#3208

Learning to Estimate Driver Drowsiness from Car Acceleration Sensors using Weakly Labeled Data

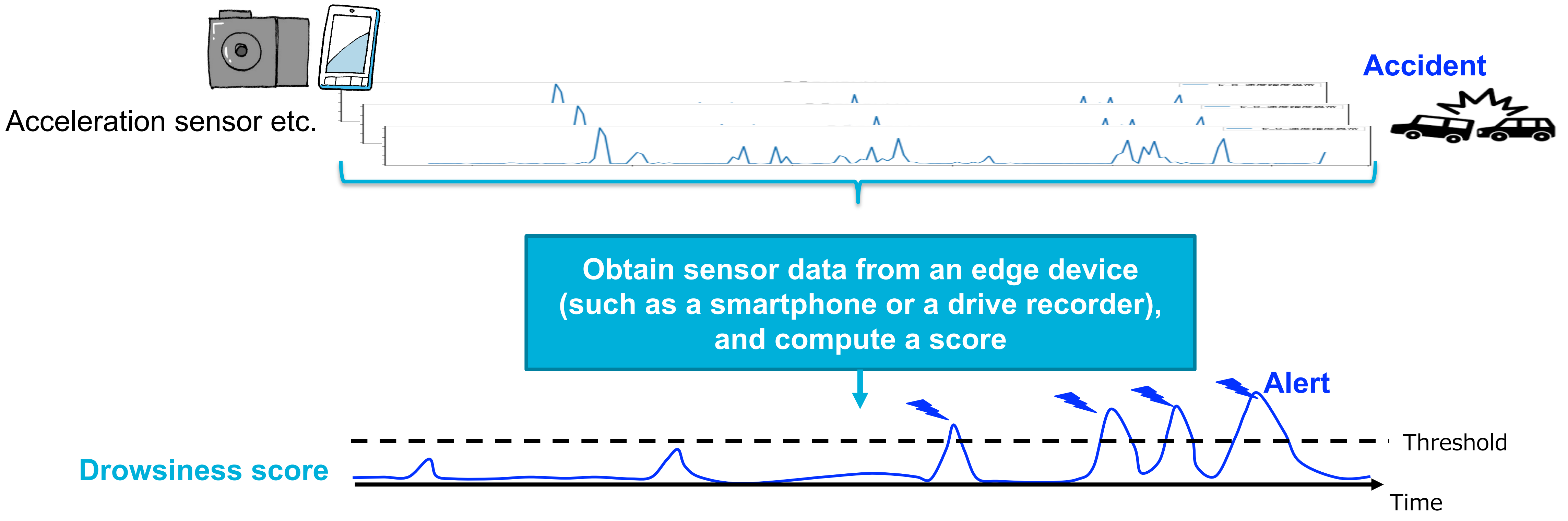
Takayuki Katsuki, Kun Zhao, and Takayuki Yoshizumi

