A hybrid Bayesian Network approach to detect driver cognitive distraction

Yulan Liang *, John D. Lee 1

Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA 52242, USA

ABSTRACT

Driver cognitive distraction (e.g., hand-free cell phone conversation) can lead to unapparent, but detrimental, impairment to driving safety. Detecting cognitive distraction represents an important function for driver distraction mitigation systems. We developed a layered algorithm that integrated two data mining methods—Dynamic Bayesian Network (DBN) and supervised clustering—to detect cognitive distraction using eye movement and driving performance measures. In this study, the algorithm was trained and tested with the data collected in a simulator-based study, where drivers drove either with or without an auditory secondary task. We calculated 19 distraction indicators and defined cognitive distraction using the experimental condition (i.e., “distraction” as in the drives with the secondary task, and “no distraction” as in the drives without the secondary task). We compared the layered algorithm with previously developed DBN and Support Vector Machine (SVM) algorithms. The results showed that the layered algorithm achieved comparable prediction performance as the two alternatives. Nonetheless, the layered algorithm shortened training and prediction time compared to the original DBN because supervised clustering improved computational efficiency by reducing the number of inputs for DBNs. Moreover, the supervised clustering of the layered algorithm revealed rich information on the relationship between driver cognitive state and performance. This study demonstrates that the layered algorithm can capitalize on the best attributes of component data mining methods and can identify human cognitive state efficiently. The study also shows the value in considering the supervised clustering method as an approach to feature reduction in data mining applications.

1. Introduction

Driver distraction has emerged as a critical risk factor of motor vehicle crashes. Recent data show that 16% of fatal crashes and 21% of injury crashes were attributed to driver distraction in 2008 (Ascone et al., 2009). The increasing use of information technologies in vehicles (e.g., navigation systems, smart phones, and other internet-based devices) will likely exacerbate the problem of distraction. From 2009 to 2010, visible headset cell phone use and visible manipulation of handheld devices while driving increased 50% – from 0.6% to 0.9%. These absolute values may underrepresent the usage of information technologies on road because drivers were observed for only approximately 10 s at sampled roadway sites and might have used technologies that were undetectable outside the vehicle, such as a blue-tooth headset) (NHTSA, 2011). An estimated
nine percent of drivers used either hand-free or hand-held phones while driving at a typical daylight moment in 2010 (NHTSA, 2011). Therefore, although drivers benefit from these devices, it is also critical for drivers to avoid distraction and direct an acceptable level of attention to the road.

A promising strategy to minimize the effect of distraction is to develop intelligent in-vehicle systems, namely adaptive distraction mitigation systems, which can provide real-time assistance or retrospective feedback to reduce distraction based on driver state/behavior, as well as the traffic context (Lee, 2009; Toledo et al., 2008). For example, when a driver is faced with an intense negotiation via cell phone in heavy traffic, the adaptive distraction mitigation system can warn the driver and encourage the driver to attend to the road, or in an extreme case, the system can automatically hold the call until the driver can get off the road. Such systems must accurately and non-intrusively detect whether drivers are distracted or not. In this context, distraction can be defined as a diversion of a driver's attention away from the activities critical for safe driving toward a competing activity (Lee et al., 2008).

Detecting driver distraction depends on how distraction changes driver behavior compared to the normal driving without distraction, which can depend on the type of distraction. Considering the nature of attentional resources that distraction competes with driving, visual distraction and cognitive distraction represent two critical types – “eye-off-road” and “mind-off-road” – although they are not mutually exclusive in real driving (Lee and Lee, 2010; Victor, 2005). Visual distraction relates to whether drivers look away from the road (i.e., on-road or off-road glances) and can be determined by momentary changes of drivers’ eye glances. A general algorithm that considers driver glance behavior across a relatively short period could detect visual distraction consistently across drivers (Liang et al., 2012).

