# A Comparative Study of Physiological Feature Selection Methods for Emotion Recognition

Andreas De Lille

**Abstract**—An emerging topic of research is emotion recognition based on physiological signals and machine learning. Emotion recognition is the process of recognizing a person's emotional state. In this work the emotion recognition was done using a combination of physiological signals and machine learning. The flow of this approach is to record physiological signals from a person, extract features and feed them to a machine learning algorithm. This algorithm will then predict the user's emotional state. Even though a lot of research has been done, there is no agreement on what features are important. This work tries to overcome this problem by comparing a wide range of features with several feature selection methods.

**Index Terms**—Emotion recognition, physiological signals, machine learning, feature selection methods

## 1 Introduction

In order to provide accurate predictions, the developers of recommender system should first get to know their customers. To do this, information about the user's context has to be gathered. In this work, the definition of context given by Abowd et al. in [?] is used: *'Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.'* In this case, the entity is a person: the user to which the recommendations should be provided. The relevant information can be anything that is of use for recommendation systems. Thus, the solution to this problem should capture as much information about the user as possible. The current framework will detect and predict the user's activity level and locations. It will also label the locations of the user so developers of recommender systems know what kind of locations the user visits.

First, some details of the framework will be explained in Section **??**. Next, the results are presented in Section **??**. Finally, a short conclusion will end this extended abstract in Section **??**.

## 2 The context detection framework

The context detection framework consists of two modules. The first module is responsible for the collection of sensor data on the user's Android smartphone. It also captures the current activity (e.g. walking, cycling or being in a vehicle) using the Google Activity Recognition API. When this module finishes collecting a sample, it stores this information in a measurement database and alerts the second module that a new measurement is available. The second module will now process the sample. In short, the processing module will detect and predict the user's activity level and location. The processing module is explained in more detail in Section **??**. The findings of the processing module are then made available through the Context Façade to enable developers of recommender system to easily query information about the user's context.

### 2.1 Data Processing

As said earlier, the Data Processing module is mainly responsible for the detection and prediction of the user's activity level and location. The detection of the activity level of the user boils down to checking the sensors of the smartphone to see if the phone is moving, has rotated, is in a noisy or illuminated environment, ... If one of the rules are triggered, the user is considered as active. The detection of the user's location is based on the coordinates of the smartphone and the observed MAC addresses of nearby access points.

For the prediction of both the user's activity level and locations a Markov based approach has been taken. In order to capture the pattern of an entire week, the Markov chains have a transition matrix for every 15 minutes of the week, resulting in 672 transition matrices per chain. Given the detected activity levels and locations, the transition matrices are gradually trained to reflect the user's habits. Note that for the location predictions different sets of matrices are kept in parallel. This allows the framework to capture different types of periods, like regular working periods and holidays. Only the model that has made the best predictions at the end of week has its transition matrices updated, such that the matrices of the other models are not polluted with updates that belong to another pattern. A new set of matrices is created if no model succeeded to predict the locations sufficiently well. To make the predictions, one now has to multiply the current state with the transition matrices recursively. Two additional correction terms are added to the recursive computations in order counter some weaknesses of the models. The Data Processing module will predict the activity level and locations for the next 24 hours and updates these predictions every time a new sample has been taken. The most recent predictions are always available through the Context Façade.

Finally, the Data Processing module is also responsible for the labeling of the locations. It will automatically detect the home locations of the user, i.e. the locations where the user sleeps most often. Next to this it will also add information of Google Places and Foursquare to the locations. This makes the locations much more useful, since it is now known at what kind of locations the user is. Unfortunately, this labeling process with Google Places and Foursquare has to be manually controlled by the user. The reason for this is that not all locations exist in these services and that sometimes incorrect locations are proposed. Therefore the user has to approve whether or not the location proposals are correct.

## 3 RESULTS

In this section the most important findings of the conducted tests will be presented. During the development of the framework the models were tested on one smartphone only. When the framework reached its current shape, a user test with 11 test subjects was conducted to validate the functionalities.

The first results of the activity submodule were already rather satisfying. Using the automatic detection functions in the Context Façade, the errors on the detection of waking up and going to sleep over a period of 28 days were 6 minutes and 11 minutes respectively. The errors on the predictions were, logically, somewhat higher: 40 minutes and 33 minutes. This test was conducted when the framework was already running for over 20 weeks. In the user test the framework had only been running for about 5 to 6 weeks. The errors on the predictions during the user test were 69 minutes and 116 minutes. This worse performance can be attributed to the fact that the framework is not sufficiently trained in the beginning. This shows that the framework needs a period to warm up and that it gets better over the weeks.

Similar tests have been conducted for the location submodule. The error measure used to evaluate the location predictions is the logarithmic loss: $-\ln(p_{correct})$. $p_{correct}$ is the predicted probability of the location that afterwards turned out to be the correct one. Thus, the logarithmic loss will be low for correct predictions with high confidence and high for incorrect predictions. In the tests during the development the results were as expected. A second model was automatically created at the start of the academic year, because the current model at that time was trained during the holiday and could not predict the new pattern. It is also observed that the loss decreased over the weeks and was lower during regular periods (e.g. school) than during more irregular periods (e.g. holiday). The user test also showed a decreasing logarithmic loss over the weeks, which indicates that the model actually learns the user's habits. However, the creation of new models did not turn out as expected. It was expected that during the Easter Holiday some students would create a new model because their pattern would be different. However, no one reached the threshold to create a new model, except for one student that had a rather irregular week. It is clear that this calls for a new way of evaluating the models that should be implemented in the future, like for example ignoring the home locations of the users in the evaluation. During a holiday most people will still sleep at the same location, such that by ignoring the home locations, the real differences in the pattern are more focused on.

## 4 Conclusion

The results show that the framework is capable of detecting, learning and predicting the user's activity level and location patterns using a Markov based mechanism. There still are some weaknesses (e.g. the creation of new location models) that should be fixed in the future, but possible solutions for most of these weaknesses are already proposed. The framework currently has limited functionality (mainly activity and location detection and prediction), but it should be fairly easy to add new modules.

## References

[1] G.D. Abowd, A.K. Deyand P.J. Brown, N. Davies, M. Smith and P. Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC), 1999*, pages 304–307.