IBM Research – Tokyo

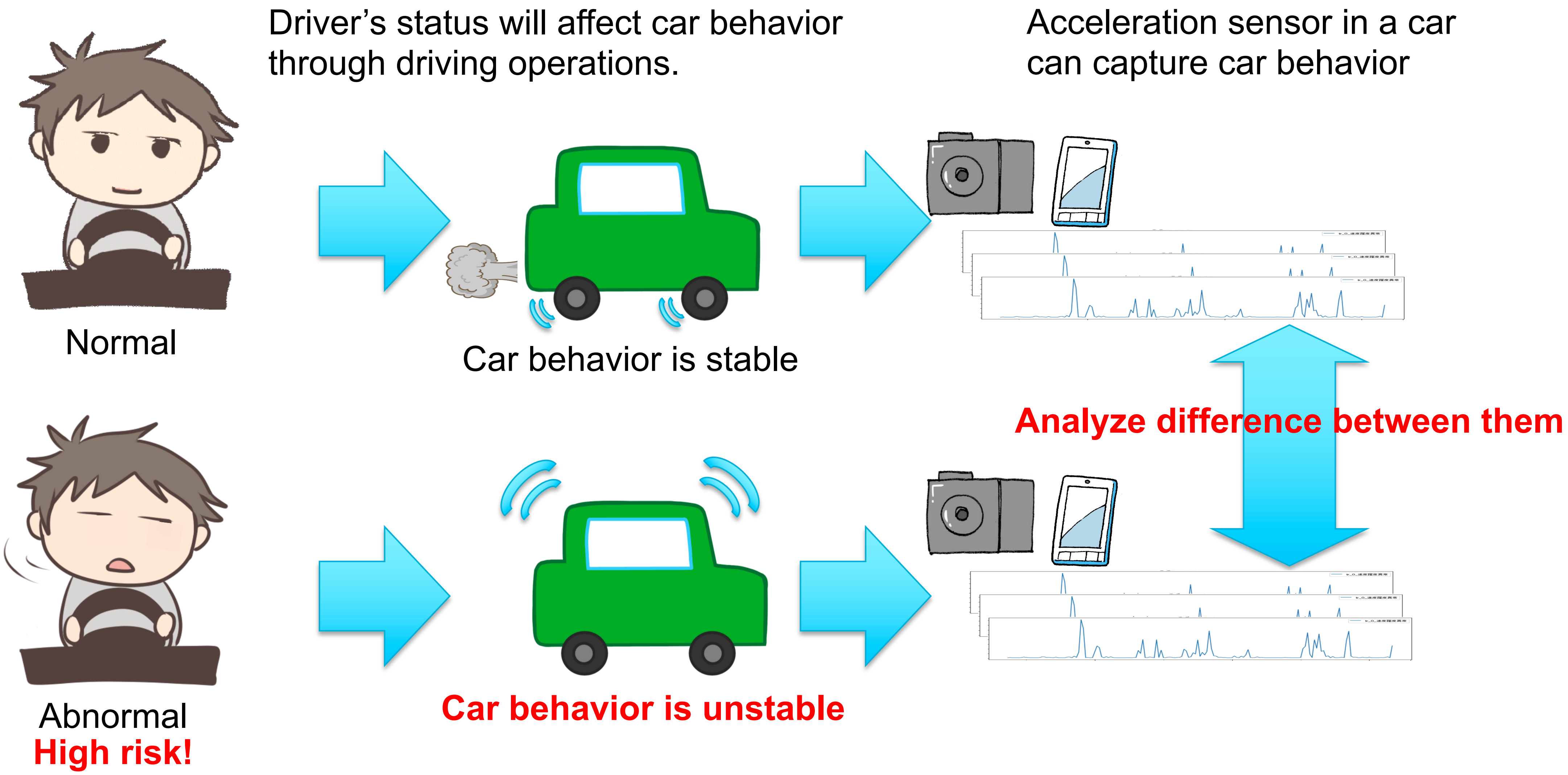
ICASSP 2020

Compute drowsiness score from car accelerations sensor data for each time

- Our method detect driver’s drowsiness that may cause an accident.
- Input: three-axis acceleration (from acceleration sensor), speed, and direction (from GPS)
- Output: **drowsiness score**



