Scenario Generation and Comparison for the Assessment of Automated Vehicles

Erwin de Gelder, Jasper Hof, Eric Cator, Jan-Pieter Paardekooper,
Olaf Op den Camp, Bart De Schutter

1

1

Abstract

The development of assessment methods for the performance of Automated Vehicles (AVs) is essential to enable to deployment of automated driving technologies, due to the complex operational domain of AVs. One candidate is scenario-based assessment in which test cases are derived from real-world road traffic scenarios obtained from driving data.

Our contribution is twofold. First, we propose a method to generate scenarios for use in test case descriptions for the assessment of AVs. The generated scenarios are based on real-world scenarios. Using a singular value decomposition, a reduction of parameters is obtained, such that the remaining parameters describe as much of the variations found in the real-world scenario as possible. Second, we propose a metric based on the Wasserstein distance that quantifies to what extend the generated scenarios are representative of real-world scenarios while covering the actual variety found in the real-world scenarios.

We illustrate our proposed method by generating scenarios with a lead vehicle braking and cut-in scenarios. Our proposed metric is used to determine the appropriate number of parameters to be selected.

2

2

Outline

1. Introduction
2. Scenario generation
3. Scenario comparison
4. Results
5. Discussion
6. Conclusions

Introduction

- Assessment of Automated Vehicles (AVs) is important to not delay the development of AVs.
- One approach is to use test cases based on scenarios captured from real-world data
- Different techniques have been proposed, e.g., Lages et al. (2013), Zofka et al. (2015), de Gelder & Paardekooper (2017), Schuldt et al. (2018), Zhao et al. (2018), Thal et al. (2020)
- Problems of existing methods:
  - Replaying observed scenarios, i.e., no variations
  - Over simplification of scenario, e.g., velocity profile follows a sinusoidal function.
  - No probability density function known – as a result, no evaluation can be made of the performance of the system once deployed on the real road.

Introduction (cont’d)

- For the generated scenarios, it is important that they
  - are representative of scenarios that could happen in real life
  - cover the same variety that is found in real life
- We propose a method for generating scenarios that do not have the aforementioned problems
- We propose a metric that quantifies to what extend the generated scenarios are representative and cover the actual variety.

5

5

Scenario generation – intro

- To generate realistic scenarios, we use a data-driven approach; observed scenarios are used to generate new scenarios.
- Approach:
  - Parametrize scenarios
  - Fit a probability distribution function (pdf) on the parameters of the observed scenarios
  - Generate parameters by drawing samples from that pdf
  - Create scenario from the generated parameters

6

6

Scenario generation – intro (cont'd)

- Choosing the parameters that describe a scenario is not trivial:
  - Many parameters might be needed to correctly describe all possible varieties in a scenario.
  - Too many parameters lead to problems with estimating the pdf (curse of dimensionality)
- Solution: Take all important parameters and perform a reduction using Singular Value Decomposition (SVD)

Scenario generation – parametrization

- We assume that a scenario can be described using a time series $y(t) \in \mathbb{R}^{n_y}$ with $t \in [0, T]$ and some additional parameters $\theta \in \mathbb{R}^{n_\theta}$
- The time series $y(t)$ is converted to parameters: $y(0), y\left(\frac{T}{n_t}\right), y\left(\frac{2T}{n_t}\right), \dots, y(T)$
- Let $\mathbf{y} = \left[y^T(0) \quad y^T\left(\frac{T}{n_t}\right) \quad \dots \quad y^T(T)\right]^T \in \mathbb{R}^{(n_t+1)n_y}$, the a scenario is represented by $(\boldsymbol{y}, T, \theta)$.
- Scenario i is represented by $x_i = [\boldsymbol{y}_i^T \quad T_i \quad \theta_i^T]^T \in \mathbb{R}^{n_x}$ with $n_x = (n_t + 1)n_y + n_\theta + 1$.

