A Short Note on Analyzing Sequence Complexity in Trajectory Prediction Benchmarks

Ronny Hug[1], Stefan Becker[1], Wolfgang Hübner[1] and Michael Arens[1]

Abstract— The analysis and quantification of sequence complexity is an open problem frequently encountered when defining trajectory prediction benchmarks. In order to enable a more informative assembly of a data basis, an approach for determining a dataset representation in terms of a small set of distinguishable prototypical sub-sequences is proposed. The approach employs a sequence alignment followed by a learning vector quantization (LVQ) stage. A first proof of concept on synthetically generated and real-world datasets shows the viability of the approach.

I. INTRODUCTION

An open question frequently encountered in the context of trajectory prediction is the quantification of dataset complexity, leading to hard-to-solve or too simple benchmarks. Current attempts in standardized benchmarking, e.g. *TrajNet++* as mentioned in [1], originate from heuristics or experience-based criteria when assembling the data basis.

Dissecting trajectory datasets, each dataset can be reduced to a small number of prototypical sub-sequences specifying distinct motion patterns, where each sample can be assumed to be a variation of these prototypes. Following this assumption, the quantification of dataset complexity can be based on the number and variation of prototypes. Towards this end, an approach for determining prototypes from a dataset is proposed. The approach applies an alignment step followed by vector quantization for clustering aligned samples. Experiments are conducted on a synthetically generated dataset and an exemplary standard benchmarking dataset in order to provide a first proof of concept. Lastly, potential uses of the proposed approach are discussed.

II. SEQUENCE ALIGNMENT

Given a set of trajectories (*samples*) $\mathcal{X} = \{X_1, ..., X_N\}$, as sequences of M subsequent points $X_i = \{\mathbf{x}_1^i, ..., \mathbf{x}_M^i\}$, each sample is first normalized by moving it into a self-centered reference frame and scaling it to unit length:

$$X_i^{norm} = \left\{ \frac{\mathbf{x}_j^i - \bar{\mathbf{x}}}{\mathbf{x}_M^i - \mathbf{x}_1^i} | j \in [1, .., M] \right\}. \qquad (1)$$

It has to be noted that this normalization solely serves the purpose of moving all samples into a common value range and is not a good normalization in terms of pooling similar samples. Then, all samples are aligned with a single prototype $\hat{Y} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_M\}$ by using affine

[1]Fraunhofer IOSB, Ettlingen, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, 76275 Ettlingen, Germany `first.lastname@iosb.fraunhofer.de`

Fraunhofer IOSB is a member of the Fraunhofer Center for Machine Learning.

transformations, which are retrieved from a regression model $\phi : X \to \{\mathbf{t}, \alpha, s\}$. $\hat{Y}$ and ϕ are learned by minimizing the mean squared error between each sample $\phi(X_i^{norm})$ and the prototype $\hat{Y}$

$$\mathcal{L}_{DA}(\phi(X_i^{norm}), \hat{Y}) = \frac{1}{M} \sum_{j=1}^{M} \|\mathbf{x}_j^i - \hat{\mathbf{y}}_j\|_2^2 \qquad (2)$$

using stochastic gradient descent. This is different to linear factor models, where the whole training set has to be considered. With respect to equation 2 and the affine transformation, the trivial solution that maps all samples onto the origin has to be avoided. A brute force approach to this problem is to enforce a minimum scale. These steps result in a minimum variance alignment of all samples with respect to the learned prototype. Further, by learning the prototype and the transformation concurrently, the prototype adapts to the most dominant motion pattern and the normalized data is aligned accordingly.

III. LEARNING VECTOR QUANTIZATION

Clustering approaches can be applied after alignment, since the alignment approach distributes random errors homogeneously over the sequence. For clustering, a vector quantization (LVQ) approach is proposed, which is inspired by [2] and can directly be integrated in a deep learning framework. Here, aligned samples are mapped onto K prototypes $\mathcal{Z} = \{Z_1, ..., Z_K\}$, with $Z_i = \{\mathbf{z}_1^i, ..., \mathbf{z}_M^i\}$, in quantized space. This results in a concise set of prototypes representing the given dataset. The prototypes are learned by minimizing the mean squared error between the aligned samples and the respective closest prototype in quantized space:

$$\mathcal{L}_{LVQ} = \frac{1}{M} \sum_{j=1}^{M} \|\mathbf{x}_j - z_q(j)\|_2^2 + \gamma \cdot L_{global},$$

$$\text{with}$$
$$z_q(j) = \mathbf{z}_{argmin_k \|\mathbf{x}_j - \mathbf{z}_k\|_2^2}, \qquad (3)$$
$$L_{global} = \frac{1}{N \cdot M \cdot K} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \|\mathbf{x}_j^i - \mathbf{z}_j^k\|_2^2.$$

In addition, a regularization term L_{global}, calculating the global error across all prototypes and all samples, is employed for driving all prototypes towards the prototype of the dominant motion pattern. Therefore, the regularization is used for capturing information from less frequent samples. Due to the fact that, it is unknown how many prototypes are necessary, unnecessary prototypes can be identified by their

