Using Hidden Markov Models

for

Sequential Body Gesture Recognition

Sharon Lourduraj

Dec 26, 2011

Abstract

A brief introduction to TEX and LATEX

Contents

1. Introduction

2. Gesture Recognition

2.1 The process

The first steps in the process is to collect the data for the gesture that needs to be trained or classified from a human actor using a motion capture system. For clarity we establish needed terms for this reprot. We call the data captured from the motion capture system a `gesture`, which is a specific instance of an `emotion`. One or more gestures might be similar to each other and they both fall under the same emotion. For example, being `angry` is an emotion, but one can show anger through various gestures. The data from the motion capture system is essentially a sequence of frames with motion capture data (angles, or position of body points in space). A `posture` is a specific frame in a gesture. Essentially, a gesture is made up of a sequence of postures.

Once the motion capture gesture data is collected, feature vectors need to be extracted from the gesture. Feature vectors are extracted from a posture or across multiple postures. The feature, location of the actor's head can be extracted from the posture, while the velocity of the actor's head needs to be extracted across two postures over time. A posture can have multiple features, combined they are called feature vectors. So for a gesture containing 500 frames (postures), 500 feature vectors can be derived. The dimension of a feature vector is dependent on the choice of the feature. Location of the actor's wrist has three dimensions in space, while the actor's elbom angle has only one dimension (the angle itself). This process is repeated for all the gestures captured.

Templates need to be extracted from these feature vectors. Templates are just a mechanism to group similar postures together. If someone is raising their hand, we can take three snapshots while the person is raising their hand. When their hand is at the initial position, when the hand moves half way up, and then the final position. Any intermediate snapshot is not necessary since, the three snapshots convery that the person is raising their hand. Thus, given a set of feature vectors we identify few templates from it. Note that templates from one gesture can be used in another gesture, hence template dataset is essentially a database of unique postures using which other gestures can be created. This introduce the idea of `codebook`, a synonym for a template database. We label each posture in the codebook with a unique number from $1 \dots n$, and one can refer to them with a `page number`.

Consider a gesture and a codebook, we need to transform each posture in the gesture into a page number in the codebook. Thus rather than having a $n \times m$ matrix of postures and features, we have a $1 \times m$ vector where each column corresponds to a page number in the codebook. Note that there is a loss of information in this transformation. The original gesture file contains accurate information about the gesture, while the codebook transformed version contains an approximate, hopefully sufficient, version of the gesture.

The concept of codebooks is necessary to work with HMMs, as HMMs are built around the concept of states and observations. A gesture can have many states depending on the complexity of the gesture. We consider the gestures to have three states, a starting state, a middle state and the end state. Between these states there are many hidden states that we are unaware of, the goal of HMM is to identify these states and provide a likelyhood estimation. A state can emit one or more observations. An observation is essentially an occurance of a posture. If we consider someone giving a wave, there are three distinct states we notice. The start state where the hands are relaxed, the middle state where the hands are bent in front of the person, and the final state where the person has their palm open and tilted about the wrist. Likewise, three distinct postures can be extracted from the wave gesture, and between these postures there are hidden states that an observer is unaware of.

A HMM is trained by providing a sequence of observations, pagenumbers from the codebook. The training calculates the probability of a specific posture occuring given that previously a particular posture occured, in the given state. Similarly, for classification an observation sequence is provided to each train HMM, through which the likelyhood of an observation sequence can be calculated. The gesture is classified to fall under a particular HMMs emotion label if its likelyhood is higher than the HMMs of the other emotion labels.

The experiments conducted can be categorized into two parts, experiments dependent on HMM topology variation, and experiments dependent on feature vector variations. Based on emperical evidence through the learning and discovery, that was invovled in implementing and running these experiments, it can be said that the HMM topology and feature vectors selected play a key role in the performance and accuracy of using any HMM. This chapter describes the varitions that were used in the experiments. The results of these experiments are discussed in chapter 3.

