Predicting the Valence of a Scene from Observers' Eye Movements

Abhishek.L (abhishek.ln.19@gmail.com) , Aravindhan.K (aravindhank11@gmail.com)

Abstract

Multimedia analysis benefits from understanding the emotional content of a scene in a variety of tasks such as video genre classification and content-based image retrieval. Recently, there has been an increasing interest in applying human bio-signals, particularly eye movements, to recognize the emotional gist of a scene such as its valence. In order to determine the emotional category of images using eye movements, the existing methods often learn a classifier using several features that are extracted from eye movements. Although it has been shown that eye movement is potentially useful for recognition of scene valence, the contribution of each feature is not well-studied. To address the issue, the contribution of features extracted from eye movements have been studied in the classification of images into pleasant, neutral, and unpleasant categories. Ten features and their fusion are assessed. The features are histogram of saccade orientation, histogram of saccade slope, histogram of saccade length, histogram of saccade duration, histogram of saccade velocity, histogram of fixation duration, fixation histogram, top-ten salient coordinates, and saliency map.

Introduction

The paper[1] utilizes machine learning approach to analyze the performance of features by learning a support vector machine and exploiting various feature fusion schemes. The experiments reveal that 'saliency map', 'fixation histogram', 'histogram of fixation duration', and 'histogram of saccade slope' are the most contributing features. The selected features signify the influence of fixation information and angular behavior of eye movements in the recognition of the valence of images.

In this paper, the dataset is studied more in detail and more machine learning approaches are used to perfectly classify the image as pleasant, neutral or unpleasant. The classifiers used in this paper to learn the dataset are linear support vector machine, K-Nearest Neighbours classifiers and Extra Trees Classifiers. Ensemble is taken for the above classifiers for better accuracy.

Features

There are totally 872 attributes in the feature vector which are used to describe all the features. Effectively, there are 10 salient features used to detect the eye movements. Among which, each feature is represented with different number of attributes thus comprising of a total of 872 attributes.

Effective Feature	Attribute Range
Velocity	1 - 50
Saccade Orientation	51 - 86
Saccade Length	87 - 136

Slope	137 - 166
Saccade Duration	167 - 226
Fixation Duration	227 - 286
Fixation Histogram	287 - 542
Saliency Histogram	543 - 552
Saliency Map	553 - 852
Top 10 coordinates	853 - 872

Dataset

The dataset contains a total of 95 images for which the users' feedback is retrieved along with their eye movements. With help of the users' feedback, these images are assigned labels as pleasant, neutral or unpleasant. So, now the dataset we have comprises of 95 records with 872 attributes and one attribute that has the class labels.

The following picture tells us about the images and its assigned class labels. This gives us the information about the number of images in each label.

Valence	Image ID
Neutral	'2020', '2102', '2104', '2130', '2190', '2200', '2271', '2272', '2280', '2210', '2214', '2215', '2220', '2221', '2230', '2305', '2357', '2372', '2383', '2385', '2393', '2396', '2397', '2435', '2441', '2485', '2487', '2491', '2493', '2495', '2499', '2512', '2513', '2516', '2520', '2595', '2635', '2690', '2704', '2749', '2770', '2780', '2795', '2830', '2840', '2870', '7506'
Pleasant	'1340', '1999', '2000', '2010', '2037', '2091', '2092', '2154', '2222', '2304', '2339', '2340', '2341', '2358', '2362', '2391', '2501', '2530', '2620', '2650', '4617', '5410', '7325', '8497'
Unpleasant	'2095', '2110', '2120', '2141', '2205', '2276', '2278', '2490', '2590', '2691', '2710', '2750', '3500', '3530', '4621', '6243', '6313', '6315', '6360', '6370', '6530', '6550', '6560', '6561'

The dataset has a total of 24 pleasant and 24 unpleasant images. The rest 47 images are labelled as neutral. Since, the dataset has equal number of pleasant and unpleasant images, the dataset is unbiased and hence no significant preprocessing is required and hence suitable for training with different ML classifiers.