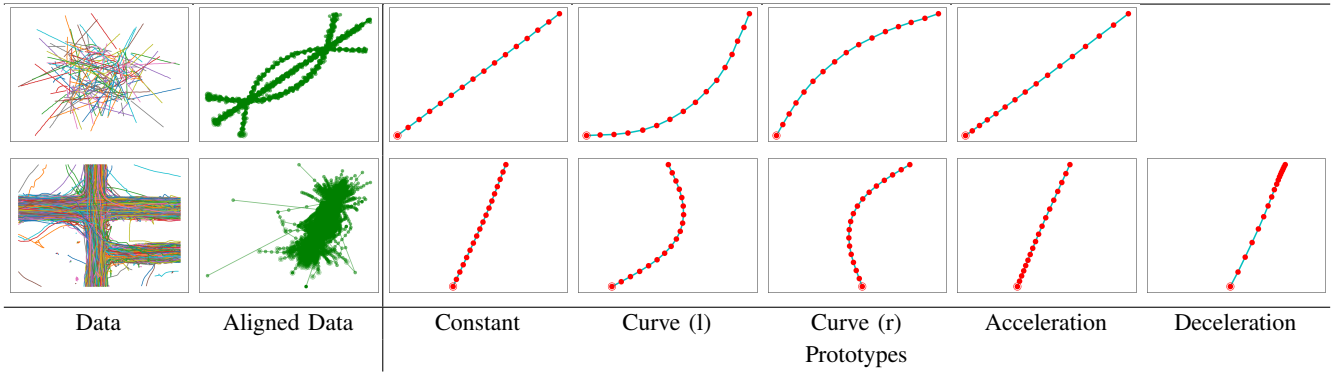| Data	Aligned Data	Constant	Curve (l)	Curve (r)	Acceleration	Deceleration

Prototypes

Fig. 1. Data, aligned data and learned prototypes for the synthetic dataset SGTD (top) and the SDD-hyang dataset (bottom). For clarity only every 2nd or 3rd trajectory point is plotted for SGTD and SDD-hyang prototypes respectively.

similarity to the dominant prototype and thus be sorted out later, in case K is too large.

Initialization. Due to the winner takes all strategy, $\mathcal{L}_{LVQ}$ only updates prototypes that have supporting samples. While the regularization term tries to account for this, initializing the K prototypes appropriately helps improving the quantization results. Thus, a Forgy initialization is applied, i.e. randomly selected samples from the dataset are used as initial prototypes [3]. This follows common practice of other vector quantization approaches, like for example K-means [4].

Refinement. A heuristic refinement scheme is employed in order to remove unnecessary prototypes when K was too large. Starting with the prototype which is supported by most training samples, additional prototypes are added based on their dissimilarity to previously added prototypes and their impact on the global error given the refined set of prototypes. By choosing prototypes that maximize the global error, a wider coverage of motion patterns is ensured. Thus, unnecessary prototypes, especially those similar to the most dominant prototype, can be sorted out.

IV. QUALITATIVE EVALUATION

As a first proof of concept, a qualitative evaluation is conducted on a synthetically generated dataset (*SGTD*) and the combined *hyang* scenes [5] taken from the *stanford drone dataset* (SDD, [6]). SGTD consists of 200 randomly distributed trajectories consisting of $M = 31$ points[1], with random orientations and different normally distributed speed profiles. It covers 4 motion patterns: constant (125 samples), accelerated (33 samples), and curvilinear (29 (left) and 13 (right) samples; direction change by $\alpha \sim \mathcal{U}(80, 90)$ degrees in either direction). For SDD-hyang, the rate has been reduced to 6 samples per second. This achieves a more noticeable difference between subsequent trajectory points. Further, all trajectory segments containing 54 points[2] are considered for training. Since this is a real world dataset, the generating features are unknown. Yet the occurrence of all basic motion patterns (constant, acceleration, decelerated

and curvilinear motion) is expected reflecting its complexity. For both datasets, $K = 10$ prototypes are used, providing sufficient capacity to capture all relevant features while also allowing to test the proposed regularization and refinement strategies.

The results of the alignment and the learned prototypes are depicted in figure 1. For SGTD (first row), it can be seen that all samples are aligned such that they follow a common mean orientation and similar samples are pooled together. All 4 prototypes necessary for modeling the dataset were found with correct association between data and prototypes. For SDD-hyang, there is not an as clear separation between different motion patterns. However, a similar behavior can be observed when looking at the aligned data. Further, as expected, the quantization yields an additional prototype corresponding to a decelerated motion.

V. DISCUSSION AND FUTURE WORK

At last, given the results in section IV, future work building on the proposed approach are discussed.

A. Complexity Analysis of Standard Benchmarking Datasets

A thorough complexity analysis of standard benchmarking datasets for trajectory prediction, building on the proposed approach should be considered. Analyzing the complexity of these datasets could offer more insight on their viability for evaluating certain aspects of machine learning models.

B. Deriving a Benchmark from Insights

Insights gained from dataset complexity analysis could be used for deriving a new benchmark for trajectory prediction, that specifically targets different aspects of a given model. This could provide a viable complementary benchmark to the recently published *TrajNet++* benchmark (mentioned in [1]; successor of *TrajNet* [7]).

C. Data Preprocessing

The proposed approach may also serve as universal preprocessing stage for trajectory prediction models. Thus an experimental study should be conducted for comparing preprocessing approaches.

---

[1]This number is arbitrary and does not affect the viability of the approach.
[2]Corresponds to the first quartile with respect to the number of trajectory points for all trajectories

REFERENCES

[1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *arXiv preprint arXiv:1905.06113*, 2019.

[2] A. van den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[3] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern recognition letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

[4] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[5] R. Hug, S. Becker, W. Hübner, and M. Arens, "On the reliability of lstm-mdl models for pedestrian trajectory prediction," *VIIth International Workshop on Representation, analysis and recognition of shape and motion FroM Image data (RFMI 2017)*, 2017.

[6] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision*. Springer, 2016, pp. 549–565.

[7] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, "Trajnet: Towards a benchmark for human trajectory prediction," *arXiv preprint*, 2018.