Scenario generation – SVD

- Let $\alpha \in \mathbb{R}^{n_x}$ a weighting factor and suppose that we have N scenarios and let $X = [\alpha x_1 - \mu \quad \cdots \quad \alpha x_N - \mu] \in \mathbb{R}^{n_x \times N}$ with $\mu = \frac{1}{N}\sum_{i=1}^{N} \alpha x_i$.
- SVD: $X = U\Sigma V^T$. $U \in \mathbb{R}^{n_x \times n_x}$ and $V \in \mathbb{R}^{N \times N}$ are orthonormal matrices. Σ takes same shape as X and has only zeros except at the diagonal; the (j,j)-th element is σ_j, $j \in \{1, \dots, \bar{N}\}$, $\bar{N} = \min(N, n_x)$, such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\bar{N}} \geq 0$.
- Because the singular values (σ) are in decreasing order, we can approximate x_i by setting $\sigma_j = 0$ for $j > d$:

$$\alpha x_i = \mu + \sum_{j=1}^{\bar{N}} \sigma_j v_{ij} u_j \approx \mu + \sum_{j=1}^{d} \sigma_j v_{ij} u_j \,.$$

Scenario generation – KDE

- Using the approximation based on the SVD, the i-th scenario is described by the vector $\tilde{v}_i^T = [v_{i1} \quad \cdots \quad v_{id}]$.
- To not rely on a functional form of the probability distribution and to also model dependencies between the different parameters, KDE is used:

$$\hat{f}_H(v) = \frac{1}{n}\sum_{i=1}^{N} K_H(v - \tilde{v}_i)\,.$$

- H is the bandwidth matrix, we use a Gaussian kernel:

$$K_H(u) = |H|^{-1/2} K\left(H^{-1/2}u\right), K(u) = (2\pi)^{-d/2} \exp\left\{-\frac{1}{2}u^T u\right\}.$$

- Sampling from $\hat{f}_H$ is straightforward: Randomly choose $i \in \{1, \dots, n\}$ and draw sample from the i-th kernel.

Scenario comparison - intro

- For generating the scenarios, we rely on the observed scenarios described using the parameters $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$.
- For testing the scenarios, let us assume that we have another set of observed scenarios described similarly using parameters $\mathcal{Z} = \{z_1, z_2, \dots, z_{N_z}\}$.
- Let the set of generated scenarios be denoted by $\mathcal{W} = \{w_1, \dots, w_{N_w}\}$.
- Ideally, we want the pdf of the generated scenarios ($\hat{f}(\cdot)$, based on $\hat{f}_H(\cdot)$) to be similar to the pdf of the real data ($f(\cdot)$).
- Our goal is to find a measure that quantifies this similarity.

11

11

Scenario comparison - Wasserstein

- The p-th Wasserstein distance is used to compare two probability distribution functions $\zeta(u)$ and $\eta(u)$ with $u \in \mathcal{U}$ and Γ denotes the joint distribution of (u, v) that has marginal ζ and η:

$$W_p(\zeta, \eta) = \left(\inf_{\gamma \in \Gamma(\zeta,\eta)} \left\{ \left(d(u,v)\right)^p \, \mathrm{d}\gamma(u,v) \right\} \right)^{1/p}$$

- In our case, we have do not have the actual distributions $f(\cdot)$ and $\hat{f}(\cdot)$, but only approximations:
$f(x) \approx \frac{1}{N_z} \sum_{i=1}^{N_z} \delta(x - z_i)$ and $\hat{f}(x) \approx \frac{1}{N_W} \sum_{i=1}^{N_w} \delta(x - w_i)$

12

12

Scenario comparison – Wasserstein (cont'd)