Variety in Dataset

The dataset contains images from different categories so that the dataset does not become biased with user's interest. If the dataset has contained images of only one or a few categories which is liked by certain users, then it gets a more probability of being marked as pleasant and hence the dataset becomes biased in one way.

But this issue is avoided as the dataset contains images of many categories and also equally distributed.

Image Category	Number of Images			
	unpleasant	pleasant	neutral	all
Building	0	0	2	2
Food	0	2	2	4
Baby	0	8	0	8
Rotten	8	0	0	8
Abstract & Conceptual	3	0	6	9
Animals	0	11	6	17
Wild Animals	8	2	7	17
Nature	4	15	7	26
Objects	3	0	24	27
Nude & Porn	0	9	19	28
Activity	6	7	19	32
People & Daily Activity	29	24	56	109
Total	61	78	148	287

The following valence category is obtained by the emotional mean valence values reported by IAPS.

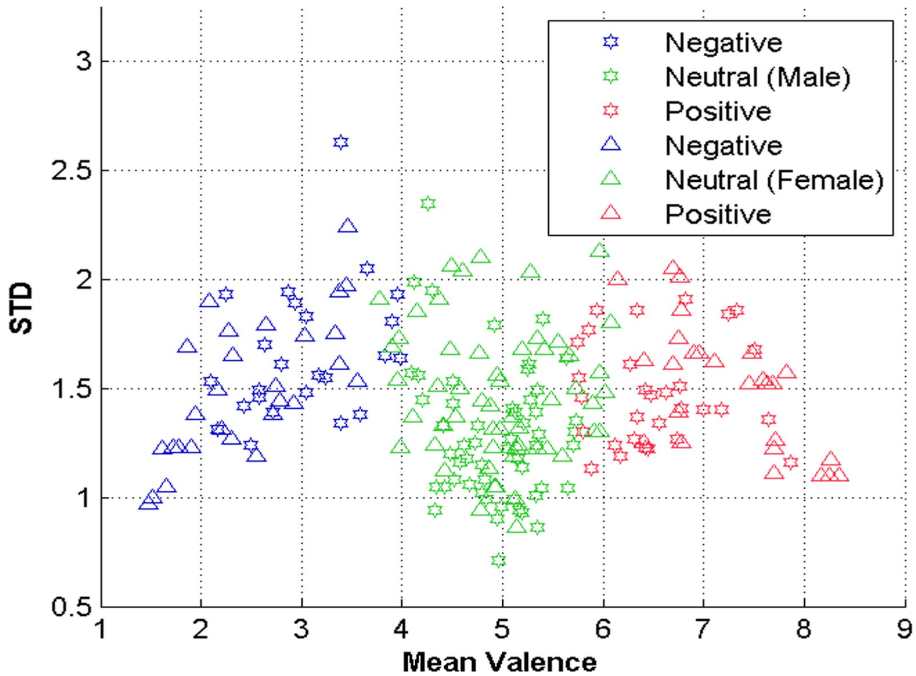
doi:10.1371/journal.pone.0138198.t001

Sample images in the Dataset

Data Distribution

The following figure is the plot between the mean valence and the standard deviation. This helps us in understanding how the data is scattered and also the role played by the gender in assigning the class labels.

This plot clearly tells that the data gathered is equally distributed between both genders and hence not biased towards any particular gender. In this way, the dataset becomes perfectly optimal to learn with any known classifier.



Algorithm

The original paper has applied three SVM kernels of linear, polynomial and radial bias function(RBF), They have considered three kernels because the stochastic nature of evolutionary methods for feature reduction/selection makes clear pre-understanding about the data separability difficult.

The choice of kernels was motivated by the fact that linear kernel is expected to perform better for linearly separable data while RBF and Polynomial kernels perform better for non-linearly separable data points which require to be mapped to a different dimension within which they are separable.

The generalization error is estimated using repeated cross-validation (CV), resulting in 10 repetitions of a 10 fold CV with three sets of train, validation, and test of the ratios of 0.9, 0.05, 0.05, respectively. In each fold, they guarantee an equal number of samples from each class category following a conservative sampling strategy.

