Task Classification Model for Visual Fixation, Exploration, and Search

Ayush Kumar
Stony Brook University
aykumar@cs.stonybrook.edu

Anjul Tyagi
Stony Brook University
aktyagi@cs.stonybrook.edu

Michael Burch
Eindhoven University of Technology
m.burch@tue.nl

Daniel Weiskopf
University of Stuttgart
Daniel.Weiskopf@visus.uni-stuttgart.de

Klaus Mueller
Stony Brook University
mueller@cs.stonybrook.edu

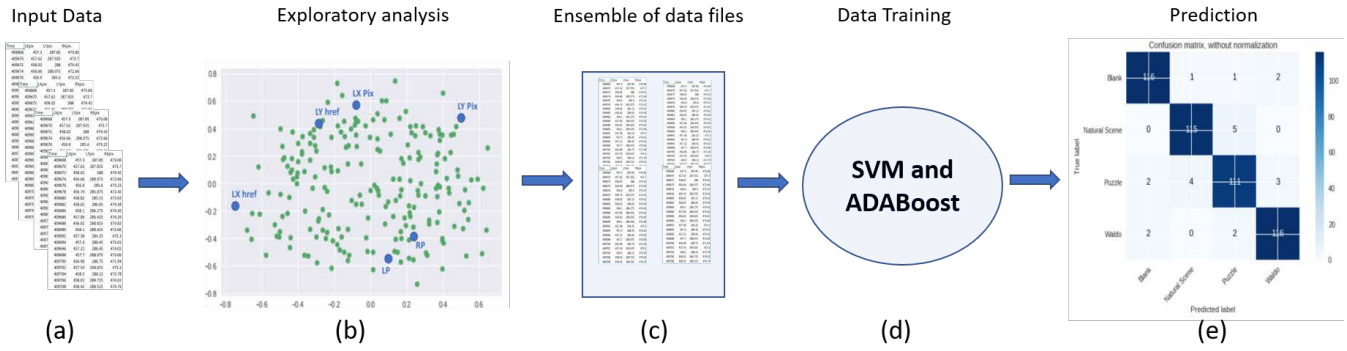

Figure 1: Overview of the classifier trained for task prediction. (a) Combine and shuffle the input files for training in the next stage of exploratory analysis. (b) Feature selection to be done in this stage. (c) Feed the task-specific user file with selected features into the trained classifier. (d) Classifier predictions are analyzed in the form of a confusion matrix shown in (e).

ABSTRACT

Yarbus' claim to decode the observer's task from eye movements has received mixed reactions. In this paper, we have supported the hypothesis that it is possible to decode the task. We conducted an exploratory analysis on the dataset by projecting features and data points into a scatter plot to visualize the nuance properties for each task. Following this analysis, we eliminated highly correlated features before training an SVM and Ada Boosting classifier to predict the tasks from this filtered eye movements data. We achieve an accuracy of 95.4% on this task classification problem and hence, support the hypothesis that task classification is possible from a user's eye movement data.

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; • **Computing methodologies** → **Machine learning algorithms**.

ETRA '19, June 25–28, 2019, Denver , CO, USA

ACM ISBN 978-1-4503-6709-7/19/06…$15.00
https://doi.org/10.1145/3314111.3323073

KEYWORDS

Classifier, eye movements, Yarbus, task decoding, visual attention

ACM Reference Format:
Ayush Kumar, Anjul Tyagi, Michael Burch, Daniel Weiskopf, and Klaus Mueller. 2019. Task Classification Model for Visual Fixation, Exploration, and Search. In *2019 Symposium on Eye Tracking Research and Applications (ETRA '19), June 25–28, 2019, Denver , CO, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3314111.3323073

1 INTRODUCTION

Visual attention is one of the most sought area in the field of computer vision and is used in day to day tasks. Extensive studies have been conducted to predict users' behavior or pattern from the visual attention of the observers. Kumar et al. [Kumar et al. 2018] used multi-metric grouping of eye movements to find out subjects leading to distinct visual groups of similar behavior. Yarbus' [Yarbus 2013] claim of predicting the observer's task is one such attempt to use visual attention as a cue to look deep into human cognition. Yarbus was the first one to establish the relationship between eye movements and human cognition, which he formed as a basis of his claim. However, Yarbus' claim to decode the observer's task from eye movements has received mixed reaction. Green et al. [Greene et al. 2012] have reconsidered Yarbus' work and argued against his claim, whereas Borji et al. [Borji and Itti 2014] defended Yarbus and supported their claim with their study. In this paper, we try different classification algorithms to support Yarbus' claim. The

results show that task classification is actually possible to predict from eye movement data with an accuracy of 95% using modern classification techniques in machine learning. To support this claim, we have used a dataset from an extensive study carried out by Otero-Millan et al. [Otero-Millan et al. 2008].

