Time-to-Saccade metrics for real-world evaluation

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we explore multiple metrics for the evaluation of time-to-saccade problems. We define a new sampling strategy that takes the sequential nature of gaze data and time-to-saccade problems into account to avoid samples of the same event into different datasets. This also allows us to define novel error metrics, evaluating predicted durations utilizing classical eye-movement classifiers. Furthermore, we define metrics to evaluate the consistency of a predictor and the evaluation of the error over time. We evaluate our method using state-of-the-art methods along with an average baseline on three different datasets.

1 Introduction

When we as humans perceive a scene, our eyes constantly due to their relatively small area of sharp vision, the fovea ([10]). This restriction also applies when we perceive a virtual environment (*VE*) through a head-mounted display (*HMD*). In both cases, it is possible to determine the different eye-movements by taking their inherent properties, such as velocity and acceleration, into account and classify them into their respective class [20, 27, 1, 14, 2, 19, 5]. However, such a classification can only be performed after capturing the sample. This, makes a real-time utilization of gaze events challenging due to the low update rates and high latencies found in commercial head-mounted displays or wearable eye-trackers [21, 15]. This is especially true for saccades, as they are temporarily short, fast eye-movements in the range of 30-80 ms [10], where wearable eye-trackers often just have a few samples to classify them correctly.

Nonetheless, knowing when a saccade event occurs would benefit several virtual reality (*VR*) applications, such as gaze forecasting [11, 12], blink or saccade detection for redirected walking [15, 22], gaze contingent rendering [3] or intend based gaze interaction [6]. Furthermore, the prediction of fixation durations is also important in other areas outside *VR*, with one example being scanpath prediction which try to predict fixation durations along with the sequence of fixation points on a visual stimulus [26]. To use these gaze events, previous applications often mitigate the latency through unnatural actions, such as intentional blinking [15] or long saccade durations [22].

A proposal for a different approach was recently introduced by Rolff et al. [18]. They redefined the problem of gaze classification as a recurrent time-to-event prediction of saccade events, predicting the time it takes until a saccade occurs. However, this approach is fairly general and can also be applied to other gaze events, such as fixations or blinks. In contrast to these classical approaches, this redefinition of gaze event classification as a recurrent time-to-event problem allows estimating the remaining time-steps for each input sample of an eye-tracker. This provides information on how long it will take until the specified event will occur. This is desirable, as most of the time it is not essential if the class for each time-step is known, but rather when its class will change. In contrast to classical gaze classification methods, the redefinition also allows to account for the latency of eye-trackers found in commercial head-mounted displays or wearable eye-trackers. To evaluate their approach, Rolff et al. [18] utilize the mean absolute error (*mae*) on a set of randomly sampled time-to-event values to evaluate how well their method performs for time-to-saccade prediction.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this paper, we define a more fitting sampling strategy than random sampling. This allows to adapt the previously used error metric to be more suited to the actual problem of time-to-saccade prediction. For this, we will take the physical limitations of an eye-tracker and the possibility to utilize past information into account. We will explore how well these metrics can be utilized to understand the prediction and provide a different evaluation method.

To summarize, our work proposes the following contributions:

- A different sampling strategy for time-to-saccade data that takes the sequential information of the time-to-event of a gaze event into account.
- Define novel error metrics using the previously defined sampling strategy, enabling a more interpretable result to infer the predictive performance of a time-to-saccade predictor.

2 Related Work

A commonly utilized metric for time-to-event problems is Harrell's concordance index (*c-index*) [9]. The *c-index* measures the correlation between the predicted risk-score and the observed time-to-event. Hence, a higher risk value should correlate with a shorter time-to-event. However, it has also been shown that the *c-index* is biased if the test set contains a high number of censored samples [23], leading to an alternative definition by Uno et al. [23]. Another metrics commonly used is the brier-score [4]. Its definition is equivalent to the mean square error (*mse*) for probabilistic predictions of binary events, hence, requiring a probabilistic prediction from the employed model. It has also been re-defined to allow censored data [8].

Besides survival-analysis related metrics, there are multiple metrics for time-series forecasting using the real values of the prediction. Commonly used metrics are the mean-absolute, mean-square, or (normalized) root-mean-square error [7, 13]. Variations of those metrics have been proposed, such as the (symmetric) mean absolute percentage error. Most of the listed error metrics, assume that an overestimation should be penalized equally. This, however, might not always be the case, thus requiring an asymmetric error function such as the asymmetric mean [7] or the linex error [7, 24].