However, detecting cognitive distraction is much more complex than visual distraction because the signs of cognitive distraction are usually not readily apparent, are unlikely to be described by a simple linear relationship, and can vary across drivers. Detecting cognitive distraction likely requires an integration of a large number of indicators (e.g., eye gaze measures) over a relatively long time and may need to be personalized for different drivers (Liang et al., 2007b). The challenge is how to integrate performance measures in a logical manner to quantify complex, even unknown, relationship between drivers’ cognitive state and distraction indicators. Data mining methods that can extract unknown patterns from a large volume of data present an innovative and promising approach to this end.

In previous studies, two data mining methods—Support Vector Machines (SVMs) and Dynamic Bayesian Networks (DBNs)—successfully detected cognitive distraction from driver visual behavior and driving performance (Liang et al., 2007a,b). SVMs, proposed by Vapnik (1995), are based on statistical learning theory and can be used for non-linear classification. To train binary-classification models, SVMs use a kernel function, \( K(x_i, x_j) = \Phi(x_i)\Phi(x_j) \), to map training data from the original input space to a high-dimensional feature space. When the mapped data are linearly separable in the feature space, the hyperplane that maximizes the margin from it to the closest data points of each class produces the minimized upper bound of generalization error and yields a nonlinear boundary in the input space. When the data are not linearly separable in the feature space, the positive penalty parameter, \( C \), allows for training error \( e \) by specifying the cost of misclassifying training instances (Hsu et al., 2008). The training process of SVMs is to minimize both training error and the upper bounds of generalization error. This method is computationally efficient and minimizes generalization error to avoid over-fitting. SVMs produce more robust models compared to the linear-regression algorithms that minimize the mean square error, which can be seriously affected by outliers in training data. Tested with the data collected in a simulator study, SVMs detected cognitive distraction with an average accuracy of 81%, outperforming traditional logistic regression method. The cognitive distraction was defined by the experimental conditions: either the drive when drivers drove under cognitive distraction or the drive without distraction. Nonetheless, SVMs do not consider time-dependent relationship between variables, and the resultant models do not present the relationships learned from data in an interpretable way.

Bayesian Networks (BNs) represent a probability-based approach and can be presented graphically (depicted in Fig. 1): nodes depicting random variables and arrows depicting conditional dependencies between variables. For example, the arrow

![Fig. 1. Two examples of Bayesian Networks.](image-url)
between variable nodes $H$ and $S$ indicates that $S$ is independent of all variables other than $H$. Dynamic BNs, one type of BNs, can model a time-series of events according to a Markov process (Fig. 1b). The training process of BN models included structure learning and parameter estimation. Structure learning identifies the possible connections between nodes in a BN, whereas parameter estimation identifies the conditional probabilities for those connections (Ben-Gal, 2007). Compared with SVMs, DBNs are easy to interpret, can consider time-dependent relationship between cognitive state and distraction indicators, and obtain more accurate and sensitive models (Liang and Lee, 2008). However, DBNs are not computationally efficient, needing an average 20 min of processing time to train a model, compared to 15 s to train a SVM model with the same training data.

To obtain accurate, efficient, and interpretable distraction detection algorithms, we combined DBNs and a feature reduction method (e.g., clustering) in a hierarchical manner (Fig. 2). The hierarchical structure has been demonstrated to be effective in some other detection systems that need to integrate a number of variables, similar to the detection of cognitive distraction. Veeraraghavan et al. (2007) combined an unsupervised clustering method and a binary Bayesian eigenimage classifier in a cascade fashion to identify driver activities in vehicles from computer vision data. Another study combined a Dynamic Bayesian Clustering and a SVM model in sequence to forecast electricity demand (Fan et al., 2006). These models have two layers. The lower-layer model summarizes basic measures into more abstract characteristics of the target so that the higher-layer model classifies example with fewer indicators. This approach can reduce the computational load and make contributions of model inputs interpretable relative to the ultimate classification.