The rest of the paper is organized as follows: Section 2 discusses the exploratory analysis part, where we use Data Context Map[Cheng and Mueller 2016] to visualize task specific details in the dataset and for feature selection. Then, Section 3 discusses the proposed classifiers to classify the task on the basis of their eye movements. Section 4 presents the result in the form of a confusion matrix which contains accuracy results. Finally, Section 5 discusses the future work and gives concluding remarks.

2 VISUALIZATION WITH DATA CONTEXT MAP

The data context map (DCM) [Cheng and Mueller 2016] is a tool built to visualize high-dimensional data in the form of a 2D scatter plot. DCM is a variation of the multiple correspondence analysis technique described in [Tyagi et al. 2018] and is used for numerical data. DCM projection tries to preserve the correlation between variables and the data points at the same time. The plot consists of both, variables and data points represented in a scatter plot and the relative distance between the variables and the data points signifies correlation. An example DCM for variables LP, RP, LX href, LY href, LX Pix and LY Pix is shown in Figure 2. The data context map uses multidimensional scaling (MDS) [Kruskal 1964] to project the data points and the variables into lower dimensions. The distance matrix in MDS corresponds to the correlations in case of inter variable correlation and Euclidean distances in case of inter data correlation. For finding correlation between variables and the datapoints, the variables are treated as a data point with value 1 in the corresponding dimension and zero for all other dimensions. All of these distances are then fused into a single matrix and the matrix is normalized before projecting with MDS.

2.1 Analysis

Data context map can point out subtle relationships between variables and the dataset. We evaluated Data Context Maps for each of the tasks in the datasets to point out interesting correlations and for feature selection. An example representation is shown in Figure 2 where we show the context map on *Finding Waldo* task. We can see the that *LP (Left Pupil)* and *RP (Right Pupil)* are close to each other in the projection. This shows that LP and RP have a high correlation. It is also interesting to note that *LX Pix* and *LX href* readings show opposite relations to that of *LP* and *RP* as they show no correlation, since being located far from each other on the context map.

Considering the positioning of data points, it is interesting to note that very few data points lie close to the *LX href* on the context map in Figure 2 . This shows that the *LX href* readings had the least correlation with the data points. This can be understood with the concept of randomness in a variable, as more random the distribution of a variable is, the less is it's correlation with respect to a set of data points. This analysis not only allows for visualizing inherent correlations, but also aids in variable filtering. After

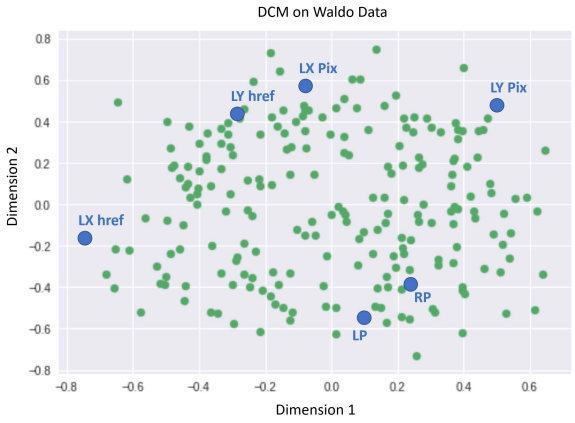

Figure 2: Data Context Map [Cheng and Mueller 2016] projection of Finding Waldo task on a randomly chosen user trial. Green points represent mapping of eye movement readings on a particular timestamp. Blue points represent corresponding variables in the dataset.

carefully analyzing the Data Context Maps of all the four tasks, we found out that the least five correlated variables are *LX Pix, LY Pix, LX href, LY href and LP* , which we later used to train the classifiers

3 LEARNING TASK CLASSIFICATION

In this task classification, we designed our study to classify four tasks from the fixation dataset. We choose gaze data along with the pupil diameter as analyzed from the the data context map visualizations for training the classifiers. As part of data preprocessing, we created an ensemble of the data by merging the files of all tasks and for all users. Each row of this merged file, which is a reading of eye movements of a user for a specific task at a timestamp was labelled with the corresponding task label. After shuffling all these rows with labels, we generated the training data where each datapoint is the eye movement reading of a user at a timestamp and the label is the label for the task. After the files are merged the dataset is standardized by calculating the z-score for each column in the dataset. As shown in Equation 1, zscore is a method to convert a data distribution to standard normal by subtracting the mean μ and dividing by the standard deviation σ for each column distribution in the dataset.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

We used zscore instead of Min Max normalization since zscore is more robust towards scaling the outliers. Now that the data is normalized, we shuffle the data points and split it to train, test, and validation sets. The testing and validation sets contain about fifteen percent of the total data points each in our experiment. A portion of this normalized data was used to train the two classifiers, *SVM classifier* and *Ada Boosting classifier with decision tree as the base classifier.* By using the readings of *LXPix, LYPix, LX href, LY href, and LP* we could train classification models which can predict the type of task being performed, given the user's readings over time for any of the tasks.