Driver's status can be estimated by sensors mounted on a car

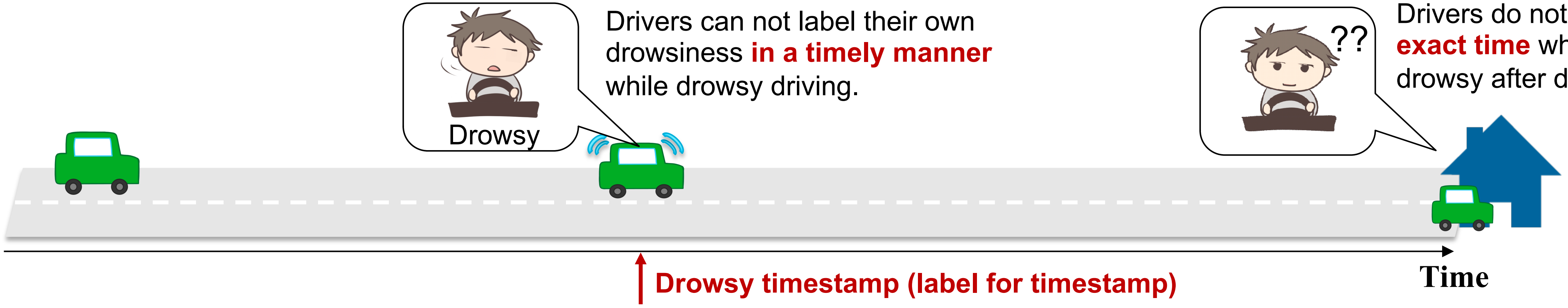


Since obtaining label for each timestamp is not a realistic goal, we use *weak* labels for whole trip not for timestamp

- **Too difficult (Ordinal setting)**

Drivers can not label their own drowsiness **in a timely manner** while drowsy driving.

Drivers do not remember **exact time** when they felt drowsy after driving.



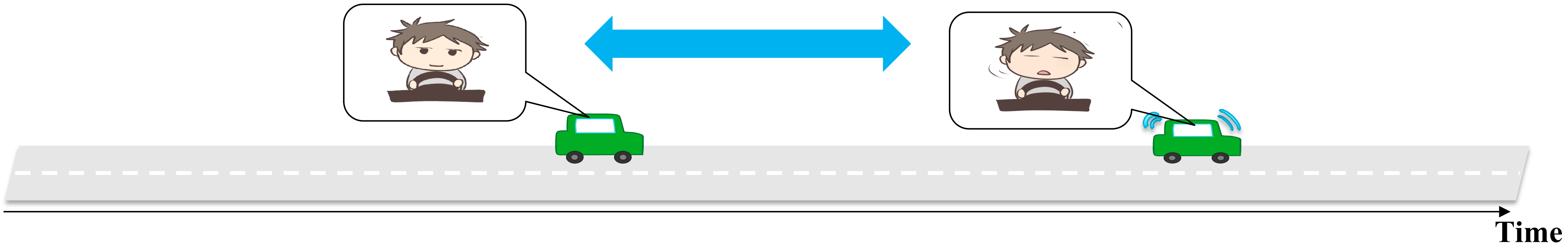
- **Easy (Our approach assigning label to whole trip)**

Drivers can answer **whether they felt drowsy while they were driving** after the trip!!

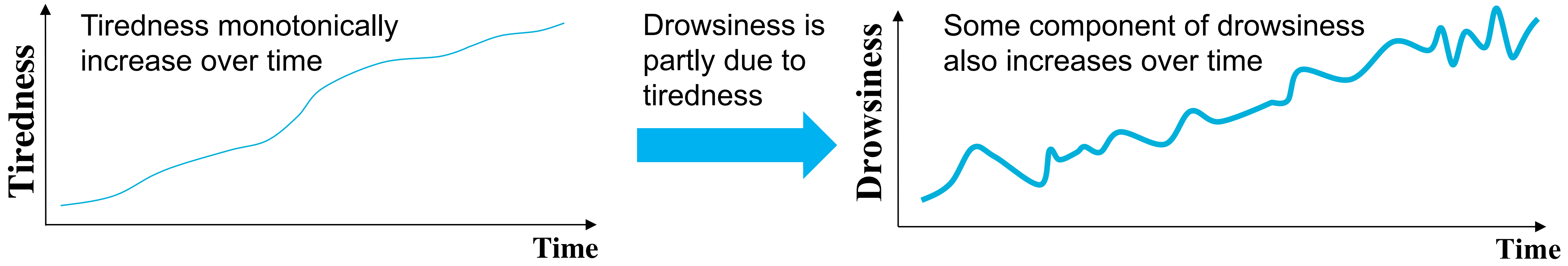
To learn the estimation function from weak labels, we impose two assumptions