Our approach uses supervised clustering models at the lower layer to identify feature behaviors associated with cognitive distraction (i.e., clusters) based on a number of performance measures. Supervised clustering methods are built upon the concept of traditional unsupervised clustering methods, but extend the concept by giving some directions (i.e., supervised) in the blind search for the structure among instances, in a manner analogous to Partial Least Squares as a supervised version of Principal Component Analysis.

At the higher-layer, a DBN model uses the labels of these feature behaviors as input values to recognize driver cognitive state. This algorithm reduces the number of input variables to the DBNs and is expected to improve computational efficiency relative to the original DBN algorithm. At the same time, the layered algorithm preserves time dependency and ease of interpretation. The objective of this study is to demonstrate that the layered algorithm is an accurate, efficient, and interpretable approach to detect driver cognitive distraction, compared with the interpretable, but inefficient, DBNs and the uninterpretable, but efficient, SVMs.

**Fig. 2.** The structure of the layered algorithm. The curving, solid arrows indicate data flow. The straight, lined arrows in the DBN algorithm indicate associations between variables.
2. Method

We used the data collected in a simulator-based experiment to train three types of detection algorithms: the layered algorithm, previous developed DBN algorithm (original DBN algorithm), and SVM algorithm. These algorithms were compared in terms of prediction performance, both accuracy and efficiency, and interpretability.

2.1. Experimental data

The data were collected on nine participants, who were between the ages of 35 and 55 ($M = 45$, $SD = 6.6$) and with normal or corrected-to-normal vision, had a valid US driver’s license and at least 19 years of driving experience ($M = 30$), and drove at least five times a week at the time of data collection. During the experiment, participants drove six 15-min drives: four distraction drives and two baseline drives. During each distraction drive, participants completed four separate interactions with an auditory stock ticker with a one-minute break in between. The stock ticker task used simple auditory stimuli consisting of 3-letter stock names and 2-digit prices, but rendered high cognitive workload to drivers. It required participants continuously track the price changes of two different stocks and report the overall trend of the changes at the end. In the baseline drives, participants did not perform any secondary task. During all drives, participants were instructed to maintain vehicle position as close to the center of a straight lane as possible, to respond to the intermittent braking of a lead vehicle, and to report the appearance of bicyclists in the driving scene.

Eye movement and driving performance data were collected at a rate of 60 Hz using a Seeing Machines faceLAB™ eye tracker and the DriveSafety™ driving simulator, respectively. Raw eye movement data described the intersection coordinates of gaze vector on the simulator screen and then were transformed into a sequence of fixations, saccades, and smooth pursuits based on the speed and dispersion of the movements (Liang et al., 2007b). Then, we calculated the temporal and spatial measures of eye movements (Table 1). For fixations that occurred when the eyes are relative stationary (within 1°–2° visual angle), we calculated fixation duration and location (horizontal and vertical). For smooth pursuits that occur when people tracked moving objects (e.g., a passing vehicle), we calculated pursuit duration, distance, direction and speed. The driving performance measures included standard deviation of steering wheel angle and lane position, and steering error. All of these measures gauged driver ability of lateral control of the vehicle. Standard deviation (SD) of steering wheel angle and SD of lane position described the variance of steer steering movements by drivers and lane position of the vehicle. Steering error described the difference between the actual steering wheel position and the steering wheel position predicted by a second-order Taylor expansion (Nakayama et al., 1999). To obtain this measure, we first averaged steering wheel position across 0.2-s time window to reduce the noise in the signal, then applied the second-order Taylor expansion to predict mean steering wheel position in a current time window ($T$) from the values in the previous two time windows ($T-1$, $T-2$), finally calculated steering error, which was the absolute difference between the predicted steering wheel position and actual steering wheel position. Steering error measures the smoothness of steering wheel movements. The smaller steering error value is, the smoother drivers move the steering wheel, indicating the better driving performance. Finally, we summarized eye movement and driving performance measures across 30-s time windows. For the purpose of modeling, we defined “distraction” as the distraction drives in which the drivers performed the stock ticker task and “no distraction” as the baseline drives because the stock ticker task imposed high cognitive workload to drivers compared to the baseline (Reyes and Lee, 2008). Further information about the experiment and data reduction can be found in Liang et al. (2007b).