2.2 HMM Topology

A HMM Topology describes how the various states could be related to one another. Mathematically, this is described by the probability distribution from one state to another. Keeping in mind the problem of recognizing body gestures while performing an activity, if we consider the gesture of being 'angry,' there are various transitions that a body goes through to exhibit an 'angry' gesture. Considering the simplest case of the 'angry' gesture from a T-posture. First the arms bend in toward the hips, the legs spread apart and finally the hands are rested on the hips, these sequence of states indicate that the person is exhibiting an angry posture. This is a linear progression of states. If the hand's rested on the hips before the legs are spread apart then the sequence of events most likely doesn't indicate the angry gesture. If one were to train a generic HMM to classify the angry gesture the probability distribution would force the HMM into a linear topology. The left-right linear topology is perfect for recognizing a gesture that is progressive in nature. There are various kinds of angry gestures. Consider an alternative where the hands bend in toward the hips, the legs spread apart, the left hand rests on the hips, while the right hand bent at the elbow points at someone or something. This gesture also represents a progressive left-right topology. Here we have two different gestures, but both indicate a form of angry gesture. With this distinction in mind we propose two different architectures for experimentation.

The first proposed architecture is to create a HMM for each distinct gesture (example: angry-1, angry-2,. . . , fear-1, fear-2,. . . ,fear-n,. . .), even though they might represent the same gesture category (example: angry, fear). In this architecture the system is able to distinguish between various forms of gestures, or the intensity of gestures, in the same gesture category. If someone is really angry they might point at someone and oscillate their hands toward them in short bursts. If someone is angry and disappointed, they might shrug the shoulders and throw their hands up in the air swiftly. The second proposed architecture is to create a HMM for each distinct gesture category only (angry, fear,. . .). In this architecture many similar gestures are grouped together into a single category.

2.3 Feature Vector

Elbow angle, position of wrist, knee angle, angle at the hips, head tilt, are some of the data that one needs to assess to distinguish one gesture from another. Together, they form a feature vector. Having the right feature vector is essential in classifying and recognizing gestures. If one has a feature vector consisting of upper body joint positions alone, the accuracy of recognizing gestures that requires having information about the lower body point positions will be low. Likewise, the information in the feature vector is also important, a feature vector consisting of joint points and a feature vector consisting of joint angles might provide different recognition results.

Dimensioned Data

Dimensioned data provide a means to an end in understanding how the HMM framework works in classifying and recognizing sequence of patterns. In the experiments conducted the dimensioned data consisted of having 3D positions of various points of the human body. When these points are plotted for a single frame it is easy to visualize how the HMM framework works to recognize and classify gestures. Plotting a sequence of frames relay information about the gesture being performed. However, there are greater disadvantages to using dimensioned data in a classification and recognzition algorithm. This data might convery information visually to a human observer, but an algorithm sees no advantage in having dimensioned data. Infact, dimensioned data restricts the algorithm from performing equally well when the input data conveys the same gesture but is performed by a different actor. Different actors exhibit different dimensioned data for the same gesture, the actors height, length of arms, torso length, length of legs are all different. Though a human observer is able to filter out these data whilst trying to recognize the gesture being performed, an AI algorithm is unable to filter or transform this data in such a way that its recognition and classification is unaffected. Thus, the need for dimensionless data.

Dimensionless Data

Dimensionless data play and important role in machine learning, specifically in gesture recognition it allows data to be generalized for any actor regardless of their physical features. When we compare angry, fear, happy or sad gestures from one actor to another, they will be quite similar when dimesionless data is used. Dimensionless data represent temporal data of a geasture, such as velocity and acceleration of points on a human body, or of joints; essentially data that represents a state in time. Such data is difficult for a human observer to visulize and categorize efectively. Consider looking at the performance of a angry gesture. By observing the velocity of points on a human body one cannot classify the gesture being performed without the help of spatial data. However, this provides no challenge for a machine learning framework, since they will be trained to recognize gestures from this prespective. Below are the various dimensionless data sets used in the experiments.

