Parameter Sharing for Spatio-Temporal Process Models

Raphael Fischer, Nico Piatkowski, and Katharina Morik

TU Dortmund, AI Group, Dortmund, Germany
http://www-ai.cs.tu-dortmund.de

Abstract. While probabilistic models such as Markov random fields can be highly beneficial for spatio-temporal data, they often suffer from overfitting and have limited use in memory-constrained systems. We present a novel method to compress trained models based on temporal parameter sharing, which reduces redundancies in the parameters.

Keywords: Markov random field · parameter sharing · spatio-temporal data.

1 Introduction

Our world is constantly changing, and machine learning provides certain insight into these interesting procedures. Capturing states of a process at different spatial sites results in spatio-temporal datasets, and creates demand for accordingly tailored and theoretically well-based methods.

Spatio-temporal random field (STRF) models, an extension of the widely investigated *Markov random field*s (MRFs), allow to further analyze such processes with undirected graphical models. STRFs have been experimentally used in traffic routing [2] and network communication [4]. Moreover their use in resource-constrained environments has been extensively explored [6]. Possibly the biggest problem of STRFs is the high number of parameters, which increases quadratically with the data complexity [9,6]. In practice this often limits the model usability in terms of runtime and storage, and also makes them prone to overfitting. We therefore propose a novel approach to counter these issues by compressing previously trained STRFs. Our method uses parameter sharing, hence the compression does neither alter the data nor the underlying graphical structure.

Parameter sharing based on quantization was recently used for hard parameter tying in MRF training [1], which also tackled the problem of overfitting and results in less complex models. Our approach is similar, however we use a different quantization based on the spatio-temporal structure of the MRF. Similar methods have also been established for neural networks [10], with the aim of reducing the complexity.

2 Methodology

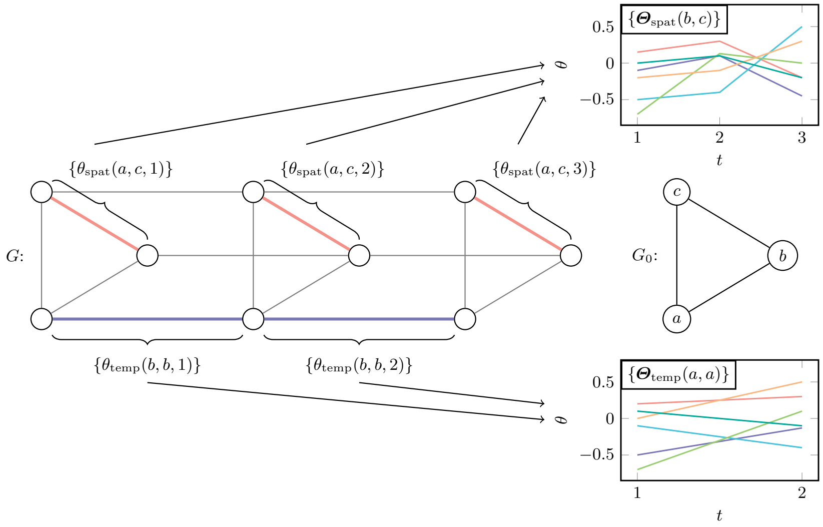
Spatio-Temporal Models A MRF represents a process in form of a multivariate random variable $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$, with dependencies between components being represented via the *conditional independence structure* (CIS) $G = (V, E)$. The concept of exponential families [9] allows parametrization via the edges of G, and optimal parameters can be determined with *maximum likelihood* estimation based on available process data. They resulting model can be queried to predict the most likely values for unobserved components of a partially given process state, i.e. to solve the *conditional maximum a posteriori* problem.

The STRF approach [7] extends the MRF framework for spatio-temporal use cases. STRF models are based on a spatial CIS G_0, which is replicated T times, i.e. the process is modeled for T different times. The replicas G_t are connected with temporal edges, the final model has temporal edges for each node and spatial edge. Such a structure allows temporal reparametrization for compressing and regularizing the parameters [6]. Our compression approach exploits the temporal dimension of STRFs further by merging parameters at different spatial sites.

The parameters of any STRF can be subdivided into spatial and temporal parameter groups, if the replica nodes in G share the same state space. The temporal edges are then represented by parameters $\theta_{\text{temp}}(v_i, v_j, t) \in \boldsymbol{\Theta}$ which describe temporal transitions from t to $t+1$ between replicas of nodes $v_i, v_j \in V_0$. Each parameter $\theta_{\text{spat}}(v_i, v_j, t) \in \boldsymbol{\Theta}$ of a spatial edge describes a state transition between replicas of $v_i, v_j \in V_0$ at time $1 \leq t \leq T$. Accordingly each parameter belongs to a set of parameters $\boldsymbol{\Theta}_{\text{temp}}(v_i, v_j)$ or $\boldsymbol{\Theta}_{\text{spat}}(v_i, v_j)$, which describe the same transition at different temporal sections of the model. This understanding is exemplary displayed in Figure 1. We call each set $\boldsymbol{\Theta}_{\text{spat}}(v_i, v_j)$ or $\boldsymbol{\Theta}_{\text{temp}}(v_i, v_j)$ a *parameter series.* We denote the number of different parameter series in the model with $s_t = |\boldsymbol{\Theta}_{\text{temp}}|$ and $s_s = |\boldsymbol{\Theta}_{\text{spat}}|$.