1) Drowsiness is measured on an ordinal scale

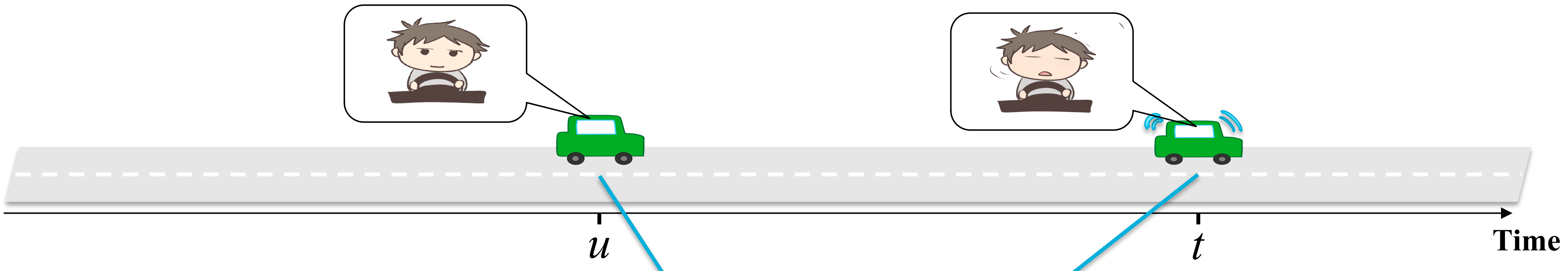
We can consider which of two different data points is drowsier



2) Drowsiness increases as the trip gets longer due to tiredness



We learn the estimation function for driver drowsiness such that estimated drowsiness score becomes large as driving time increases

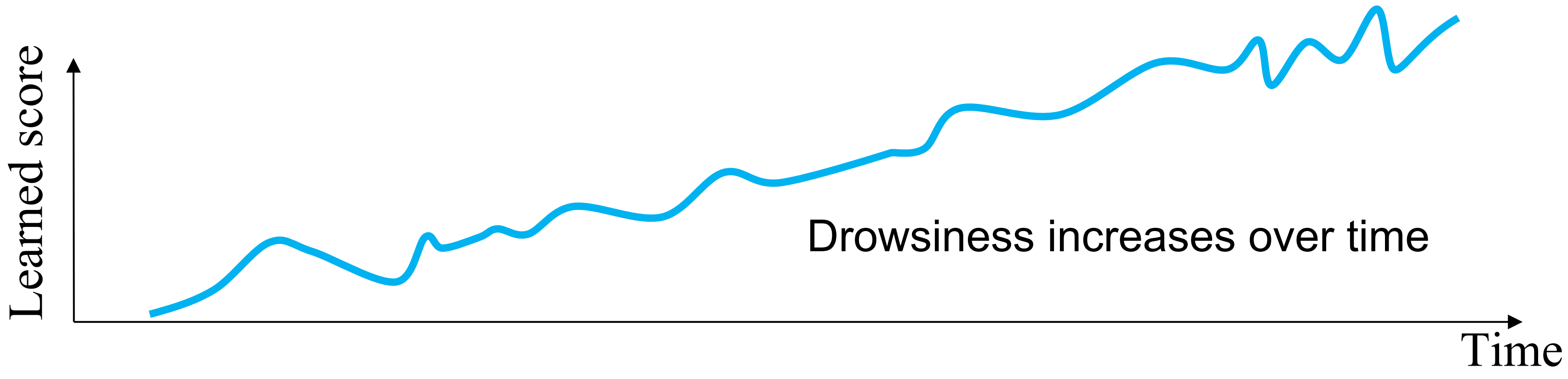


If $u < t$, our learning algorithm adjusts f as

Compare

$$f(\boldsymbol{x}_u) < f(\boldsymbol{x}_t)$$

$\boldsymbol{x}$ is input features from acceleration sensors at each time



Learning ordinal scale of drowsiness from weakly labeled data

- Optimize the following expected loss (objective function) in data labeled for the whole trip as drowsy