- This lead to the following Wasserstein distance (perhaps explain how to get from the general Wasserstein distance to this one in an appendix):

$$W(\mathcal{W}, \mathcal{Z}) = \inf_T \sum_{i=1}^{n_w} \sum_{j=1}^{n_z} d(w_i, z_i) T_{i,j}$$

with $\sum_{i=1}^{n_w} T_{i,j} = \frac{1}{n_z}, \forall j \in \{1, \dots, n_z\}, \sum_{j=1}^{n_z} T_{i,j} = \frac{1}{n_w}, \forall i \in \{1, \dots, n_w\},$

$$T_{i,j} \geq 0, \forall i \in \{1, \dots, n_w\}, j \in \{1, \dots, n_z\}$$

- Here, $d(w, z) = \|\alpha w - \alpha z\|$ is the distance between two scenarios (we will define it later)

13

13

Scenario comparison - Adaptation

- If $N_z \to \infty$ and $N_w \to \infty$, then $W(\mathcal{W}, \mathcal{Z})$ will approach $W_1(\hat{f}, f)$.
- For N_w, we can choose a fairly high number, as it is only limited by the available computing resources.
- However, to increase N_z, more data is needed and this is generally more expensive.
- Therefore, we propose an adaptation, based on the following intuition: Because $\mathcal{X}$ and $\mathcal{Z}$ are based on the same underlying pdf, $W(\mathcal{W}, \mathcal{X})$ should be approximately equal to $W(\mathcal{W}, \mathcal{Z})$. If this is not the case, it means that the generated scenarios are too much skewed towards the training data.

14

14

Scenario comparison – Adaptation (cont'd)

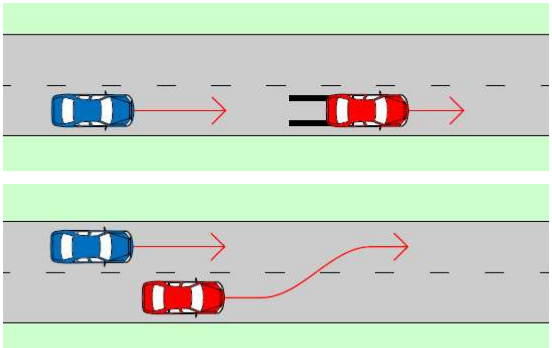
- We introduce a penalty in case the generated scenarios are skewed too much towards the training scenarios. The metric becomes:
$$M(\mathcal{W},\mathcal{Z},\mathcal{X}) = W(\mathcal{W},\mathcal{Z}) + \beta\big(W(\mathcal{W},\mathcal{Z}) - W(\mathcal{W},\mathcal{X})\big)$$
- Here, β is a parameter that controls the weight of the penalty.
- The goal is to minimize this metric.
- In the experiment, we will show what an appropriate value of β could be.

15

15

Results – intro

- Two examples:
  - Braking lead vehicle: $x(t)$ is the speed of the lead vehicle
  - Cut-in: $x(t)$ is two-dimensional as it contains both the speed and the lateral position w.r.t. ego lane.



16

16

Results – lead vehicle braking

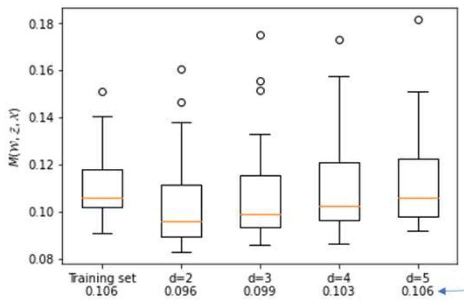
- y is the speed of the lead vehicle, $n_y = 1$
- $n_t = 50$
- No extra parameters: $n_\theta = 0$
- $n_t = 50, n_x = 51$
- $N = 983, N_z = 327, N_w = 10000$
- $\alpha_1 = \alpha_2 = \cdots = \alpha_{50} = 1, \alpha_{51} = 25$
- $\beta = 0.5$
- For the KDE, the data is first scaled, such that each parameter has standard deviation. Then, the bandwidth matrix $H = h^2 I$ is used, where h is determined using one-leave-out cross validation.