After applying three SVM kernels, the paper reports an accuracy of 49% altogether.

So, to come up with a better accuracy percentage on the above dataset, this paper implements the following algorithms.

Oblique Decision Tree Ensemble via Multisurface Proximal SVM

A paper[2] named Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine was published in the year 2015 by the renowned authors Le Zhang and Ponnuthurai Nagaratnam Suganthan was studied in detail and also was applied to our dataset.

The above paper discusses about a new approach to generate oblique decision tree ensemble wherein each decision hyperplane in the internal node of tree classifier is not always orthogonal to a feature axis. All training samples in each internal node are grouped into two hyper-classes according to their geometric properties based on a randomly selected feature subset.

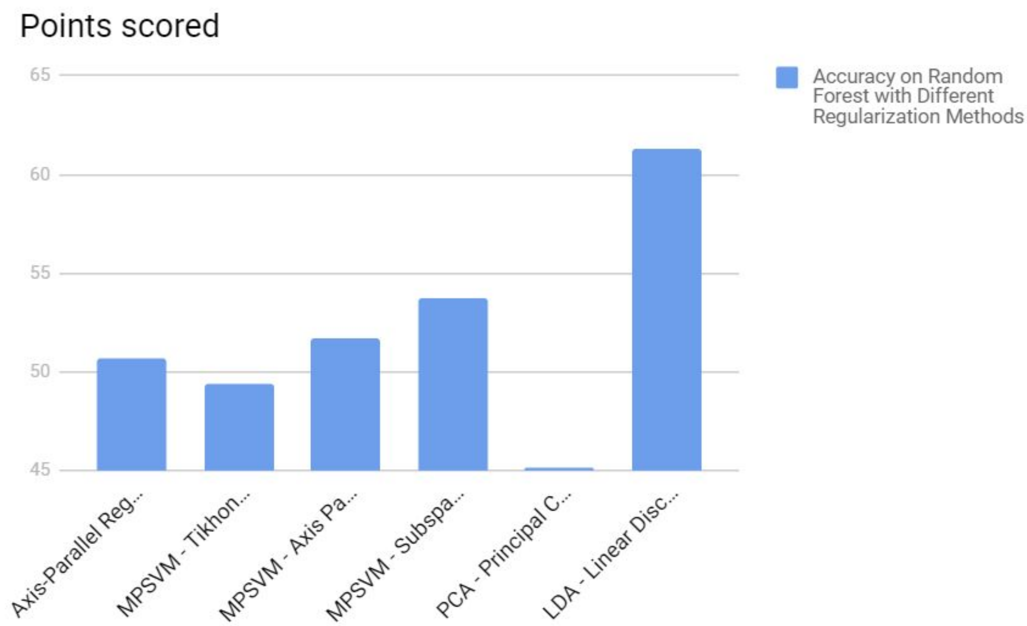
Then multisurface proximal support vector machine is employed to obtain two clustering hyperplanes where each hyperplane is generated such that it is closest to one group of the data and as far as possible from the other group. Then, one of the bisectors of these two hyperplanes is regarded as the test hyperplane for this internal node. Several regularization methods have been applied to handle the small sample size problem as the tree grows.

Different number of trees are grown and the corresponding accuracy is reported. After a particular thresholding point, the accuracy remains the same without any change irrespective of the change in number of trees.

The maximum retrieved accuracy is noted down and the bar graph is plotted for the accuracy that is finally reported. The accuracy retrieved after construct random forest with different regularization techniques is depicted in the below table.

Regularization Method	Accuracy of Random Forest
Axis-Parallel Regularization	50.668
MPSVM - Tikhonov regularization	49.43
MPSVM - Axis Parallel Regularization	51.678
MPSVM - Subspace Regularization	53.763
PCA - Principal Component Analysis	45.161
LDA - Linear Discriminant Analysis	61.29

The bar graph below is plotted for the observed accuracy of random forest versus different regularization methods.