3 Methodology & Metrics

For our experimental setup, we follow, if not noted otherwise, Rolff et al. [18], allowing for comparability between both approaches. First, we would like to highlight a disadvantage of their evaluation, the random sampling. This does not take the temporal property of the gaze data into account, as the samples of the same time-to-saccade sequence might have been selected for different datasets. As a result, it is impossible to evaluate properties, like if the predictor was consistent with its prediction or how the error of the overall sequence of a single sequence have been, without the predictor already seen part of the data. This makes it challenging to interpret the reported error metrics, as it is not clear how the predictor behaves over time.

Here we change the methodology, by introducing a new sampling strategy. This sampling strategy utilizes the existing time-to-event annotations. As we split the gaze signal exactly if an event happens instead of having randomly chosen individual samples. It results in several sequences containing individual samples as data points. This also allows us to derive some properties. One being that the first time-to-event value in the sequence is always equal to the length of the whole sequence and the last time-to-event value is always one step before the desired event. Another advantage is that the time-to-event of such a sequence is always strictly monotonically decreasing, with the rate being depended on the update rate of the eye-tracker. With the measured update rate $f_i > 0$ of the eye-tracker at step i each time-to-saccade value (tts) can then be calculated through: $\text{tts}_{i+1} = \text{tts}_i - f_i^{-1}$.

Using these observations, it allows us to explore additional error metrics that account for the temporal properties of a time-to-saccade sequence and which take and information of eye-movement classifiers into account. Same as Rolff et al. [18], we would also like to advise against the usage of earlier listed time-to-event metrics for the evaluation of time-to-saccade predictions, even under the new sampling strategy. First, time-to-saccade predictions are rarely right censored. This is due to the capturing setup and the observation that gaze events are repeatable events that happen every 300 to 2500 ms. As a result, the only right censored sequences are at the end of an eye-tracking session. Thus, depending on the length of data capture, often only corresponding to a small proportion of

the captured dataset. This would only require the non-censored time-to-event metrics. Moreover, as the task of time-to-saccade prediction requires predicting the time-to-event as accurate as possible, other metrics such as the *c-index* do not provide helpful information on their accuracy. Here, it is better to utilize metrics for time-series forecasting, as they are concerned with the difference between the actual time-to-event and the predicted time. However, these are fairly general and do not allow insight into a time-to-saccade predictor model. Hence, we propose the additional metrics:

Consistency: To measure how consistent the model is in its prediction, we define consistency of a sequence j with length l as the relative difference [7] between the current and the next prediction. Ideally, this change should be equal to the update rate of the eye-tracker, due to the formulation of time-to-saccade. Hence, we can define consistency as:

$$c_j = \sum_{i=0}^{l-1} \frac{||p_{i+1} - p_i| - f_{i+1}|}{|f_{i+1}|}$$

As this gets evaluated over each sequence, we can derive the mean consistency of a dataset through the arithmetic mean.

Average overshot and undershot rate: The overshot and undershot values measure if the model generally tends to predict too short or too long durations for its prediction. In the non-recurrent definition of a time-to-saccade problem, this would be equal to the actual duration of the event with the predicted one. This, however, is not possible with a recurrent definition, as the predictor outputs an estimate for each step. Therefore, we estimate the mean time-to-saccade p to estimate the over- and undershot rate. While this is not optimal as the undershot might be at the start of a time-to-saccade prediction and therefore not be at the final prediction shortly before the event, we assume this to be a reasonable approximation for a general overview. Further, we assume the real time-to-event to be t. With those, we define the overshot and undershoot offset o of the sequence as: $o = t - p$. Using this offset, we can define a prediction to be an overshot if $o > 0$ and an undershot if $o < 0$. Using those, we can define the arithmetic mean over all sequences as the average overshot and undershot rate. We assume undershots to be more of a problem than overshots due to their ability to trigger downstream methods with the user being aware of them. In contrast, an overshoot can be mitigated through the utilization of data samples from the eye-tracker.

Average sequence and undershot error: Assuming that we would not perform an action if the predictor performed an overshoot, we can define the undershot error as the part of the prediction where it undershoots the signal. In this case, we would not be able to notice when employing the predictor in an application due to the abundance of additional information that we can utilize. Hence, the ***undershot error*** measures if a model undershoots the actual target time-to-event and assumes a perfect prediction otherwise. In case of an average time-to-saccade error (avg. tts.), we omit this assumption and estimate the error based on the average time-to-saccade of the sequence. To stay consistent with previous literature, we utilize the mean absolute and mean square error for undershot error estimation.