17

17

Results – lead vehicle braking (cont'd)

- We repeat the experiment 50 times. Each time, $N = 983$ datapoints are allocated for "training" and the remaining $N_z = 327$ datapoints are allocated for "testing".
- $d = 2$ gives the best median score.



median

18

18

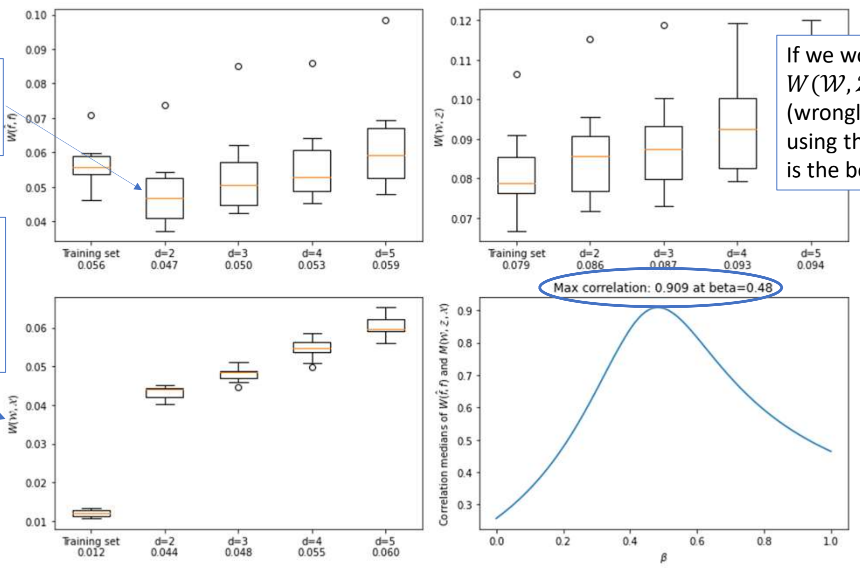
Results – lead vehicle braking (cont'd)

- To motivate our choice of β, we repeat the experiment, but we assume the approach for generating scenarios with $d = 2$ is the true pdf.
- Therefore, we can generate as many scenarios as we want as treat them as observations. As a result, N_z can be increased, such that the Wasserstein distance $W(\mathcal{W}, \mathcal{Z})$ is closer to $W_1(\hat{f}, f)$. We will use $N_z = 10000$ to approximate $W_1(\hat{f}, f)$.
- As shown in the next slide, the correlation between the medians of the "true Wasserstein" (approximated using $N_z = 10000$) and "our metric" ($M(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ with $N_z = 327$) is maximized at $\beta \approx 0.48 \approx 0.5$.

Results – lead vehicle braking (cont'd)

The "true Wasserstein" gives the best result for $d = 2$.

The higher this value, the more the generated scenarios are different from the training samples.



If we would use $W(\mathcal{W}, \mathcal{Z})$, we would (wrongly) think that using the training set is the best option.

Results – cut-in

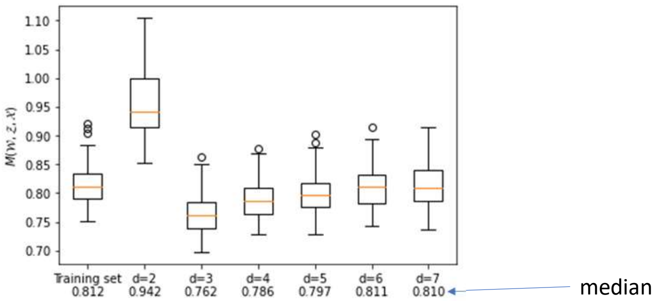
- y is the speed of the other vehicle and the lateral position of the other vehicle, $n_y = 2$
- $n_t = 50$
- Extra parameters: Initial speed ego vehicle, initial relative longitudinal position other vehicle, $n_\theta = 2$
- $n_t = 50, n_x = 103$
- $N = 224, N_z = 74, N_w = 50000$
- $\alpha_1 = \alpha_3 = \cdots = \alpha_{99} = 0.0033, \alpha_2 = \alpha_4 = \cdots = \alpha_{100} = 0.012, \alpha_{101} = 0.57, \alpha_{102} = 0.23, \alpha_{103} = 0.067$
- $\beta = 0.2$
- For the KDE, the data is first scaled, such that each parameter has standard deviation. Then, the bandwidth matrix $H = h^2 I$ is used, where h is determined using one-leave-out cross validation.