Point velocities are essentially velocity of data points on a human body at an instance in time. We realize that gestures are dependent on speed at which it is being performed. If someone ducks within a second, it can be said that they reacted to a dangerous situation. If they ducked slowly, they are most likely reacting to a neutral situation. Velocities convery such information to the machine learning framework in a dimensionless fashion. Angles are dimensionless, as they are relative to two bodies. If we take the eblow angle, that is the angle between the upper arm and forearm, regardless of the actor it will be similar for a specific gesture. Alternatively, a combination of joint angles and relative positions of certain features on the human body can also be employed.

2.4 Datasets

Two datasets

3. Experiment Analysis and Results

3.1 Initial experiments

The first step in the experiment was to understand how the HMM framework worked, a dimensioned feature vector was used for this purpose. A left-right linear HMM was used for each gesture, rather than for each gesture category. The codebook vectors were obtained manually. Table 3.1 summarizes the results obtained from the various test runs.

The first type of test consisted of 70 gestures, 35 of which were used for training and 35 for training. We realized that the training sample is quite low, so a second round of tests were conducted which consisted of 90 gestures, 35 were used for testing and 55 for training. Again a third round with even more training data was conducted, with 120 gestures in total, 85 for training and 35 for testing. From the table we can see that the total classified gestures improves as we increase the number of training samples. The number of correctly classified gestures also improves, but at a very low rate. To understand the low classification rate it was necessary to take a closer look at the framework.

	Correctly Classified (%)	Total Classified	Testing data	Training data	Total Gestures
Manual #1	8 (22.8)	10	35	35	70
Manual #2	10 (28.5)	24	35	55	90
Manual #3	11 (31.1)	31	35	85	120
K-means	3 (8.5)	10	35	55	90

Table 3.1: Recognition

Whilst training a HMM for a specific gesture, the framework produces the sequence of observations listed in table 3.2, for one of the fear gestures. Each number corresponds to a page in the codebook, where the contents of the page is a single frame of data from the motion capture system. Thus, the frame in page 1 appears after the frame in page 23, similarly the frame in page 10 appears after the frame in page 1, so on. The HMM training algorithm trains a probabilitic model for such a pattern of observation, such that if the sequence in the pattern is altered the HMM would return a low probability for that sequence to appear.

Now, we look at the the observation pattern for one of the fear gestures in the testing set that was missclassified as one of the angry gestures. When the pattern is compared against the fear gesture in the

Fear in Training	28	28	28	28	28	28	28	28	14	14	14	30	30	30	30
Fear in Testing	21	21	21	21	21	21	10	10	10	10	10	10	20	32	33
Angry in Training	28	28	28	21	21	21	21	21	21	21	28	28	28	28	28

Table 3.2: Sequence of observations for an incorrectly classified fear gesture.

training set, it is very different. However, when we compare the pattern to an angry gesture in the training set, half the pattern matches a sequence of observations in the angry pattern. Though only the repitition of the number 21 in the angry gesture matches the fear gesture, the HMM classifies it as angry with a very high probability.

Angry in Training	24	24	24	24	24	24	21	21	22	22	22	22	21	21	24
Angry in Testing	24	24	24	24	24	21	22	22	22	22	21	23	23	24	24

Table 3.3: Sequence of observations for a correctly classified angry gesture.

Table 3.3 displays the observation pattern for a correctly classified angry gesture. Note the similarity of the patterns between the training angry gesture and the testing angry gesture. The HMM classified this as an angry gesture with a very high probability. From these tables we conclude that both, the observation that follows the current observation, and the repitition of the same observation in a sequence plays a key role in determining the outcome of a HMM.

