}$ is the Frobenius inner product. Because both $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ are normalized matrices, we get $s_F(\mathbf{H}_1,\mathbf{H}_2) \leq 1$. Hence, the quantity

$$\delta_F(\mathbf{X},\mathbf{Y}) = 1 - \left|\left\langle\hat{\mathbf{X}},\hat{\mathbf{Y}}\right\rangle_F\right| \tag{12}$$

is a dissimilarity measure (semi-metric).

For the latter distance exists an isometrically embedding into the Euclidean space [17]. The geodesic distance realizes a non-Euclidean embedding. Both metrics can be seen also as examples to compare sets of vectors stored in the matrices $\mathbf{X}$ and $\mathbf{Y}$, i.e. they are particular realizations of a Hausdorff-metric [22].

4 Grassmann Manifold Dissimilarities for USE in GLVQ

For application of GLVQ to data classification by means of Grassmann manifolds, the input data are collected in k-frames, i.e. matrices $\mathbf{X} \in \mathbb{R}^{n\times k}$ with $0 < k \leq n$. The prototypes are also matrices $\mathbf{W}_j \in \mathbb{R}^{n\times k}$ as described before.

Doing so, all previously introduced dissimilarities (8) (12) can be used immediately in the Median-GLVQ. Further, the semi-metric $\sigma_F(\mathbf{X},\mathbf{W}_j)$ is differentiable with respect to the prototype matrix $\mathbf{W}_j$. However, the resulting complicate derivative expressions do not recommend an application in GLVQ due to the probable numerical instabilities. Otherwise, the δ_F measure is generally not differentiable because of the absolute value operator inside.

Although at first glance the SSA-based approaches seem to be inappropriate for SGDL in GLVQ variants, there are interesting options: For the chordal metric d_c from (9) exists an isometrically embedding into the Euclidean space [17]. Hence, the relational GLVQ is applicable with the prototypes here are linear

combinations of the k-frames, i.e. $\mathbf{W}_j = \sum_l \alpha_{jl}\mathbf{X}_l$ and SGDL is carried out with respect to the linear coefficients α_{jl}.

For the geodesic metric the things become more subtle but are still manageable. Particularly, the prototypes can be moved along the geodesic path as suggested in [17] for unsupervised vector quantization learning by self-organizing maps. Following this paper, the geodesic path $\mathbf{G}(\tau)$ between $\mathfrak{H}(\mathbf{X})$ for a given k-frame $\mathbf{X}$ with rank k and $\mathfrak{H}(\mathbf{W})$ for a prototype $\mathbf{W}$ with rank k in the Grassmann manifold $\mathcal{G}_k^n$ is given as

$$\mathbf{G}(\tau, \mathbf{X}, \mathbf{W}) = \mathbf{X} \cdot \mathbf{V} \cdot \cos(\boldsymbol{\Theta})\, t + \mathbf{U} \sin(\boldsymbol{\Theta})\, t \tag{13}$$

with $\mathbf{G}(0, \mathbf{X}, \mathbf{W}) = \mathbf{X}$ and $\mathbf{G}(1, \mathbf{X}, \mathbf{W}) = \mathbf{W}$ are valid . Here, the quantities $\mathbf{U}$, $\boldsymbol{\Theta}$, and $\mathbf{V}$ are obtained from the singular value decomposition

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V} = \left(\mathbf{I} - \mathbf{X} \cdot \mathbf{X}^T\right) \mathbf{W} \left(\mathbf{X}^T\mathbf{W}\right)^{-1} \tag{14}$$

together with $\boldsymbol{\Theta} = \tan(\boldsymbol{\Sigma})$. If $\mathbf{X}^T\mathbf{W}$ is not invertible, the pseudo-inverse can be applied in (14) for approximation. Further, if the matrices $\mathbf{X}$ and $\mathbf{W}$ do not have full rank k respective subspace representations for $\mathfrak{H}(\mathbf{X})$ and $\mathfrak{H}(\mathbf{W})$ have to be applied [23]. The prototype movement in this variant takes place as

$$\Delta\mathbf{W} \propto \mathbf{W} + \xi\left(\mathbf{X}, \mathbf{W}^{\pm}\right) \cdot \mathbf{G}(\varepsilon, \mathbf{X}, \mathbf{W}) \tag{15}$$

with $\xi(\mathbf{X}, \mathbf{W}^{\pm})$ playing the same scaling factor role as $\xi(\mathbf{x}, \mathbf{w}^{\pm})$ from (7) for standard GLVQ.

5 Conclusion

Data representation in terms of data clouds in Grassmann manifolds is a geometric approach to deal with noisy data. In this contribution we investigate, how the concept of Grassmann manifolds can be incorporated into the framework of learning vector quantization approaches for prototype based classification. Particularly, we discuss several distance and dissimilarity measures for Grassmann manifolds in the light of a possible application for the prototype based classifier.

References

[1] M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.

[2] D. Nebel, M. Kaden, A. Bohnsack, and T. Villmann. Types of (dis-)similarities and adaptive mixtures thereof for improved classification learning. *Neurocomputing*, page in press, 2017.

[3] D. Lai and G. Chen. Dynamical systems identification from time-series data: A Hankel matrix approach. *Mathematical and Computer Modelling*, 24(3):1–10, 1996.