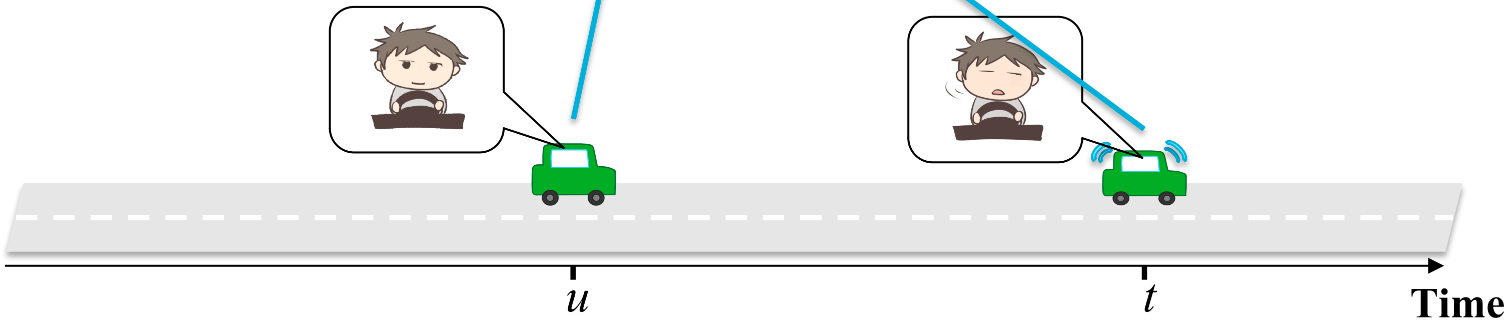
$$f^* \equiv \operatorname*{argmin}_f E[L(f)]$$

$$L(f; \boldsymbol{x}_t, \boldsymbol{x}_u) \equiv \max\big(0, 1 - \operatorname{sgn}\big(t - u\big)\big(f(\boldsymbol{x}_t) - f(\boldsymbol{x}_u)\big)\big)$$

It does not require to know absolute label's value and we just compare which data is drowsier

It returns larger loss when score with shorter driving time is greater than that with longer driving time

- $\boldsymbol{x} \in \mathbb{R}^D (D \in \mathbb{N})$: D-dimensional input features
- f : Estimation function for drowsiness
- $L(f)$: Loss function
- t , u : Timestamps



Stochastic Optimization for Scalable Learning

- Let $\boldsymbol{P}^{(i)}$ be all candidate pairs of samples in i-th trip $\boldsymbol{X}^{(i)}$; we empirically approximate the expectation in the objective function, as

$$\mathcal{L}(f) \simeq \hat{\mathcal{L}}(f)$$

$$\equiv \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\boldsymbol{P}^{\{i\}}|}\sum_{\boldsymbol{x}_t^{(i)},\boldsymbol{x}_u^{(i)}\in\boldsymbol{P}^{\{i\}}} L\left(f;\boldsymbol{x}_t^{(i)},\boldsymbol{x}_u^{(i)}\right)$$

- For stable learning, we add a regularization term, $R(f)$, and then derive the gradients as

$$\frac{\partial\hat{\mathcal{L}}(f)}{\partial\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\boldsymbol{P}^{\{i\}}|}\sum_{\boldsymbol{x}_t^{(i)},\boldsymbol{x}_u^{(i)}\in\boldsymbol{P}^{\{i\}}}\frac{\partial L\left(f;\boldsymbol{x}_t^{(i)},\boldsymbol{x}_u^{(i)}\right)}{\partial\boldsymbol{\theta}} + \lambda\frac{\partial R(f)}{\partial\boldsymbol{\theta}}$$

- $\boldsymbol{\theta}$: Model parameters

Algorithm 1 Stochastic optimization for learning to estimate driver drowsiness.