If one were to extract template frames from a training dataset of 90 gestures, approximately 1-2 template frames per gesture, we would have approximately 225 templates. This also means that our codebook has 135 pages. A large codebook means a larger training dataset is necessary to estimate a large observation probability matrix for our HMM. We quickly realize that many of the gestures share similar postures, a manual selection of templates ensured that the posture selection for codebook pages were unique in nature. Through observations and visual analysis we realized that in our experiments the problem lies in the creation of this codebook. Manual creation of the template frames though accurate was infeasible when the training data set became larger and larger. It was not as straight forward as selecting a few universal template for angry gestures, and use these templates to convert the training dataset into a sequence of observation patterns as in tables 3.2 and 3.3. It was necessary to go through each training dataset to identify unique gestures, and extract frames from them to properly classify similar gestures. The training data itself contained new patterns that needed to be extracted.

We introduced a clustering algorithm to replace the manual codebook creation process. The `k`-means clustering algorithm, creates `n` clusters and groups similar data together in the `i`th cluster of the `n` clusters. The algorithm clusters similar data by calculating the distances of the centeroid of the given data, thus data that is closer to each other are grouped in the same cluster. Now, the codebook creation was simplified for a large dataset. Table 3.1 lists the results of the test runs conducted after using the clustering algorithm for codebook creation. The results are not impressive at all, visual analysis of the codebook vectors and the observation patterns created for each posture tells us why.

Having a codebook that is as large as 225 pages, even for small variations between two similar gestures the observation sequences are widely different, as in the fear gesture in table 3.2. Consider the images in figures 3.1, they represnt figures from codebook pages 2, 11 and 18. Visually they are very similar to each other, however the k-means algorithm clusters them into different pages because of small variations that categorize these images as outliers, such as hip angle, position of wrist, head tilt, and spacing between the feet. These are very sublte from the figures, however the k-means algorithm picks up on these.

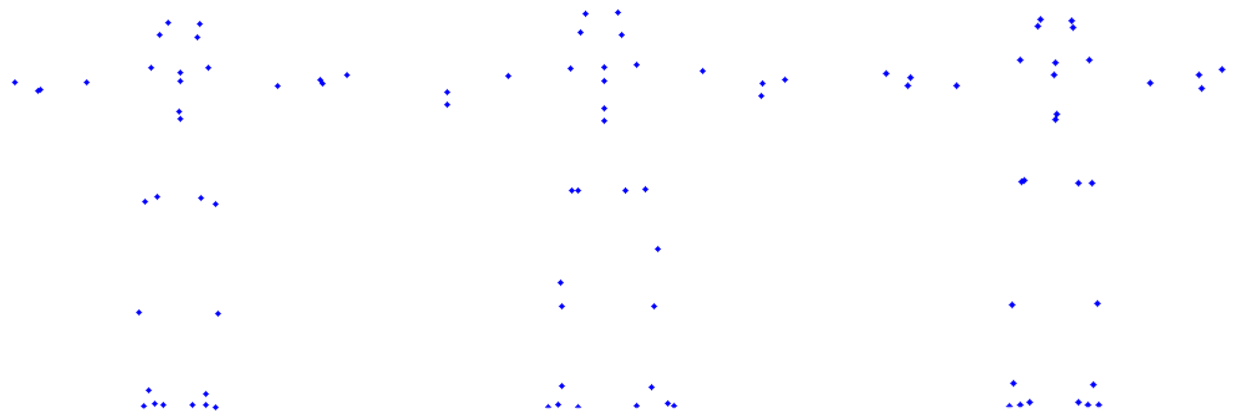
(a) Codebook Page 2 (b) Codebook Page 11 (c) Codebook Page 18

Figure 3.1: Similar codebook pages

With this in mind, we plotted the postures in the pages listed for the fear gesture, from table 3.2; pages 21 and 28 were similar, and pages 14 and 10 were similar. The misclassification in the experiment is caused by such occurances, of similar postures being classified into different codebook pages. Similar postures are being categorized under different pages because of small variations in position of points on the human body in 3D-space. We suspect that the use of dimensioned feature vectors is causing this, so we proceed by using dimensionless feature vectors.