After reduction, each row in the data set, referred to as an instance, included 19 distraction indicators (i.e., continuous measures of driver visual behavior and driving performance summarized across 30 s, Table 1) and corresponding cognitive state of drivers in that period (“distraction” as 1, “no distraction” as 0). These 19 indicators were divided into three groups

<table>
<thead>
<tr>
<th>Groups</th>
<th>Distraction indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eye movement temporal measures</td>
<td>Blink frequency</td>
</tr>
<tr>
<td></td>
<td>Mean and standard deviation (SD) of:</td>
</tr>
<tr>
<td>Fixation duration</td>
<td>Pursuit duration</td>
</tr>
<tr>
<td>Pursuit distance</td>
<td>Pursuit direction</td>
</tr>
<tr>
<td>Pursuit speed</td>
<td>Percentage of the time spent on performing pursuit movements in each time window</td>
</tr>
<tr>
<td>Eye movement spatial measures</td>
<td>Mean and SD of:</td>
</tr>
<tr>
<td>Horizontal fixation location coordinates</td>
<td>Vertical fixation location coordinates</td>
</tr>
<tr>
<td>Driving performance measures</td>
<td>SD of steering wheel position</td>
</tr>
<tr>
<td></td>
<td>Mean steering error</td>
</tr>
<tr>
<td></td>
<td>SD of lane position</td>
</tr>
</tbody>
</table>

Table 1

Three groups of distraction indicators.
based on their correlation and meaning—eye movement temporal measures, eye movement spatial measures, and driving performance measures. In the resultant detection models, the output was drivers’ cognitive state, and the inputs were the 19 distraction indicators.

2.2. Training for the layered algorithm

We adopted a supervised clustering method in the lower layer to identify three cluster models: each from one group of distraction indicators (Fig. 2). In contrast with traditional unsupervised clustering methods, the supervised clustering method identifies clusters for a classified dataset so that the majority of cases in one cluster belongs to one class (Zeidat et al., 2006). For example, we identify three clusters based on the eye movement temporal measures: 95% of cases in two of the clusters belong to “distraction” and only 5% of cases belong to “no distraction” while in the third cluster there are 90% of cases of “no distraction” and 10% of cases of “distraction”. Therefore, this method minimizes cluster impurity (i.e., the percentage of the instances belong to a minor class of clusters) and the overall number of clusters. The following equation represents an example of the optimization problem of supervised clustering.

Minimize \( q(X) = \text{Impurity}(X) + \beta \cdot \text{Penalty}(k) \)

\[
\text{Impurity}(X) = \frac{\text{# of data in minor classes}}{n},
\]

\[
\text{Penalty}(k) = \begin{cases} 
\sqrt{(k - c)/n} & k \geq c \\
0 & k < c 
\end{cases},
\]

where \( X \) is a clustering solution, \( \beta \) is the weight to balance the impurity and the penalty of large number of clusters, \( k \) is the number of clusters in \( X \), \( n \) is the total number of training data, and \( c \) is the number of classes in the data (Zeidat et al., 2006). The cluster impurity reflects the percentage of the data in minor classes, which take smaller proportion of data in a cluster than another class (Eick et al., 2004). The number of clusters can be adjusted using the penalty term for large number of clusters (\( \beta \cdot \text{Penalty}(k) \)). Supervised clustering identifies multiple clusters for one class and may discover some heterogeneous effects of cognitive distraction by identifying more than one clusters for each cognitive state. We referred to identified clusters as feature behaviors for the cognitive state.

At the higher layer, \( H(t) \) and \( E(t) \) (i = 1, 2, 3, Fig. 2) represent driver cognitive state and corresponding behaviors at a time step \( t \). Three cluster models identify feature behaviors (\( E(t) \)) from three aspects of distraction indicators. The arrows represent the associations between cognitive state and behaviors, and the across-time arrow defines transitions between the cognitive states at two consecutive time steps.