Require: Training data $\{\boldsymbol{X}^{(i)}\}_{i=1}^{N}$ and hyperparameter λ

Ensure: Model parameter $\boldsymbol{\theta}$ for f

1: Let $\mathcal{A}$ be an external stochastic optimization method
2: **while** No stopping criterion has been met **do**
3: Randomly select i from 1 to N
4: Randomly select t and u from 1 to T_i $(t \neq u)$
5: $G \leftarrow \frac{\partial L\left(f;\boldsymbol{x}_t^{(i)},\boldsymbol{x}_u^{(i)}\right)}{\partial\boldsymbol{\theta}} + \lambda\frac{\partial R(f)}{\partial\boldsymbol{\theta}}$
6: Update $\boldsymbol{\theta}$ by $\mathcal{A}$ with the gradient G

- We can estimate the drowsiness score $\hat{y}$ for the new data $\boldsymbol{x}$ as

$$\hat{y} = \hat{f}(\boldsymbol{x})$$

$$= \hat{\boldsymbol{\theta}}^{\top}\boldsymbol{x}$$

The proposed method achieves better estimation performance for drowsiness than anomaly detection and classification approaches.

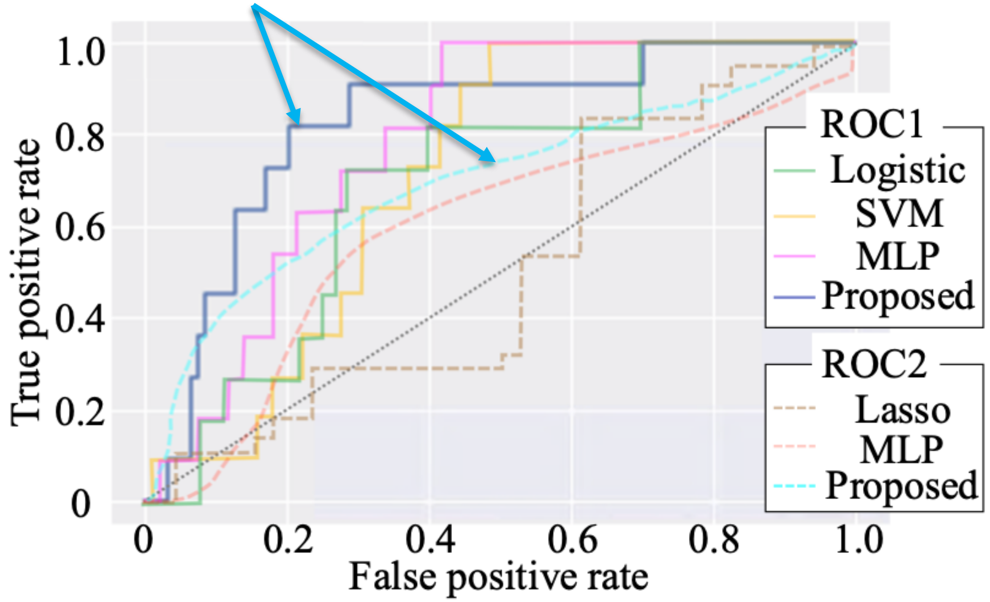
- The driving data consisted of 11 drowsy trips and 94 normal trips, including highway and ordinary road trips. There were about 40 driving hours (over 100,000 samples)

- The performance of our drowsiness score was evaluated regarding two types of the *area under the curve* (AUC) in cross-validation :
  - AUC1) classifying each trip as either a drowsy trip or normal trip based on the maximum score during the trip
  - AUC2) classifying each sample with a drowsy timestamp or normal timestamp based on the score.

- We compared the results of the proposed method with anomaly detection and classification approaches.