Sectioning: The prediction of a model may change with time. Depending on the utilized method, it might not have enough information to predict an accurate time-to-saccade. This could be for example be at the beginning during the execution of the last saccade. As a result, the prediction may improve over time without being inherently evident from the evaluation. Hence, we split each time-to-saccade sequence into n parts of equal length and evaluate them separately to each other.

4 Evaluation & Discussion

To evaluate our approach defined in Sec. 3 we utilize a linear regressor with a Nyström approximation [25] trained through stochastic gradient descent [17]. This has been chosen, as it was identified as the best performing regressor among four other classical methods [18]. The models were trained as specified in [18]. One notable exception while training is the used sampling strategy. Here, we made sure that samples leading to the same gaze event are placed inside the same train, test, or validation dataset. To train the models, we utilize the DGaze [11], FixationNet [12] and EGTEA Gaze+ [16] datasets. In addition, we use some artificial prediction strategies to evaluate the proposed metrics on synthetically generated predictions. For those, we employ: mean time-to-saccade (mean), zero prediction (zero), maximum time-to-saccade (max), and random time-to-saccade prediction (rand). A more extensive evaluation of those can be found in the appendix.

Table 1: Results of the SGD regressor and average time-to-event using the metrics described in Sec. 3 and the mean square error (mse) and mean absolute error (mae). A lower error is preferred.

Metric	SGD			avg. time-to-event		
	DGaze	FixationNet	EGTEA	DGaze	FixationNet	EGTEA
mse↓	0.1285 s^2	0.2390 s^2	0.1672 s^2	0.2314 s^2	0.3647 s^2	0.3043 s^2
mae↓	0.2556 s	0.3567 s	0.2668 s	0.3387 s	0.4261 s	0.3677 s
avg. tts mse↓	0.0494 s^2	0.0887 s^2	0.0420 s^2	0.0672 s^2	0.1035 s^2	0.0745 s^2
avg. tts mae↓	0.1747 s	0.2422 s	0.1484 s	0.1792 s	0.2260 s	0.1765 s
undershot mse↓	0.0319 s^2	0.0537 s^2	0.0311 s^2	0.0664 s^2	0.1003 s^2	0.0718 s^2
undershot mae↓	0.0857 s	0.1105 s	0.0799 s	0.1666 s	0.1968 s	0.1470 s
o. / u. shot rate↓	0.61/0.39	0.63/0.37	0.60/0.40	0.27/0.73	0.36/0.64	0.43/0.57
consistency↓	1.64	1.39	1.12	1.0	1.0	1.0

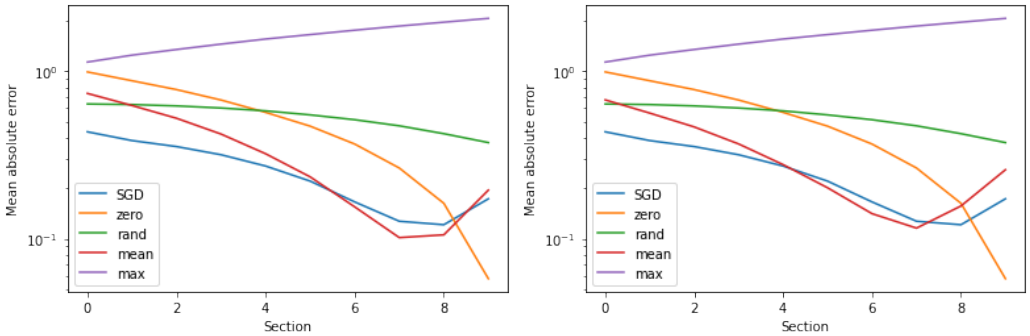

Figure 1: Error of different predictors on different sections as explained in Sec. 3. We divide all time-to-saccade sequences into 10 different sections of equal length to estimate the mean square error on the DGaze [11] (left) and FixationNet [12] (right) datasets.