Compression of parameter series In most models the parameter series at different spatial sections will probably show analogies and thus redundancies. As an example, a street network model might represent crossings which behave similarly over time. We therefore propose to quantize the set of parameter series, meaning that each series $\boldsymbol{\Theta}_{\text{spat}}(v_i, v_j)$ or $\boldsymbol{\Theta}_{\text{temp}}(v_i, v_j)$ is replaced by a specified centroid $\tilde{\boldsymbol{\Theta}}_{\text{spat}}(v_i, v_j)$ or $\tilde{\boldsymbol{\Theta}}_{\text{temp}}(v_i, v_j)$. As parameter series are essentially T- or $(T-1)$-dimensional vectors, vector quantization methods such as *k-means* clustering [3] can be easily applied for finding good centroids. In a compressed model, several parameter series share the values of the corresponding centroid series, hence we name it *parameter series cluster sharing* (PSCS) compression.

The compression rate r determines the number of temporal and spatial cluster centroids c_t and c_s that need to be found, depending on the number of original series $r = \frac{s_t + s_s}{c_t + c_s}$. The space savings s can be computed with $s = 1 - \frac{c_t + c_s}{s_t + s_s}$ As an example, a model with $s_t = 1024$ different temporal parameter series and no spatial parameters, which are being replaced by $c_t = 128$ clustered centroids, would be compressed with $r = 1024/128 = 8$, i.e. a rate of 8:1, $1 - (128/1024) = 87.5\%$ of its original memory space could be saved.



$\{\boldsymbol{\Theta}_{\text{spat}}(b,c)\}$
0.5
0
−0.5
1
2
3
t
θ
$\{\theta_{\text{spat}}(a,c,1)\}$
$\{\theta_{\text{spat}}(a,c,2)\}$
$\{\theta_{\text{spat}}(a,c,3)\}$
G:
G_0:
c
b
a
$\{\theta_{\text{temp}}(b,b,1)\}$
$\{\theta_{\text{temp}}(b,b,2)\}$
θ
$\{\boldsymbol{\Theta}_{\text{temp}}(a,a)\}$
0.5
0
−0.5
1
2
t

Fig. 1. Example for a STRF G with spatial graph G_0 and $T = 3$, whose parameters can be interpreted as parameter series (diagonal edges are not shown).

It is important to understand that our approach does not modify the CIS, as compression only affects the parameters. This allows nodes in the model to behave similarly, but does not enforce them to act totally equal. PSCS drastically decreases memory requirements and thus facilitates the resource-constrained use of STRFs. It also generalizes the model, which might alleviate overfitting.

3 Experiments

Experimental Settings We chose the INSIGHT Dublin city data for our experiments [5,6]. It was collected in 2013 from Sydney Coordinated Adaptive Traffic System (SCATS) traffic sensors in Dublin, Ireland. Figure 2 depicts the placement of sensors in the street network of the city. The dataset features discretized speed measurements of 2367 SCATS sensors for $N = 134$ full days, with $T = 12$ measurements per day (every two hours). The spatial data dependencies are approximated with help of the Chow-Liu algorithm, i.e. G_0 is a tree with 2366 edges. The resulting STRF structure contains $12 \cdot 2366$ spatial and $11 \cdot (2366 \cdot 2 + 2367)$ temporal edges.

An extensive framework for probabilistic computations with *belief propagation* and *gradient descent* allowed us to work with STRF models[1], we slightly adadapted it to extract the parameter series. Python scripts were run to cluster the series with *scikit-learn*'s *MiniBatchKMeans* implementation[2] [8] (the value k is given by c_t and c_s).

[1] https://randomfields.org/px

[2] https://scikit-learn.org/

Table 1. Quantitative results of compressed models at different compression rates

Rate	$\|\boldsymbol{\Theta}_{\text{spat}}\|$	$\|\boldsymbol{\Theta}_{\text{temp}}\|$	Savings	A1 (± SD)	A2 (± SD)	A3 (± SD)
1:1	27796	84157	0%	77.46% (± 0.69)	52.79% (± 2.61)	77.54% (± 0.61)
1.25:1	22236	67324	20%	76.18% (± 0.64)	54.20% (± 1.86)	76.30% (± 0.62)
1.67:1	16677	50493	40%	76.24% (± 0.68)	54.14% (± 2.07)	76.26% (± 0.61)
2.5:1	11118	33662	60%	76.21% (± 0.67)	54.14% (± 2.24)	76.28% (± 0.62)
5:1	5559	16831	80%	76.28% (± 0.64)	53.79% (± 2.41)	76.32% (± 0.60)
100:1	277	841	99%	63.86% (± 0.21)	52.11% (± 0.52)	63.86% (± 0.26)