Method		AUC1	AUC2
Anomaly detection	Lasso	0.41	0.51
	Glasso	0.36	0.44
Classification	Logistic	0.69	0.34
	SVM	0.71	0.47
	MLP	0.79	0.59
Proposed		**0.82**	**0.69**

Proposed method were closest to the upper left corner



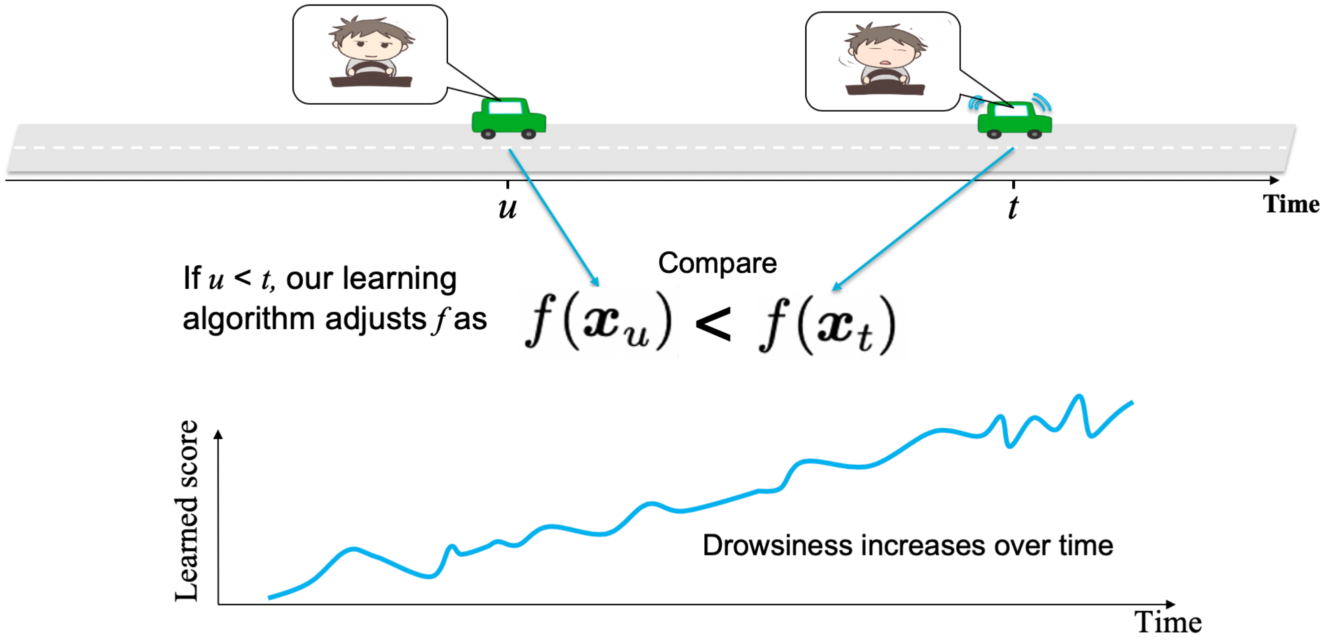
Important features for estimating drowsiness.

- Positive effect on drowsiness
  - Forward-backward jerk
    - Unstable **driver's** acceleration and braking
  - Forward-backward acceleration
  - Vertical jerk

- Negative effect on drowsiness
  - Lateral acceleration
    - **Route or vehicle's** steering characteristics which are not directly related to driver operation
  - Magnitude of acceleration
  - Angle-acceleration

Conclusion

- We formulated a learning problem for estimating driver drowsiness from car-acceleration sensor data using a weakly labeled data. The data for the whole trip rather than at each timestamp were labeled as drowsy or normal.
- We proposed a learning algorithm based on the assumption that some aspects of driver drowsiness increase over time due to tiredness.
- An experimental evaluation on real driving datasets demonstrated that the proposed method outperformed the baseline methods.



Method		AUC1	AUC2
Anomaly detection	Lasso	0.41	0.51
	Glasso	0.36	0.44
Classification	Logistic	0.69	0.34
	SVM	0.71	0.47
	MLP	0.79	0.59
Proposed		**0.82**	**0.69**

Thank you

Takayuki Katsuki, Kun Zhao, and Takayuki Yoshizumi
IBM Research – Tokyo