CS-772, Probabilistic Machine Learning
MID-TERM REPORT
GESTURE RECOGNITION SYSTEM

14th March, 2015

Prepared by-

Abhinav Jain – 13022
Rishab Jain – 13567
Peshal Agrawal – 13472

Gesture Recognition system focuses on the continuous mapping of a movement to a sound or control parameter.

Our work focuses on the discrete classification of a gesture. For the task at hand, we investigated how a model can be trained to recognise both static postures and gestures that consist of a cohesive sequence of movements that occur over a variable time period, which shall be referred to as temporal gestures. This report lays down the facts and details we have collected so far. The software we would be using is SARC Eyesweb and Gesture recognition toolkit.

For the recognition of static gestures, we will use **Adaptive Naive Bayes Classifier (ANBC**) for the advantages that ANBC has over other classification algorithms pertaining specifically to the problem of gesture recognition. Some of the advantages of using this algorithm are:

- It can be trained with relatively less number of training examples with the added advantage of tuning its parameters with incoming data: adapting itself.
- It can also recognize gestures in a continuous stream of data that may also contain non-gestural data without having to first train a null-model

NAÏVE BAYES CLASSIFIER:-

For the classification of static gestures, the multivariate Gaussian density is a suitable likelihood function to use, particularly in the instance where the feature vector **X** for a given gesture '$\mathbf{g}_{k'}$ is continuous valued.
It is beneficial to add an additional weighting coefficient (Θ_{kn}) for the nth dimension of the kth gesture. This simple addition of a weighting coefficient enables one general classifier to be trained say for left handed gestures, right handed gestures and two handed gestures, rather than creating and training three individual classifiers.

Gaussian model ($\boldsymbol{\Theta}$) for the kth gesture can therefore be represented by:

$$\boldsymbol{\Theta}_k = \{\mu_k, \sum_k, \Theta_k\}$$

After the Gaussian models have been trained for each of the G classes, an unknown N-dimensional vector x can be classified as one of the G classes using the maximum a posterior probability estimate (MAP). Prior is chosen such that it is constant across all the gestures and therefore would be ignored.

A rejection threshold, β_k, will also be calculated for each of the G gestures to enable the algorithm to classify any of the G gestures from a continuous stream of data that also contains non-gestural data. A suitable rejection threshold can therefore be computed during the algorithms training phase to enable the rejection of non-gestural data in the real-time classification phase. The rejection threshold, β_k, can be computed for each of the G gestures by taking the average confidence level of all the training data for class k minus γ (adjustable by user) standard deviations.

ADAPTIVE MODE:-

In this mode the algorithm will slowly refine μ_k, $\sum_k$, β_k for each of the G gestures, overwriting the previous models that have been computed earlier.
For the adaptive online training phase, the user must first decide on three parameters, the *maximum training buffer size*, the *update rate* and the *scalar on the number of standard deviations*. These parameters control the maximum number of training examples to save for each class in the model, how fast the algorithm retrains the model and the number of standard deviations to use when calculating the classification threshold in the model respectively.

If x is correctly classified as g_k and is greater than or equal to β_k then:

- Add x to the training buffer, removing the oldest training example if the buffer is full and increment the update counter by 1.
- If the update counter is equal to the update rate then recompute μ_k, $\sum_k$, β_k using the data in the training buffer. Reset the update counter to 0.

The exact value of these parameters will vary depending on the sensors and features being input to the algorithm and on the types of gestures the user wishes to classify, however, the certain general rules can be applied.

For the recognition of the temporal gestures, we would implement **SVM** and **Hidden Markov Model** on the same training data comparing the results. Both of these algorithms would be extended to adapt well to the problem of classifying gestures.

Multivariate temporal gestures are more difficult for a machine learning algorithm to recognise than static postures because of the inherent variability in human movement combined with the complex task of segmenting a multivariate temporal gesture from a continuous stream of data.
Sliding window as a segmentation method would be used, which uses a fixed size sliding window that is run continually over the data.

To reliably test a trained model's classification performance, we will calculate models' error rate using certain generalisation error functions.
This can be achieved by using a small subset of the training data to continuously train and test the model under different parameter values, with the parameter values resulting in the lowest generalisation error then being used to train the model using all the data.
In scenarios where only a limited amount of training data is available, a hold-out procedure (K-Fold Cross Validation), where some of the data is held-out of the training set and used for the test set will come into play.

Evaluation of the classification abilities of the aforementioned algorithms would be done using classification errors such as -

- ACVR (Average Cross Validation Ratio),
- ACCR (Average Correct Classification Ratio)
- APR (Average Precision Ratio)
- ARR (Average Recall Ratio)
- ANRR (Average Null Recall Ratio)

Performing the task at hand would also require pre-processing (feature extraction) and post-processing of the training data along with model training. Post-processing stage would consist of combining the results of different classifiers using ada-boost to train a super-classifier (not for real time classification).