We trained and tested detection models for each individual driver. First, we normalized performance measures by calculating z-scores. Then, we blocked the normalized data by two consecutive instances and assigned these blocks randomly into training and testing datasets. Both data sets contained multiple sequences of instances. Training data was composed of two thirds of the total data, and the remaining one third served as testing data. We trained the detection models with only the training data and used the testing data as “unseen” cases to evaluate the algorithms. For the layered algorithm, the training procedure included building three cluster models at the lower layer and training the DBN model at the higher layer. The cluster models were trained using SRIDHCR algorithm (Zeidat et al., 2006) programed with Matlab R2006b. The final number of clusters for the cluster models ranged between two and six across different drivers. The DBN model in the layered algorithm was trained using the Matlab toolbox (Murphy, 2004) and accompanying Bayesian Network structure learning package (LeRay, 2005).

2.3. Alternative algorithms

The layered algorithm was compared with (1) the original DBN with 19 distraction indicators as inputs and (2) the SVMs with the 19 distraction indicators at the last and current time steps as inputs. Adding distraction indicators at the previous time points to the SVM algorithms allows the SVMs to consider driver performance measures in two successive time steps. The inputs of this SVM models included 38 \((19 \times 2)\) continuous distraction indicators. The original DBNs were trained in the same way as the higher layer DBN in the layered algorithm using the Matlab toolboxes. For SVMs, we chose the Radial Basis Function (RBF) as the kernel function and searched for ideal parameters using 10-fold cross-validation. We trained and tested the SVM models using LIBSVM Matlab toolbox (Chang and Lin, 2001). The further information about training and testing SVMs could be found in (Liang et al., 2007b).

2.4. Algorithm evaluation

We evaluated the algorithms in terms of prediction effectiveness, computational efficiency, and interpretability. Prediction effectiveness was measured by detection accuracy, hit rate, false alarm rate, and \( d' \) and response bias used in signal detection theory (SDT) (Stanislaw and Todorov, 1999). \( d' \) represents the ability of the model to detect driver distraction. The larger the \( d' \)
value, the more effectively the model detects distraction. Response bias signifies how the model tends to under- or over-identify distraction. A value less than zero represents a tendency to overestimate driver distraction; and vice versa.

Computational efficiency measures included CPU time to train and test the models. The computer used was a SONY VAIO laptop with Intel® Core™2 CPU (T5500 @ 1.66 GHz) and 1 GB of RAM, running on Microsoft Windows XP Service Pack 3. The Matlab software ran with no other applications running at the same time.

Interpretability was gauged by the strength of dependencies between performance measures and driver cognitive state calculated based on the resultant models. We used the normalized variant of the mutual information (denoted by $C_{XY}$), also called coefficients of constraint or uncertainty coefficient.

$$C_{XY} = I(X;Y) / H(X) = 1 - H(Y|X)/H(Y),$$

where $X$, and $Y$ are two random variables, $I(X;Y)$ represents mutual information of $Y$ given $X$, $H(X)$ is the entropy of $X$, $H(Y|X)$ is the entropy of $Y$ given $X$, and $H(Y)$ is the entropy of $Y$. Mutual information, $I(X;Y)$, describes the information shared by two random variables (Guhe et al., 2005); that is, how much uncertainty of $Y$ is reduced by knowing $X$. Its normalized variant describes the percentage of the uncertainty of $Y$ is reduced by knowing $X$. This measure could be calculated based on the conditional dependent relationship learned by the BN algorithm. In this evaluation, $X$ represents each of categories of feature behavior identified by each of three cluster models for the layered algorithm or each of 19 distraction indicators for the original DBN, and $Y$ represents driver cognitive state. The higher $C_{XY}$ is, the more indicative the feature behavior for the layered algorithm or performance measure for the original DBN was to driver cognitive state. Because the SVM algorithm does not provide the relationships learned from data in an interpretable way, interpretability of the resultant model was quite low and cannot be compared with other two algorithms.