Table 1 shows the measured results of the predictions on the DGaze [11], FixationNet [12] and EGTEA Gaze+ [16] datasets. While close to previous literature [18], the results are still slightly different due to the different sampling method. However, it is also evident, that this results in a higher overshot rate, as the predictor is not able to estimate the correct time-to-saccade for most data samples. Moreover, the consistency of the SGD predictor is not as optimal as the average prediction. This is expected, as the average predictor reports very consistent results by predicting the mean value for every sample. It can also be seen that the undershot error reports much lower results for the SGD predictor when compared to the average time-to-event. This is consistent with the undershot rate, as the predictor undershoots less than the average predictor, making it more useful for real-world applications. Fig. 1 shows the evaluation of the 5 different baseline predictor models along with the SGD predictor over multiple sections of all sequences. Here, it can be seen that the SGD predictor outperforms all baselines most of the time, except for a brief range 20-30% of the length before the actual event, where it is outperformed by the mean absolute error. This indicates that the SGD tends to do better than the other predictor, but eventually fails shortly before the actual event. We also performed additional evaluations, which can be found in the appendix due to space restrictions.

5 Conclusion

In this paper, we proposed a new sampling strategy that lets us take the sequential information of gaze data for time-to-saccade prediction into account. This enabled us to define multiple new metrics capturing the consistency and duration of time-to-saccade predictors, as well as capturing the overall behavior of them over different parts of time-to-saccade sequences. To evaluate these, we use the state-of-the-art time-to-saccade predictor and compared it against a simple average baseline. However, we also expect future work on this topic, especially overshot and undershot evaluation, as they currently just evaluate the average over- and undershot over the whole sequence but do not take the prediction strategy of a proposed model into account.

References

[1] Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. Smooth pursuit detection based on multiple observers. In *Proceedings of the 2016 Symposium on Eye Tracking Research & Applications (ETRA'16)*.

[2] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior research methods*, 49(2):616–637, 2017.

[3] Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[4] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78 (1):1–3, 1950.

[5] Asim H Dar, Adina S Wagner, and Michael Hanke. Remodnav: robust eye-movement classification for dynamic stimulation. *Behavior research methods*, 53(1):399–414, 2021.

[6] Brendan David-John, Candace Peacock, Ting Zhang, T. Scott Murdison, Hrvoje Benko, and Tanya R. Jonker. Towards gaze-based prediction of the intent to interact in virtual reality. In *Proceedings of the 2021 Symposium on Eye Tracking Research & Applications (ETRA'21)*, 2021.

[7] Francis X Diebold. *Elements of forecasting*. Citeseer, 1998.

[8] Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

[9] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *JAMA*, 247(18):2543–2546, 1982.

[10] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.

[11] Zhiming Hu, Sheng Li, Congyi Zhang, Kangrui Yi, Guoping Wang, and Dinesh Manocha. Dgaze: Cnn-based gaze prediction in dynamic scenes. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 26(5):1902–1911, 2020.

[12] Zhiming Hu, Andreas Bulling, Sheng Li, and Guoping Wang. Fixationnet: Forecasting eye fixations in task-oriented virtual environments. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 27(5):2681–2690, 2021.

[13] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[14] Oleg V Komogortsev and Alex Karpov. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior research methods*, 45(1):203–215, 2013.

[15] Eike Langbehn, Frank Steinicke, Markus Lappe, Gregory F. Welch, and Gerd Bruder. In the blink of an eye: leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

[16] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635, 2018.

[17] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22(3):400–407, 1951.

[18] Tim Rolff, Frank Steinicke, and Simone Frintrop. When do saccades begin? Prediction of saccades as a time-to-event problem. In *Proceedings of the 2022 Symposium on Eye Tracking Research & Applications (ETRA'22)*, 2022.

[19] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA'00)*, 2000.

[20] Mikhail Startsev, Ioannis Agtzidis, and Michael Dorr. 1d cnn with blstm for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, 51(2):556–572, 2019.

[21] Niklas Stein, Diederick C Niehorster, Tamara Watson, Frank Steinicke, Katharina Rifai, Siegfried Wahl, and Markus Lappe. A comparison of eye tracking latencies among several commercial head-mounted displays. *i-Perception*, 12(1), 2021.

[22] Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie Kaufman. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[23] Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

[24] Hal R Varian. A bayesian approach to real estate assessment. *Studies in Bayesian econometric and statistics in Honor of Leonard J. Savage*, pages 195–208, 1975.

[25] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.

[26] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. Predicting goal-directed human attention using inverse reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[27] Raimondas Zemblys, Diederick C Niehorster, and Kenneth Holmqvist. gazenet: End-to-end eye-movement event detection with deep neural networks. *Behavior research methods*, 51(2):840–864, 2019.