Probabilistic models for analysis of performed task based on eye movement

Author: Andrej Vítek (andrej2024@gmail.com)
Supervisor: Dr. Jozef Tvarožek
Slovak University of Technology

Motivation

Problem:
- Eye movements and gaze patterns contain a lot of information about the observer, performed task and visual stimuli
- Models which are using gaze, are often based only on the evaluation of **aggregated** metrics of gaze, thereby losing temporal information

Solution:
- Create additional features by using probabilistic models, which could abstract temporal information from eye-tracking data
- Use created features to improve results of standard models

Contributions

Creation and evaluation of new method:
- Shown improvement by additional 5-7 % over baseline

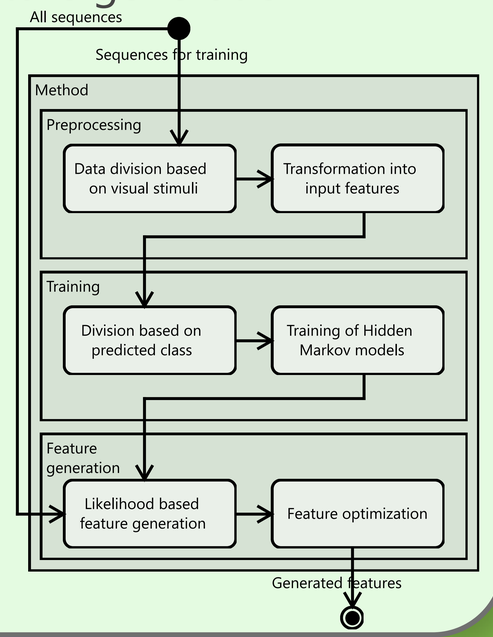
Analysis of features:
- Domain of **code reading**
- Feature importance per type of code complexity

Method for feature generation

Uses **Hidden Markov models**:
- Data based approach
- Sequence based learning

Models trained using combination of **input features**:
- Fixation coordinates
- Saccade length
- Saccade angle (relative/absolute)
- Fixation duration

Features created by using likelihood measure (Sequence generation probability)



Datasets

Properties:
- Eye-tracking
- Tasks given in same order

Dataset	Participants	Programs
1	55	4(6)
2	135	15(45)
3	139	15(45)
4	114	15(45)
5	111	15(45)

Task:
- Code comprehension
- Write correct output after execution

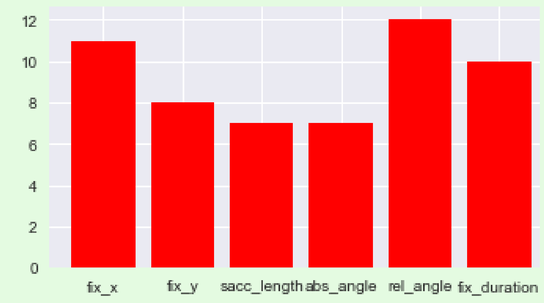
```
1 // Uvažujte funkciu gn(). Čo vypíše nasledujúci program?
2
3 int gn( int aa, int bb )
4 {
5     int cc = aa + bb;
6
7     bb = cc / aa;
8
9     return bb + cc;
10 }
11
12
13 int main()
14 {
15     printf("%d", gn( 3, 2 ));
16
17     return 0;
18 }
19
```

Analysis of features

Programs in datasets divided into groups based on their type of **complexity**:
- Arithmetic based
- Logic based
- Loops based
- Argument based

Analysed features importance for each type of complexity (features = input features)

Example (Arithmetic complexity):



Results

Model task:
- Prediction of answer correctness by using eye-tracking data

Evaluated with multiple classifiers:
- Gradient boosted trees
- Support vector machine
- Logistic regression

Results compared with baseline model which only uses aggregated features

Greatest improvement is achieved by Gradient boosted trees classifier:
- Improvement by additional 5-7 % over baseline model