3. Results and discussion

3.1. Prediction effectiveness and computational efficiency

We used Friedman’s non-parametric tests (Gibbons, 1993) to compare each evaluation measure across the three types of algorithms. The layered, original DBN, and SVM algorithms achieved similar prediction effectiveness: all five measures were not statistically different between different algorithms (Table 2). However, the training and testing time was much shorter for the layered and SVM algorithms compared to the original DBN algorithm (Table 2). To train the layered or SVM algorithm for each driver required 13–17 s on average, in contrast to 1146 s (19.1 min) required for the original DBNs. To test the algorithms with the same testing dataset required 0.95 s for the layered algorithm, 0.17 s for SVM algorithm, and 5.91 s for the original DBN.

Although detection performance of the layered algorithm was similar to the other two algorithms, the layered algorithm improved computational efficiency from the original DBNs and can be more practical for distraction mitigation. For example, based on CPU time for testing, the original DBNs take as much as five times longer than the layered algorithm to identify cognitive distraction. In some real-world driving situations where cognitive distraction plays a critical role in diminishing safety, distraction must be detected quickly to avoid safety mishaps. Timely, accurate detection is an essential evaluation criterion for distraction mitigation.

At the same time, the results that the layered algorithm improved computational efficiency and maintained detection performance, showing that the supervised clustering method can effectively integrate a number of distraction indicators. Compared with the original DBN algorithm with 19 inputs, the DBN in the layered algorithm required only the three inputs to achieve similar results. This means that these three inputs carried a similar amount of useful information to indicate driver cognitive state as the 19 indicators. The supervised clustering method can capture the feature behaviors of distraction accurately because the method takes into account not only the distribution of data, but also the impurity of the resultant clusters. Therefore, supervised clustering is a useful approach to reduce the number of features for computationally intensive methods.

Compared with the layered algorithm, the SVM algorithm required a similar amount of CPU time for training and needed even less time to make a prediction, which presents one advantage of SVMs in this application. However, the SVM detection model was a black box and did not provide any useful information to interpret the relationship between distraction indicators and cognitive state of drivers.

<table>
<thead>
<tr>
<th>Layered algorithm mean (SD)</th>
<th>SVMs mean (SD)</th>
<th>DBNs mean (SD)</th>
<th>Friedman’s test $\chi^2$ (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88 (8)</td>
<td>90 (5)</td>
<td>88 (16)</td>
<td>0.67 (0.72)</td>
</tr>
<tr>
<td>$d'$</td>
<td>3.50 (1.81)</td>
<td>3.06 (1.37)</td>
<td>4.80 (2.54)</td>
</tr>
<tr>
<td>Response bias</td>
<td>1.82 (4.38)</td>
<td>−1.36 (3.64)</td>
<td>−0.22 (4.47)</td>
</tr>
<tr>
<td>Hit rate</td>
<td>0.88 (0.08)</td>
<td>0.94 (0.04)</td>
<td>0.92 (0.14)</td>
</tr>
<tr>
<td>False alarm rate</td>
<td>0.14 (0.16)</td>
<td>0.23 (0.14)</td>
<td>0.16 (0.23)</td>
</tr>
<tr>
<td>CPU time for training (s)</td>
<td>13 (2)</td>
<td>17 (4)</td>
<td>1146 (131)</td>
</tr>
<tr>
<td>CPU time for testing (s)</td>
<td>0.95 (0.12)</td>
<td>0.17 (0.03)</td>
<td>5.91 (0.73)</td>
</tr>
</tbody>
</table>
3.2. Interpretability

For the layered algorithm, the analysis of the normalized variants of mutual information for three behavioral characteristics was consistent with the results for the original DBN algorithm. The previous results with the original DBN algorithm shows that blink frequency is the most indicative measure and spatial distribution of eye movements and fixation duration also signifies driver cognitive state (Liang et al., 2007a). This study found that eye movement temporal measures including blink frequency and fixation duration had the highest normalized variant (56%), followed by eye movement spatial measures (45%) and driving performance measures (only 23%). This suggests that the layered algorithms captured similar information from data as the single-layered DBN algorithms, which may explain the similar prediction performance for the layered and original DBN algorithms. Nonetheless, it was impossible to extract such information about the relationships between driver behavior and cognitive states from the SVM models.