21

21

Results – cut-in (cont'd)

- We repeat the experiment 50 times. Each time, $N = 224$ datapoints are allocated for "training" and the remaining $N_z = 74$ datapoints are allocated for "testing".
- $d = 3$ gives the best median score.



22

22

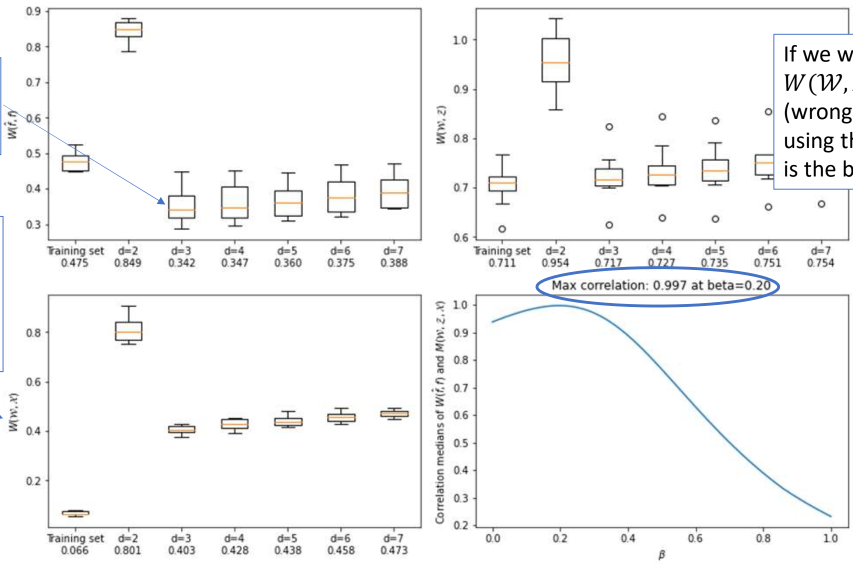
Results – cut-in (cont'd)

- To motivate our choice of β, we repeat the experiment, but we assume the approach for generating scenarios with $d = 3$ is the true pdf.
- Therefore, we can generate as many scenarios as we want as treat them as observations. As a result, N_z can be increased, such that the Wasserstein distance $W(\mathcal{W}, \mathcal{Z})$ is closer to $W_1(\hat{f}, f)$. We will use $N_z = 5000$ to approximate $W_1(\hat{f}, f)$.
- As shown in the next slide, the correlation between the medians of the "true Wasserstein" (approximated using $N_z = 5000$) and "our metric" ($M(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ with $N_z = 74$) is maximized at $\beta \approx 0.20$.

Results – cut-in (cont'd)

The "true Wasserstein" gives the best result for $d = 3$.

The higher this value, the more the generated scenarios are different from the training samples.



If we would use $W(\mathcal{W}, \mathcal{Z})$, we would (wrongly) think that using the training set is the best option.

Discussion

- This paper discusses a method to generate “realistic” scenarios. To do an effective assessment, the focus should be on scenarios that might lead to critical behaviour (mention “importance sampling”)
- Choosing β is a bit arbitrary. More research is needed.
- Conditional sampling is not as straightforward anymore, because the final parameters (resulting from the SVD) do not have a clear physical meaning (point out that there may be a solution for this).

25

25

Conclusions

- Summary of achievements:
  - New method for generating scenarios
  - New metric for evaluating the appropriateness of the generated scenarios
  - Two examples that illustrate the method
- Future outlook:
  - Apply method for other types of scenarios
  - Apply importance sampling

26

26