3.1 Method: SVM Classifier

Support Vector Machines [Cortes and Vapnik 1995] is a well studied classifier built to work with binary classification tasks. The main objective is to find the hyper plane which can separate the data points. The data points can be projected to higher dimensions for easier separation using the kernel function. For multiclass classification, one vs. rest classification [Bottou et al. 1994] technique is designed to use multiple binary classifiers to be trained for every label in the dataset [Hsu and Lin 2002]. Thus, in this case, four different SVM classifiers are trained to each of the four task classifications. The i^{th} SVM is trained with the samples of the i^{th} class as positive and all others as negative. For the given training data $(x_1, y_1), ..., (x_l, y_l)$ with the labels in class from *1* to *k*, the multiclass SVM solves the problem shown in Equation 2.

$$\begin{aligned} \min_{\omega^i, b^i, \epsilon^i} \quad & \frac{1}{2}(\omega^i)^T \omega^i + C \sum_{j=1}^{l} \epsilon_j^i (\omega^i)^T \\ & (\omega^i)^T \phi(x_j) + b^i \geq 1 - \epsilon_j^i \qquad y_j = i \qquad (2) \\ & (\omega^i)^T \phi(x_j) + b^i \leq -1 + \epsilon_j^i \qquad y_j \neq i \\ & \epsilon_j^i \geq 0 \qquad j = 1, ..., l \end{aligned}$$

Minimizing the above function separates the positive and negative labels by maximum margin which can be controlled with the cost term C. In our experiment, the value of C = 1,000 gave the best accuracy of around 80%. We trained SVM on 77,000 data points sampled from the merged dataset with stratified sampling to keep the number of samples from each task class consistent with other tasks. ϕ is the kernel function which is used to map the data points from lower to higher dimensions for a better separation of the data points. In our experiment, we use the RBF kernel shown in Equation 3 which is a linear combination of non-linear interpolations of the input to achieve the highest accuracy with SVMs on this task. The trained SVM classifier is used to predict the task from user trials by taking the mode of predicted classes for all samples.

$$f(x) = \sum_{i=1}^{n} \alpha_i g(\|x - x_i\|). \qquad (3)$$

3.2 Method: Adaptive Boosting

Adaptive Boosting [Freund et al. 1996] is one of the boosting techniques where the main idea is to use an ensemble of classifiers to train on the same dataset. It is based on the concept that training many simple classifiers of accuracy above 50%, the majority voting of these simple classifiers for each data point is likely to produce better classification results than a complex single classifier. Each classifier is trained to perform better in classifying the training data points which were misclassified by the previous classifiers. Initially, the probability of picking each sample from the dataset is set to $1/N$. After each iteration, when a trained classifier is added to the ensemble, the probability of generating each sample for the next classifier is recalculated. Let ϵ_k be the sum of probabilities of all the misclassified instances for the classifier C_k. Then for the classifier C_{K+1}, the probability of picking the incorrectly classified samples by C_k is increased by a factor of $\beta_k = (1 - \epsilon_k)/\epsilon_k$. These probabilities are then normalized so that the sum of probabilities equals 1 [Opitz and Maclin 1999]. For the base classifier in Ada Boosting, we used Decision Trees with a max depth constraint equal to 6. Trained Ada Boost classifier is used to predict the task from user trials by taking the mode of predicted classes for all samples. Training an ensemble of 100 classifiers with Ada Boosting on 77,000 data points resulted in the accuracy range of 93% - 95% in the user task classification. Accuracy comparison between SVM and ADA Boosting is shown in Table 1.

Table 1: Accuracy of user task classification

Model	Accuracy	Parameters
SVM	31.2%	Linear Kernel with C = 1,000
SVM	80.3%	RBF Kernel with C = 1,000
Ada Boosting	**95.4%**	Number of estimators = 100, decision tree max depth = 6

4 RESULTS

As shown in Table 1, Ada Boosting with a base classifier as decision tree performed better than SVM in this case. The decision tree separates the data based on attributes to multiple dimensions whereas SVM uses a nonlinear kernel to map the data to higher dimensions. Since a decision tree classifier with boosting works better for this dataset, it suggests that the variables *LX Pix, LY Pix, LX href, LY href, and LP* can be separated based on some intervals of their values for each of the tasks. The detailed classification results can be seen in the form of a confusion matrix in Figure 3. The least number of misclassified samples were from the *Blank* task with just 4 users being misclassified. This is reasonable because every blank scene would have similar patterns for every user and it is easy to classify. The tasks *Finding Waldo* and *Natural Scene* both had 5 of the users misclassified followed by the task *Puzzle* with most number of misclassifications counting to 8. This shows that the task *Puzzle* is more specific to the task being performed as compared to the users' behavior of looking at a scene. Different puzzles require different ways to compare for each user and thus makes it harder for the classifier to predict the task based on just the user behavior.