More specifically, studying layered algorithms could clarify some aspects of cognitive distraction that had not been revealed in traditional statistical analysis (e.g., ANOVA). An example model trained with the data from one driver illustrates
this benefit (Fig. 3). We focused on each aspect of the driver behavior (i.e., three groups of distraction indicators) and identify
the meaning of the feature behaviors regarding cognitive state and their generalizability to other drivers.

A cluster model of eye movement temporal characteristics produced three feature behaviors (Tem-ND, Tem-D1, and
Tem-D2 in Fig. 3). The first behavior (Tem-ND) was primarily comprised of “no distraction” cases, and the second and third
behaviors (Tem-D1 and Tem-D2) “distraction” cases. The feature behavior of no-distraction had lower blink frequency
compared to the behaviors of distraction, indicating that drivers tend to blink more frequently when distracted. This may
indicate diminished attention to visual control, which can increase involuntary eye movements and disrupt consolidation
of visual information (Strayer et al., 2003).

Another eye movement temporal measure, fixation duration, showed a bidirectional effect of cognitive distraction.
Although both indicating cognitive distraction, the first feature behavior of distraction (Tem-D1) presented a relatively long
duration and the second feature behavior of distraction (Tem-D2) represented a relatively short duration of fixations com-
pared to the feature behavior of no-distraction. More interestingly, these two feature behaviors of distraction occurred at the
different rates; longer fixation durations were 1.7 times more likely to occur than shorter fixation durations (Within-time
transition A in Fig. 3, Tem-D1: 35%, Tem-D2: 60%). We paired the experimental conditions and the feature behaviors by time
(Fig. 4) to examine the circumstances when eye behavior presents either one of these two feature behaviors of cognitive dis-
traction and found that the instances defined as “distraction” (i.e., in the drives with secondary tasks), but occurring during
the one-minute breaks between the secondary tasks (the light shaded areas in Fig. 4) were mostly labeled as the behavior
with shorter fixation duration, that is Temp-D1 (Tem-D1: 78%, Tem-D2: 22%). It suggests that this feature behavior of distri-
ction represents intermediary behavior of cognitive distraction, or possibly a recovery process as drivers restore their sit-
uation awareness. The behavior characterized by longer fixation duration (Tem-D2) represented a typical pattern of eye
movements during cognitive distraction or when the driver was fully engaged in the secondary task. The behavior with
shorter fixation duration (Tem-D1) may depict a transitional behavior when the driver started to become, but was not fully,
engaged in the task or during a short period after the task finished. This bidirectional effect of cognitive distraction could not
be discovered with traditional statistical analysis, like ANOVA. However, the effect of cognitive distraction on fixation dura-
tion varied substantially between drivers—some drivers had longer fixation duration and others had shorter duration when
distracted.

The cluster model associated with eye movement spatial measures produced three clusters (Spa-ND, Spa-D1, and Spa-D2
in Fig. 3). The first feature behavior (Spa-ND) primarily comprised of “no distraction” cases, and the other two (Spa-D1, Spa-
D2) primarily comprised of “distraction” cases. Both feature behaviors of distraction indicated that drivers tended to look
down during distraction, illustrated by larger vertical position of fixation (meanly). It suggests that during distraction these
drivers focused on the roadway close to the their vehicle, but not straight ahead. This pattern could reduce the drivers’ capa-
bility to foresee the driving situation. But this effect varied across individuals. Among nine drivers, three drivers tended to
look down, and two drivers tended to look up during distraction, suggesting that driver eye-gaze patterns are somewhat idi-
osyncratic when visual scanning is disrupted by cognitive load (Harbluk et al., 2007; Victor et al., 2005).