Ensemble of Classifiers[8]
In the view of increasing the accuracy of our prediction, this paper trains the data with three famous classifiers and takes a hard-voting ensemble of all the classifiers. In this way, a final accuracy of **72.41%** is obtained..
The three classifiers used for training the data are as follows.

1) Linear SVM Kernel[5]

In general, SVM uses a hyperplane or a set of hyperplanes to discriminate two or more classes of the data from each other by maximizing the distance between the hyperplane and the closest points of each class while the classification error is minimized. It has a sound theoretical foundation and generates global solutions without getting stuck in a local minima.

2) K-Nearest Neighbours Classifier[6]

The KNN algorithm is a supervised learning algorithm that predicts the new test data based on the point/class that makes minimum euclidean distance with this data. In this way, KNN algorithm takes the K number of nearest neighbours and the class label that presides the maximum among he neighbours is also assigned to our new test data.

In this paper, KNN algorithm is applied with different values of K, that is, different number of neighbours are used for prediction. The best accuracy was observed when prediction is done with taking the class labels of the top 4 nearest neighbours (K=4).

3) Extra Trees Classifier[7]

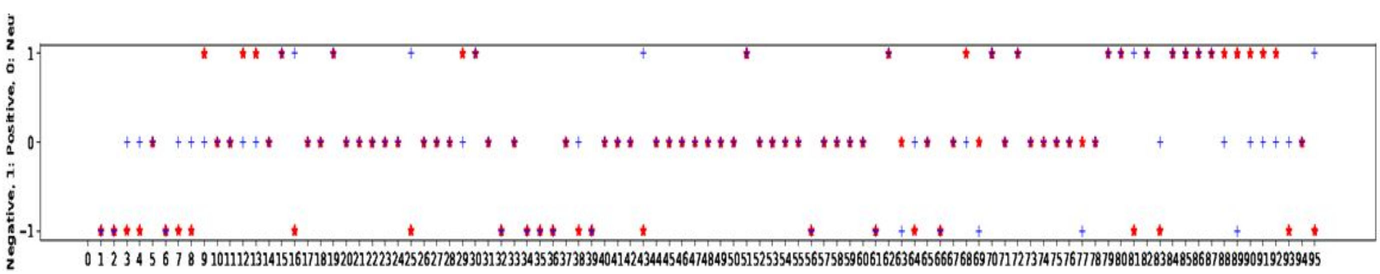
An “extra trees” classifier, otherwise known as an “Extremely randomized trees” classifier, is a variant of a random forest. Unlike a random forest, at each step the entire sample is used and decision boundaries are picked at random, rather than the best one. In real world cases, performance is comparable to an ordinary random forest, sometimes a bit better.

Result

As noticed above, the highest accuracy was observed for the ensemble method of the three classifiers - Linear SVM kernel, K Nearest Neighbours, Extra Tree Classifiers.
The highest accuracy observed on the dataset is **72.41%.**

In the following plot, both the actual class labels and the predicted class labels for all the 95 images are plotted. This helps us to visually perceive how accurately our algorithm has predicted the labels and whether it matches with the original label or not.

In the following plot, -1 represents the unpleasant class, 0 represents the neutral class, 1 represents the pleasant class. Also, red colour mark represents the actual class label whereas the blue mark represents the class predicted by our ensemble algorithm.



References

[1] Predicting the Valence of a Scene from Observers’ Eye Movements -- 2015
by Hamed R.-Tavakoli, Adham Atyabi, Antti Rantanen, Seppo J. Laukka, Samia NeftiMeziani, Janne Heikkilä

[2] Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine -- 2015
by Le Zhang, Ponnuthurai Nagaratnam

[3] A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II -- 2002
by Kalyanmoy Deb, Associate Member, IEEE, Amrit Pratap, Sameer Agarwal, and T. Meyarivan

[4] scikit-learn -- Machine Learning in Python

[5] sklearn.svm.SVC -- http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[6] sklearn.neighbors.KNeighborsClassifier --
http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[7] sklearn.ensemble.ExtraTreesClassifier --
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[8] sklearn.ensemble.VotingClassifier --
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html