We prepared a 5-fold cross validation, i.e. 20% of the full data was used for testing in each split, providing averaged results and *standard deviation* (SD). As our experiment dataset is fully observed, we artificially concealed 50% of the test data. The STRF is used to predict the most likely values based on the remaining observations, and comparison with the originally measurements allows us to compute the prediction accuracy. We came up with three realistic scenarios why measurements might be missing: some SCATS sensors break down on single days (A1), only the beginning of the days are recorded (i.e. future prognosis) (A2), or all traffic sensors have random malfunctions (A3). Accordingly we prepared three different versions of the test data, and obtain the accuracy values A1-A3.

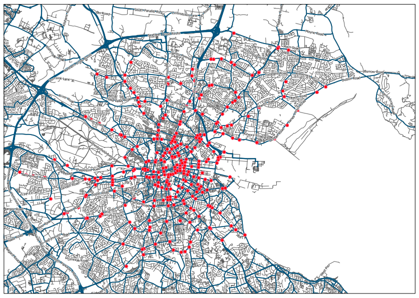


Fig. 2. Placement of traffic sensors.

Experimental Results We used PSCS for compressing the models to 80, 60, 40, 20, and 1% of their original size. The quality of the compressed models in terms of the prediction accuracy is shown in Table 1.

First thing to notice is that our future prognosis scenario (A2) has a significantly lower accuracy, for which the lack of local information is the most plausible explanation. Predicting data for full break down of some sensors scores a similar accuracy (A1) to predicting randomly missing data (A1). Probably the local spatial information is more important for predictions then temporally available data.

As expected A1 and A3 decrease with firmer compression (i.e. higher value of r), however the performance only significantly drops after a rate of 5:1. In the second scenario (A2) a compression is even able to increase the prediction accuracy, which might indicate that the original model slightly outfitted the training data. One can also see that compression increases the robustness of the model, as the SD is decreases with strong compression.

4 Conclusion

Our novel PSCS approach allows to compress spatio-temporal Markov models. The method eliminates parameter redundancies in the STRF, without affecting the CIS. Our experiments show that even drastically compressed models still perform well and are even able to outperform the original model, while requiring way less memory storage.

PSCS also holds potential for future work, which we want to shortly discuss here. Firstly one could incorporate a-priori knowledge of cluster assignments into the training routine. With adding a regularizing term one could enforce a soft temporal parameter series tying. By only training the centroid values (i.e. parameter sharing instead of regularization) it would also be possible to establish a hard tying of parameter series, which reduces the complexity during training.

Acknowledgement This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01|S18038A).

References

1. Chou, L., Sarkhel, S., Ruozzi, N., Gogate, V.: On parameter tying by quantization. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 3241–3247. AAAI'16, AAAI Press (2016)
2. Liebig, T., Piatkowski, N., Bockermann, C., Morik, K.: Dynamic route planning with real-time traffic predictions. Information Systems **64**, 258–265 (2017)
3. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Information Theory **28**(2), 129–136 (1982)
4. Michaelis, S., Piatkowski, N., Morik, K.: Predicting next network cell ids for moving users with discriminative and generative models. In: Mobile Data Challenge Workshop at Int. Conf. on Pervasive Computing. Newcastle, UK (June 2012)
5. Panagiotou, N., Zygouras, N., Katakis, I., Gunopulos, D., Zacheilas, N., Boutsis, I., Kalogeraki, V., Lynch, S., O'Brien, B., Kinane, D., Marecek, J., Yu, J.Y., Verago, R., Daly, E., Piatkowski, N., Liebig, T., Bockermann, C., Morik, K., Schnitzler, F., Weidlich, M., Gal, A., Mannor, S., Stange, H., Halft, W., Andrienko, G.L.: Insight: Dynamic traffic management using heterogeneous urban data. In: ECML/PKDD (2016)
6. Piatkowski, N.: Exponential families on resource-constrained systems. Ph.D. thesis, Technical University of Dortmund, Germany (2018)
7. Piatkowski, N., Lee, S., Morik, K.: Spatio-temporal random fields: compressible representation and distributed estimation. Machine Learning **93**(1), 115–139 (2013)
8. Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th International Conference on World Wide Web. pp. 1177–1178. WWW '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1772690.1772862
9. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning **1**(1-2), 1–305 (2008)
10. Yang, Y., Ruozzi, N., Gogate, V.: Scalable neural network compression and pruning using hard clustering and l1 regularization. CoRR **abs/1806.05355** (2018)