Finally, the cluster model built from driving performance measures produced three clusters (Dri-ND1, Dri-ND2, and Dri-D
in Fig. 3). The first two behaviors featured “no-distraction” cases, and the last one featured “distraction” cases. The com-
parison between the first feature behavior of no-distraction and the feature behavior of distraction suggested that the driver
steered more abruptly during cognitive distraction than no-distraction even when the steering angle changed in a similar
range. Although sharing similar steering-angle variance (std_steer), the feature behavior of distraction (Dri-D) had larger
steering error than that of no-distraction (Dri-ND1). This effect was found in six out of nine drivers. Meanwhile, the DBN
model at the higher layer showed that clusters three feature behaviors could occur during “no distraction” (Within-time
transition C in Fig. 3, Dri-ND1: 0.45; Dri-ND2: 0.35; Dri-D: 0.20), while the feature behavior of distraction (Dri-D) occurred

![Fig. 4. The prediction of feature behaviors of the eye movement temporal measures.](image-url)
predominantly over the other two during "distraction" (Within-time transition C in Fig. 3, Dri-ND1: 0.14; Dri-ND2: 0.09; Dri-D: 0.77). It suggests that when drivers are not distracted, their driving performance varies substantially, but when they are distracted, their performance is more regular. It may reflect driver’s ability to employ many strategies to support satisfactory performance when demands are low, but relatively few strategies support satisfactory performance when demands are high (Goodrich et al., 1998). The transitions between cognitive states across time was an identity matrix because the definition of distraction used in this study led to no natural transition between the cognitive states of drivers in training data.

In summary, the layered algorithm significantly improved the computational efficiency from the original DBN algorithm in detecting driver cognitive distraction. In the layered algorithm, the supervised clustering models at the lower layer effectively integrated 19 distraction indicators into three feature behaviors, which differentiated “distraction” and “no-distraction” states. At the higher layer, a DBN algorithm received only three inputs, instead of 19 for the original DBN algorithm, were trained and made prediction much faster. In practice, the layered algorithm carries significant advantage over the original DBN algorithm to achieve timely and accurate detection to support mitigation strategies for cognitive distraction.

Moreover, studying the trained layered algorithms revealed that the temporal characteristics of eye movement were the most predictive indicators to cognitive distraction, followed by the spatial characteristics of eye movements and driving performance measures. These results were consistent with the findings from studying the trained original DBN algorithms. Also, the layered algorithms reveal some aspects of cognitive distraction that had not been revealed in traditional statistical analysis. This information can be used to guide future research on how drivers react to cognitive workload while driving and to help engineers to focus on the most predictive indicators of distraction when developing adaptive distraction mitigation systems. Examining the layered algorithms for different drivers also demonstrated great individual differences among drivers under cognitive distraction.

The training data used in this study were obtained from an simulator-based experiment and lacked natural transition between the cognitive states of drivers. The future studies may consider data collected under a more naturalistic driving setting. Data mining methods, such as those used in this study, are particularly important in understanding naturalistic drivering data (Chong et al., 2013; McDonald et al., in press).

4. Conclusions

Based on the results, although the layered algorithm did not improve cognitive distraction detection accuracy, but it did significantly improved computational efficiency. The layered algorithm also provides useful insights concerning the effects of cognitive distraction on driver behavior, which have no equivalent in the SVM algorithm and other traditional statistical tests. This study demonstrated that data mining methods can identify human cognitive state from eye glance behavior and driving performance.

Acknowledgments

A subset of results will be presented at the 7th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Bolton Landing, New York. The authors are grateful to William J. Horrey, Yueng-hsiang Huang, and Marvin Dainoff for comments on this manuscript and to Michelle L. Reyes for her effort in data collection.

References