5 CONCLUSION

In this paper, we have supported the hypothesis that it is possible to decode the task, which concludes that eye gazes from eye movement data, carry cognitive information such as a mental state which is highly related with the task the observer is carrying out. Using different classification algorithms we successfully predicted the observer's task from eye movement data. We used SVM and Ada Boosting classifiers with an accuracy of 80.3% and 95.4%, respectively. While our results are performing better than most of the classifiers used for the similar task, there is a possibility of obtaining better accuracy by selecting more relevant features and other classification algorithms in the literature. Also for this study, we removed the blinking samples from the dataset but it can be interesting to study the blinking patterns for future work. We will be running our classification model on a dataset from Green et al. [Greene et al. 2012] and Yarbus [Yarbus 2013] to support their claim too and compare the accuracy.

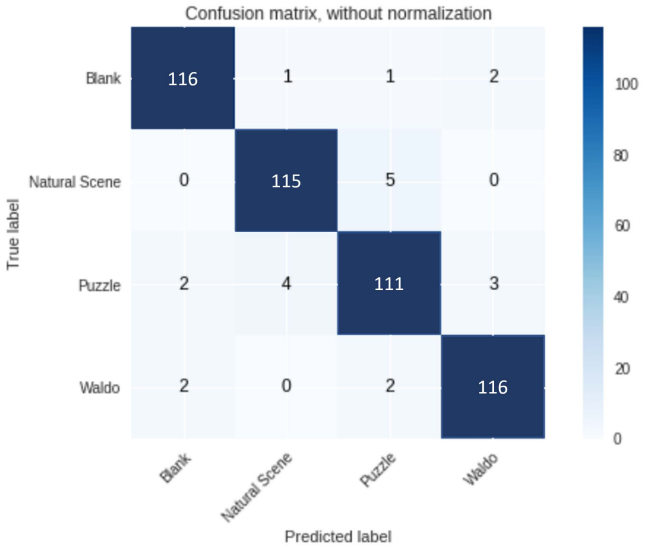

Figure 3: Confusion matrix of classification results from Ada Boosting with Decision Trees.

ACKNOWLEDGMENTS

This research was partially supported by NSF grant IIS 1527200 and MSIT, Korea, under the ICT Consilience Creative program (IITP-2019-H8601-15-1011) supervised by the IITP.

REFERENCES

Ali Borji and Laurent Itti. 2014. Defending Yarbus: Eye movements reveal observers' task. *Journal of vision* 14, 3 (2014), 29–29.

Léon Bottou, Corinna Cortes, V Vapnik, JS Denker, H Drucker, I Guyon, LD Jackel, Y LeCun, UA Muller, E Sackinger, et al. 1994. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of 12th International Conference on Pattern Recognition*. IEEE, 77–82.

Shenghui Cheng and Klaus Mueller. 2016. The data context map: Fusing data and attributes into a unified display. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 121–130.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *icml*, Vol. 96. Citeseer, 148–156.

Michelle R Greene, Tommy Liu, and Jeremy M Wolfe. 2012. Reconsidering Yarbus: A failure to predict observersâĂŹ task from eye movement patterns. *Vision research* 62 (2012), 1–8.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks* 13, 2 (2002), 415–425.

Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1 (1964), 1–27.

Ayush Kumar, Rudolf Netzel, Michael Burch, Daniel Weiskopf, and Klaus Mueller. 2018. Visual Multi-Metric Grouping of Eye-Tracking Data. *Journal of Eye Movement Research* 10, 5 (2018), 11.

David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research* 11 (1999), 169–198.

Jorge Otero-Millan, Xoana G Troncoso, Stephen L Macknik, Ignacio Serrano-Pedraza, and Susana Martinez-Conde. 2008. Saccades and microsaccades during visual fixation, exploration, and search: foundations for a common saccadic generator. *Journal of vision* 8, 14 (2008), 21–21.

Anjul Tyagi, Ayush Kumar, Anshul Gandhi, and Klaus Mueller. 2018. Road Accidents in the UK (Analysis and Visualization). IEEE VIS.

Alfred L Yarbus. 2013. *Eye movements and vision*. Springer.