THIS IS A TEMPLATE/EXAMPLE FOR E4040 Project REPORT Updated 10/3/2021
CONSULT WITH additional instructions given in a document called “E4040.*.ProjectInstructions”
Based on IEEE Conference format (https://www.ieee.org/conferences/publishing/templates.html)

GAN for Uncertainty Quantification of Human Trajectories Prediction

E4040.2021Fall.SGAN.report
Zhaobin Mo zm2302, Yongjie Fu yf2578
Columbia University

Abstract

Uncertainty Quantification (UQ) of the human driver car-following (CF) behavior is crucial for reliable and robust prediction, given various sources of uncertainty in human driving behaviors. In this paper, we aim to characterize the uncertainty in the CF behavior using the generative neural network. We apply our model to a real-world dataset, the Next Generation SIMulation (NGSIM) dataset to demonstrate the superiority of our model.

1. Introduction

Characterizing car-following (CF) behavior is crucial to human-aware and social-aware motion planning of autonomous vehicles.

CF models mainly bifurcate into two categories: the physics-based [1] and the data-driven [2] models. Physics-based models approximate the CF behavior based on known physics laws or empirical rules. Those models are easy to interpret, while they may fail to learn the complex human behavior due to ideal models assumptions. Data-driven models can learn underlying patterns directly from data without prior knowledge or assumptions. However, this type of model may not produce interpretable and physically-consistent results, and is also data-hungry.

The main challenge for modeling the CF behavior lies in various sources of uncertainty, including internal uncertainty like driver heterogeneity and external uncertainty like measurement noise. Uncertainty Quantification (UQ) aims to characterize the CF behavior in a stochastic manner. For physics-based models, the main UQ method is Bayesian approximation [3]. Data-driven UQ methods include Bayesian approximation [4], ensemble methods [5], and generative models like the variational autoencoder [6] and generative adversarial networks (GANs) [7].

2. Summary of the Original Paper

2.1 Methodology of the Original Paper

The original paper [8] applies the GAN model to predict the pedestrian trajectory. The main contribution of this paper is that it proposes “social pooling” to consider the hidden states of neighbouring pedestrians when making predictions.

2.2 Key Results of the Original Paper

Key results of original paper:

1) Lower estimation error compared to baselines.
2) Ablation study: pooling v.s. without pooling. GAN model with “social pooling” outperforms the one without.

3. Methodology (of the Students’ Project)

We have made modifications to the original paper for two reasons. First, the original paper is too complex, which contains the GAN structure and “social-pooling”, and we focus on the GAN structure, which is the main framework of the original paper. Second, the original paper is about pedestrians trajectory prediction, while we are with a transportation background and thus we change to apply the proposed method to the car-following problem. We also explained this modification in the proposal. Following that we enumerate the difference between the original paper and our implementation as follows:

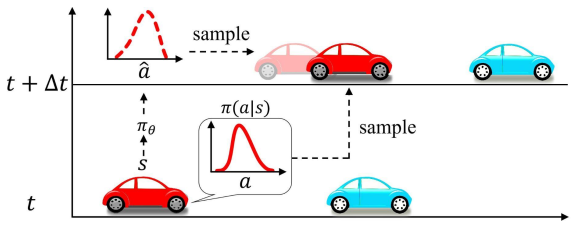
1. We change the application field from pedestrian trajectory prediction to human driving behavior prediction, which is a car-following modeling problem.
2. We partially implemented the GAN model without “social pooling” due to its complexity.

3.1. Objectives and Technical Challenges

The objective of this project is to use a GAN model to predict the driver’s action according to the current driving states, and tune the model to maintain a low error. Technical challenges are as follows:

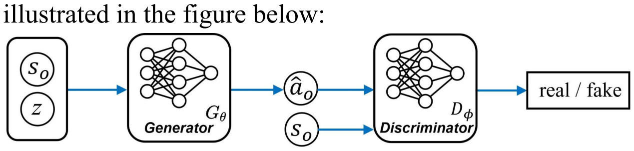
1. GAN model is hard to train and suffers from “mode collapse”. To mitigate this issue, we add a posterior estimator to penalize the spike prediction distribution.
2. There is a huge oscillation in human driver’s accelerations. Thus we apply data smoothing and predict the target velocity instead.

3.2. Problem Formulation and Design Description



Problem statement. In the figure above, a red car is following a blue car along the horizontal axis, and the vertical axis is time. It is assumed that a driver obeys an underlying stochastic policy that maps from driving states to a distribution over actions. A CF model learns a surrogate policy that approximates the ground-truth policy. At the current time step, the red car samples its action given its current state, which leads to the true position (solid red car) at the next time step. Meanwhile, a surrogate model predicts the action distribution and samples an action, which leads to the estimated position (transparent red car) at the next time step. The key problem is to quantify the uncertainty of prediction and its discrepancy with regard to the true action.