[4] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.

[5] P. Simard, Y. LeCun, and J.S. Denker. Efficient pattern recognition using a new transformation distance. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 50–58. Morgan-Kaufmann, 1993.

[6] T. Hastie and P.Y. Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 13(1):54–65, 1998.

[7] D. Keysers, W. Macherey, H. Ney, and J. Dahmen. Adaptation in statistical pattern recognition using tangent vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):269–274, 2004.

[8] J. Hamm and D.D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 376–388, 2008.

[9] Teuvo Kohonen. Learning Vector Quantization. *Neural Networks*, 1(Supplement 1):303, 1988.

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

[11] D. Nebel, B. Hammer, K. Frohberg, and T. Villmann. Median variants of learning vector quantization for learning of dissimilarity data. *Neurocomputing*, 169:295–305, 2015.

[12] B. Hammer, D. Hofmann, F.-M. Schleif, and X. Zhu. Learning vector quantization for (dis-)similarities. *Neurocomputing*, 131:43–51, 2014.

[13] M. Kaden, M. Riedel, W. Hermann, and T. Villmann. Border-sensitive learning in generalized learning vector quantization: an alternative to support vector machines. *Soft Computing*, 19(9):2423–2434, 2015.

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

[15] P.A. Wedin. *On angles between subspaces of a finite dimensional inner product space*, pages 263–285. Number 973 in Lectur Notes in Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.

[16] S. Chepushtanova and M. Kirby. Sparse Grassmannian embeddings of hyperspectral data representations and classification. *IEEE Geoscience and Remote Sensing Letters*, 14(3):434–438, 2017.

[17] M. Kirby and C. Peterson. Visualizing data sets on the Grassmannian using self-organizing maps. In *Proceedings of the 12th Workshop on Self-Organizing Maps and Learning Vector Quantization (WSOM+ 2017), Nancy, France*, pages 32–37, Los Alamitos, 2017. IEEE Press.

[18] B. Li, O.I. Camps, and M. Sznaier. Cross-view activity recognition using Hankelets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, Providence (USA)*, pages 1362–1369, 2012.

[19] L.L. Presti, M. LaCascia, S. Sclaroff, and O. Camps. Hankelet-based dynamical systems modeling for 3D action recognition. *Image and Vision Computing*, 44:29–43, 2015.

[20] L.L. Presti and M. LaCascia. Ensemble of Hankel matrices for face emotion recognition. In V. Murino and E. Puppo, editors, *18th International Conference on Image Analysis and Processing, ICIAP 2015 Genoa (Italy)*, volume 9280 of *LNCS*, pages 586–597, 2015.

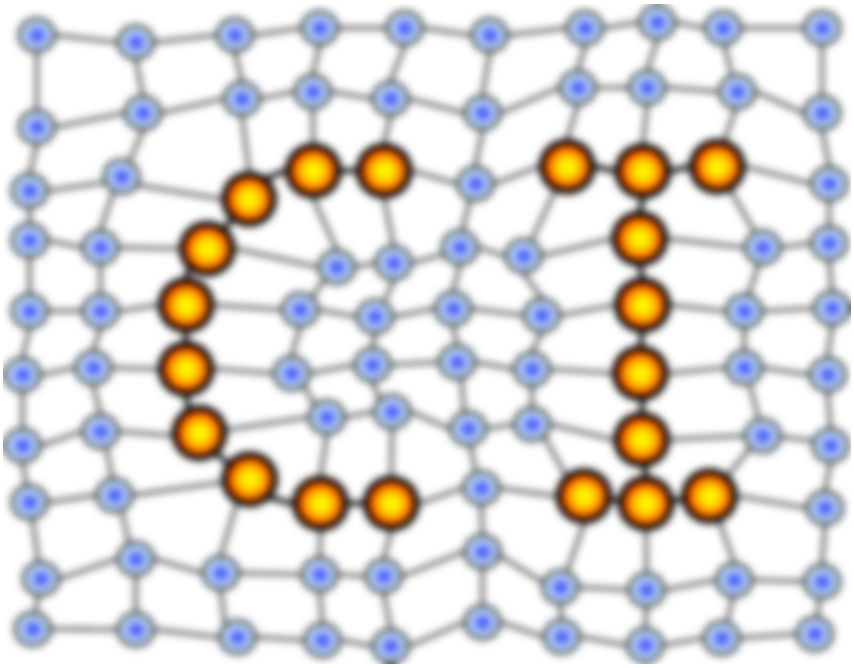
[21] L.L. Presti and M. LaCascia. Boosting Hankel matrices for face emotion recognition and pain detection. *Computer Vision and Image Understanding*, 156:19–33, 2017.

[22] S. Saralajew, D. Nebel, and T. Villmann. Adaptive Hausdorff distances and tangent distance adaptation for transformation invariant classification learning. In A. Hirose, editor, *Proceedings of the International Conference on Neural Information Processing (ICONIP) , Kyoto*, volume 9949 of *LNCS*, pages 362–371. Springer, 2016.

[23] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemann geometry of Grassmannian manifolds with a view on algorithmic computation. *Acta Applicacandae Mathematika*, 80(2):199–200, 2004.

MACHINE LEARNING REPORTS

Report 03/2017



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.