We also make one final observation about the current architecture. Every gesture has a HMM associated with it even though it might be similar to an existing gesture that has already been trained. This does not pose a problem if every single gesture in the data set were unique. In our data set, the gestures that belong to the `angry` category are not all unique, many of the gestures are similar with small varitions. In a problem that invovles humans, whose features and expressions are varied, it is necessary to group up these variations in gestures under a single gesture. Thus, we consider training all unique and non-unique gestures that belong to the `angry` category under one HMM. If this HMM results in a high probability of a gesture occuring during testing, then it will be classified as `angry`.

3.2 Improving performance

The initial experiments conducted above has given us important insights into how gesture recognition heavily depends on organizing the codebook pages. The second part of the experiment modified the HMM architecture such that smiliar gestures were categorized and trained under a single HMM. Thus for a dataset that contains four emotions (angry, fear, sad, and happy) four HMMs were created.

	Angry	Fear	Happy	Sad
Angry	13	2	0	1
Fear	11	7	6	3
Happy	2	1	11	6
Sad	3	1	1	19

Table 3.4: Confusion matrix for 3-state HMMs with velocity feature vectors.

Table 3.4, displays the confusion matrix for one of the test runs that classified 98% of the testing data, and 56% of the testing data were classified correctly with high probability. The test run had a dataset of 192 gestures, 93 were used for training the HMMs, and 89 were used for testing. This test run used velocity feature vectors of the training and testing data, for codebook creation and template matching respectively. Note the classification rate is very high this is caused by the change in the HMM architecture from using one HMM for every gesture to one HMM for every emotion (were similar gestures are grouped together). This change resulted in having a sufficient amount of training data for each HMM, approximately 23 gestures were used to train each of the angry, fear, sad, and happy gestures. The confusion matrix allows us to identify areas where improvements can be made, we note that most of the `fear` gesture is being misclassified as `angry`. The only property to control in a HMM model is the number of states associated with it. The results in table 3.4 are for HMM models with three states (for each emotion). Table 3.5 lists the confusion matrix for a table where the `angry`, `happy` and `sad` HMMs have three states and the `fear` HMM has four states. We note that the classification of the `fear` emotion has improved slightly, but the classification of the

	Angry	Fear	Happy	Sad
Angry	5	4	3	4
Fear	5	9	9	4
Happy	1	4	9	6
Sad	1	2	1	20

Table 3.5: Confusion matrix where fear HMM has 4-states.

`angry` emotion has dropped. Comparing table 3.4 and 3.5, the classification of `happy` emotion has dropped and now the majority of the `fear` motions are also being classified as `happy`. We proceed by adding more training data to see if this has any effect. A caveat that needs to be noted here is that, if there are gestures that are unique to a particular emotion that were not being classified properly but now this gesture is in the training data, this gesture will not be tested for in the testing data. Thus, the results need to be interpreted with caution.

	Angry	Fear	Happy	Sad
Angry	4	2	0	0
Fear	0	5	2	1
Happy	1	0	4	2
Sad	1	2	0	6

Table 3.6: Confusion matrix for 3-state HMMs with additional training data.

Table 3.6 lists the confusion matrix for which there were 152 gestures for the training data, approximately 38 gestures for each emotion, and 30 gestures for the training data. With all the data being classified and a 63% correct classification rate. With the caveat, that some unique gestures might not be in the testing data, in mind, we can say in general that higher training data improves the recognition rate of the HMMs. One scenario to think about is how do the classification rates change when only two of the emotions are trained and tested for.