GAN model for CF problem. The GAN model is illustrated in the figure below:



The loss functions for the generator and the discriminator are depicted as below:

$$\mathcal{L}_G(\theta,\lambda) = \alpha \mathbb{E}_{q(s_o)p(z)}\left[D_\phi(s_o,\hat{a}_o)\right]$$

$$\simeq \frac{\alpha}{N_o}\sum_{i=1}^{N_o} D_\phi(s_o^{(i)},\hat{a}_o^{(i)}),$$

$$\mathcal{L}_D(\phi) = -\mathbb{E}_{q(s_o)p(z)}\left[\log D_\phi(s_o,\hat{a}_o)\right]$$

$$-\,\mathbb{E}_{q(s_o,a_o)}\left[\log(1-D(s_o,a_o))\right]$$

$$\simeq -\frac{1}{N_o}\sum_{i=1}^{N_o} \log D_\phi(s_o^{(i)},\hat{a}_o^{(i)}) + \log(1-D_\phi(s_o^{(i)},a_o^{(i)})),$$

A posterior estimator is employed to mitigate the mode collapse of GANs. The basic idea is to use a posterior estimator, which is an additional neural network, to learn the mapping from the generated state-action pair to the posterior probability of the latent variable. A *reconstruction error* is defined as the expectation of the negative log-likelihood and is added to the loss function of the generator. After adopting the posterior estimator technique, the loss function of the generator is depicted as

$$\mathcal{L}_G^{new}(\theta,\lambda) = \mathcal{L}_G(\theta,\lambda) - m_o\mathbb{E}_{q(s_o)p(z)}[\log Q_\xi(s_o,\hat{a}_o)],$$

The Adam optimizer is used to update the parameter by minimizing the corresponding loss functions.

Algorithm 1: Training process of GAN for CF modeling.

Initialization:
Initialized networks parameters θ^0, ϕ^0.
Epochs $epochs$; Batch size m.
Learning rate lr;
Require: Adam optimizer
Input: data $\{(s_o^{(i)},a_o^{(i)})\}_{i=1}^{N_o}$

for $iter \in \{1,\ldots,epochs\}$ **do**
  Sample batches $\{(s_o^{(i)},a_o^{(i)})\}_{i=1}^{m}$
  Calculate $\mathcal{L}_D$
  $\phi \leftarrow \phi - lr \cdot \text{Adam}(\phi, \nabla_\phi \mathcal{L}_D)$
  Calculate $\mathcal{L}_G$
  Calculate $\mathcal{L}_G^{new}$
  $\theta \leftarrow \theta - lr \cdot \text{Adam}(\theta, \nabla_\theta \mathcal{L}_G^{new})$
end for

The code for this project is available at: https://github.com/ecbme4040/e4040-2021fall-project-sgan-zm2302-yf2578

4. Implementation

4.1 Data

NGSIM dataset [9] is an open dataset that collects vehicle trajectories every 0.1 seconds. We focus on US Highway 101.

4.1 Deep Learning Network

Introduced in the previous section.

4.2 Software Design

Apart from the aforementioned structure and training algorithm, we introduce the baselines and metrics we used, together with the configuration details. In addition, to mitigate the mode collapse, we also add a posterior estimator to the original GAN structure.

Evaluation metrics. We use two metrics to measure the deviation of every test point from the mean of our model prediction: Root Mean Square Error (**RMSE**) and Mean Absolute Error (**MAE**).
In addition, we use two metrics to evaluate the difference between the prediction distribution and the sample distribution: Kullback–Leibler (**KL**) divergence and Negative Log Predictive Density (**NLPD**) computed using Parzen window. As the sample distribution cannot be computed explicitly, we approximate the KL divergence using a probabilistic classification technique

To mitigate the randomness in both training and test procedures, all models are trained for 3 rounds, and each trained model is evaluated 10 times with new test data. The mean and standard deviation of each metric is recorded.

Baseline. We use the Monte Carlo (MC)-Dropout model, denoted as **NN-Drop**, as our baseline. This is a very simple but practical model for UQ. The idea of NN-Drop is to maintain dropout in the test procedure to enable stochastic prediction.

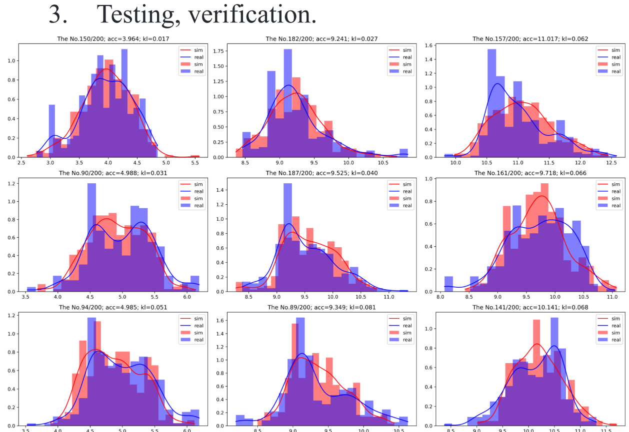
Parameter setting. The generator, discriminator, and posterior estimator share the same network complexity, which consists of 4 layers with 20 neurons in each layer. An He uniform initializer [10] is used. GAN models are trained using an Adam optimizer with a learning rate of 0.001 and other hyperparameters as default. Each model is trained for 1000 epochs, and the batch size is 128. Sizes of the training data is 500. We randomly generate another 200 data as test data.

5. Results

5.1 Project Results

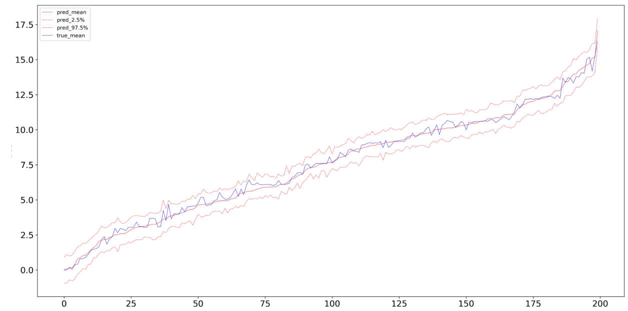
The distribution of the prediction compared to the ground-truth is illustrated in figure x.

1. Description of results,
2. Figures, plots,
3. Testing, verification.



In the figure, the x-axis is the target velocity and the y-axis is its probability density. The red and blue curves are the results of the ground-truth and prediction, respectively.

We also draw the 95% *prediction interval*, which is depicted in the figure below. The x-axis is the sorted index of data, y-axis is the velocity. The blue solid line is the ground-truth. The red solid line is the mean of the prediction, and the red dashed lines are the 95% prediction bounds. It shows that ground-truth falls in the prediction bounds, and thus demonstrates that the proposed model can model the CF behavior considering uncertainty.



5.2 Comparison of the Results Between the Original Paper and Students' Project

The results are shown in the following.

Prediction error. Errors of four metrics of the GAN and the NN-Drop models are shown in the table below, where our model outperforms the NN-Drop.

Model	RMSE	MAE	KL	NLPD
NN-Drop	1.287 ± 0.023	0.989 ± 0.018	0.782 ± 0.012	1.454 ± 0.019
GAN(ours)	0.764 ± 0.018	0.574 ± 0.012	0.409 ± 0.091	0.794 ± 0.050

Computation time. We also compare the computation time of the proposed Integrated PhysGAN with all other models. The results are shown in the figure below. The column ``Networks" records the number and type of neural networks for each model. ``NN" stands for neural networks that take the state s as the sole input. ``G", ``D" and ``Q" stand for generators, discriminators, and posterior estimators, respectively. Note that although the proposed model is more time consuming, in this case study the total training time is about 10 minutes, which is a medium value.

Model	Networks	Computation Time	
		Training (ms/epoch)	Inference (ms)
NN-Drop	1 NN	4.59	0.16
GAN	1 G + 1 D + 1Q	8.79	0.12

5.3 Discussion of Insights Gained

From the distribution results, we can see that the distribution of the human behavior is not Gaussian, and it may not be single-mode. While the GAN can still learn the pattern. This is what the physics-based model may fail to do because this kind of model usually makes assumptions toward the noise distribution.

6. Future Work

Incorporate the “social pooling” into this work. Explore new training algorithm the fasten the training speed

7. Conclusion

This project applis the GAN model to the uncertainty quantification (UQ) of the human car-following (CF) behavior. Our proposed model has been demonstrated to be able to learn the CF behavior from the real-world data, and the superiority is demonstrated by comparing it to the baseline model..

7. References

[1] Chen, Chenyi, et al. "Calibration of MITSIM and IDM car-following model based on NGSIM trajectory datasets." Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety. IEEE, 2010.
[2] Zhu, Meixin, Xuesong Wang, and Yinhai Wang. "Human-like autonomous car-following model with deep reinforcement learning." Transportation research part C: emerging technologies 97 (2018): 348-368.
[3] van Hinsbergen, Chris P. IJ, et al. "Bayesian calibration of car-following models." IFAC Proceedings Volumes 42.15 (2009): 91-97.
[4] Jospin, Laurent Valentin, et al. "Hands-on Bayesian Neural Networks--a Tutorial for Deep Learning Users." arXiv preprint arXiv:2007.06823 (2020).
[5] Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." arXiv preprint arXiv:1612.01474 (2016).
[6] Böhm, Vanessa, François Lanusse, and Uroš Seljak. "Uncertainty quantification with generative models." arXiv preprint arXiv:1910.10046 (2019).
[7] Silva, Vinicius LS, Claire E. Heaney, and Christopher C. Pain. "GAN for time series prediction, data assimilation and uncertainty quantification." arXiv preprint arXiv:2105.13859 (2021)
[8] Gupta, Agrim, et al. "Social gan: Socially acceptable trajectories with generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
[9] Punzo, Vincenzo, Maria Teresa Borzacchiello, and Biagio Ciuffo. "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data." Transportation Research Part C: Emerging Technologies 19.6 (2011): 1243-1262.
[10] He, Kaiming, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision. 2015.

8. Appendix

8.1 Individual Student Contributions in Fractions

	zm2302	yf2578	
Last Name	Mo	Fu	
Fraction of (useful) total contribution	1/2	1/2	
What I did 1	Coding	Coding	
What I did 2	Write report	Write report	