	Angry	Happy	States	Recognition rate
Angry	11	11	3	71%
Happy	2	24	4	
Angry	15	7	3	78%
Happy	3	23	3	
	Angry	Sad		
Angry	14	6	3	81%
Sad	1	20	3	
Angry	18	2	3	83%
Sad	4	17	2	
	Angry	Fear		
Angry	17	4	3	66%
Fear	10	12	3	
	Fear	Happy		
Fear	14	4	3	72%
Happy	8	20	3	
Fear	11	7	2	74%
Happy	4	24	4	
	Fear	Sad		
Fear	15	9	4	77%
Sad	0	21	3	
Fear	19	5	4	85%
Sad	0	21	4	
	Sad	Happy		
Sad	14	4	4	61%
Happy	2	13	4	

Table 3.7: Confusion matrix for comparison agianst each emotion.

Table 3.7 displays a confusion matrix, recognition rate and the number of states used for each emotion's HMM. We note that when comparing two emotions to each other an average of 15-20% missclassification rate exists. Table 3.7 also displays improvements on recognition rate when comparing some emotions, for example `fear` vs. `sad`. Training the HMMs with 4 states fear and 3 states for sad, a recognition rate of

77% is achieved, but result is improved upon by training both HMMs with 4 states. Optimization of HMMs for states is an important step in improving the recognition rate. This property not only affects how well the respective gestures are classified (a sad gesture classfied as a sad emotion), but also how other gestures are classified against it (a fear gesture classified as a sad emotion). Improving a classification rate for one gesture might affect the classification rate of another gesture. Thus when one or more emotions are invovled, as in tables 3.5 or 3.6, a balance of states between each individual HMM needs to be achieved.

3.3 Unacted dataset

The experiments that have been discussed until now were performed on an acted dataset, where the actor acting the gesture was asked to specifically demonstrate an emotion. A test was also run on an unacted data set, where the actor was asked to perform a task, during this task various emotions were extracted and classified. The idea was to capture the natural essensce of an emotion. To make sense of the emotions they were classified into three emotions, `positive`, `negative`, and `neutral`. A `positive` emotion can consists of happy, excited, or surprised, while a `negative` empotion consists of angry, fear, or sad. All other gestures were considered `neutral`. Even for a human observer it was difficult to label the dataset with one of the emotions, thus training a HMM to detect and recognize such gestures would be a test in robustness.

	Positive	Negative	Neutral
Positive	6	1	1
Negative	2	3	1
Neutral	2	2	11

Table 3.8: Confusion matrix for the unacted dataset.

From the tests performed on the acted data set, we see that a dimensionless velocity feature vector with one HMM for each emotion gave us better performance. We apply the same preference towards training and testing the unacted data set. Table 3.8 displays the confusion matrix for the unacted data set. A total of 64 gestures were extracted and labeled manually from the dataset, of which 33 was used for training the HMMs, and 31 for testing. It is important of to note the distribution of the emotions in the dataset. Out of the labeled gestures, 19 were labeled as `positive`, 17 as `negative` and 28 as `neutral` gestures. The framework recognized 61% of the gestures correctly. The `positive` and `negative` HMMs used 2 states while 3 states were used for the `neutral` HMM. A modified test was run to see how additional training data affects the

	Positive	Negative	Neutral
Positive	3	1	0
Negative	0	2	3
Neutral	0	1	4

Table 3.9: Confusion matrix for the unacted dataset.

classificaiton results; 49 gestures were used for training, and 15 were used for testing. Table 3.9 displays the results of the test. We note that in both cases (tables 3.8 and 3.9) the `negative` emotion is performing poorly. The tests did not use a sufficient amount of gestures for training to conclude the reasons behind the missclassification concretely. 60% of the testing data was classified correctly. Thus we see no improvement in adding more training data in this case, but an observation to note was that only 2 additional `negative` gestures were added to the training data in the modified test, as more `negative` gestures were unavailable.

The purpose was these tests were to determine the robustness of the HMM framework for recognizing gestures under different circumstances (acted, or unacted). From the results above we can safely conclude that the HMMs are sufficiently robust to adapt to situations as long as sufficient training is provided.

4. Conclusion

Bibliography