Tactile gesture recognition using Hidden Markov Models

Pierre-Luc Bacon

December 16, 2011

Abstract

Hidden Markov Models (HMM) have been used extensively for speech recognition [5, 8], online handwriting word recognition [2, 4], and gesture recognition of the American Sign Language (ASL) from video [9]. This paper draws ideas from those studies and proposes an HMM-based algorithm for detecting tactile patterns on a *skin sensor* manufactured by Meka Robotics LLC. Due to the novelty of this technology, many unknowns and constraints had to be overcome. The HMM model provided the desirable property of being insensitive to temporal variations in gesture executions while maintaining a simple feature vector definition. Extensions to the presented approach, based on Principal Component Analysis (PCA), are proposed for dealing with scalability concerns.

1 Introduction

Handwriting recognition was studied thoroughly over the last two decades and Hidden Markov Model turned out to be the dominant approach in most studies. Using some similar ideas, techniques for recognizing hand gestures of the American Sign Language (ASL) in video sequences were also developed. Altogether, these different applications share the common property of dealing with spatial variations of an object through *time*, representing them as *time series.*

Similar in nature to handwriting recognition, gesture recognition on tactile screens for mobile devices also have become prominent over the last few years. The present paper distinguishes itself from this class of applications by focusing on a special type of pressure sensor mainly geared towards robotics applications : a *skin sensor* developed by Meka Robotics LLC. This type of sensor is meant to cover large surfaces at low cost while providing access to pressure readings (16 bits integers at the moment) at different discrete locations. A rationale for performing gesture recognition on such a sensor can be provided in terms of its potential outreaches in human robot interaction (HRI).

As opposed to the other classes of applications outlined previously, a tradeoff on cost versus resolution makes the traditional approaches difficult to apply directly for this problem. For this reason, the general HMM framework brought in these studies must be adapted to this resolution constraint, but yet be flexible enough to scale to high dimensional setups. Indeed, the modular characteristic of Meka's sensor makes it subject to configurations that could easily grow beyond a thousand simultaneous signals being delivered to a host computer. Clearly the *curse of dimensionality* is an eminent concern for any approach aimed at the problem studied here.

With the canonical HRI application for this sensor, ensuring both spatial and temporal invariance across different executions of a same gesture would also be of first importance. The HMM model presented in this paper lends itself naturally to the temporal robustness requirement. A training problem was encountered during this experiment and solutions are proposed to avoid falling in such a pitfall. Potential approaches for better handling dimensionality *explosion* and issues due to spatial scaling of gestures are finally presented.

2 Related work

2.1 Handwriting recognition

In both [4] and later in [2], left-to-right HMMs are used to as letter models. The set of specialized letter-hmm's is then used to produce full word models by taking the concatenation of their letters.

This approach is preferable over word models as it does not suffer from vocabulary increase : word models can be created only by specifying their letter sub models. Furthermore, because different words can share many letters in common, letter models avoid potential redundancy of otherwise full word hmms as encountered in earlier studies [7]. To that account, [6] appears as a turning point towards letter models. As opposed to the studies that followed, [6] specializes some single-character HMM depending on letters found before and after it under different writing contexts.

As most models proposed for speech recognition, HMM's used for handwriting recognition use left-to-right topology with no state skipping [4, 2], contrasting with the earlier studies [7, 6]. Optimization of the number of states involved in left-to-right HMM's in [2] showed that a subtle improvement in the recognition rate was obtained by increasing their number from 8 to 12. Although it had a slightly different architecture, [6] used a seven states letter model.

Feature vectors can be extracted in different ways from the input signal. A classification of the different segmentation methods is made in [4] and two main classes are distinguished: *point oriented* and *segment oriented*. The former refers to feature extraction methods that produce observation sequences in an incremental fashion by considering local features at a given *step*. A window of fixed or variable size can be defined for a given step and only local features that are contained under it are added to the feature vector. The *segment oriented* approach differs from the latter by considering different potential feature vectors at different scales : a *global* approach.

An additional pre-processing step is always involved before feature extraction. In most cases, it consists of *slant correction* of the input words and rescaling [2, 4]. Cubic spline interpolation is used in [4] to form a continuous signal, whereas [6] only ensured that the sampling rate was high enough.

The different quantities expressed in feature vectors usually consists of the local tangent slope angles [6, 4, 2] but can also contain other geometrical features such as the number of pixels, centre of mass or contour gradient [2].

2.2 Video gesture recognition

HMM-based approaches for handwriting recognition have also been applied to the closely related problem of gesture recognition from video feeds. In some studies, only hand gestures are considered for the recognition of words in the American Sign Language [9, 3], while others attempt at extracting and recognizing *gesture-drawn* letters [10, 1].

The input representation differs considerably for these two classes of applications. However, angle, inertia and centre of mass (position) are common quantities contained in a typical feature vector [9, 1] from the class of *geometric features* representations.

A simple feature vector definition for video consists for *templates* made from a subset or complete raw data. The case of [1] is an example of a template approach that consists of a series of points sampled along the path during gesture execution. However, the PCA and Independent Component Analysis (ICA) techniques are applied to on this series of points before applying them to an HMM.

[3] uses a similar idea but does not apply the template directly to the HMM. Indeed, after extraction and normalization of the hand area to 32x32 pixels , PCA is applied on the template to reduce its dimensionality. Then a multi-scale Gaussian search is performed to match the extracted image area to a pre-defined template, and a decision tree assists the search operation. An HMM is then used only to recognize the temporal structure of a gesture using a sequence of discretized template indices forming a gesture.

[10] takes a different approach by computing the short-time Fourier transform (STFT) of the motion paths for every gesture and take the amplitude coefficients of the spectrum to form feature vectors. This vector is then transformed to a discrete representation by applying the LGB algorithm to perform vector quantization.

While [3] uses template matching to segment areas of interest, [9] uses colored gloves for tracking hand position and [1] utilizes a stereoscopic vision system to locate and track special markers in 3D space. Although similar for their use of HMM, video gesture recognition clearly differs from handwriting recognition in terms of segmentation techniques.

3 Proposed approach

For sake of simplicity, a template feature vector definition was used as a first attempt to use HMM for the problem of tactile gesture recognition. The following sections will detail the pre-processing steps applied on the input signal as well as the software decision considerations, HMM design and training procedures. Potential approaches for better handling dimensionality issues will be considered and an alternate feature vector definition will be proposed.

3.1 Pre-processing

The pressure patterns created by the different gestures were recorded on a 4 by 3 *taxels* (tactile sensing element) array produced by Meka Robotics LLC (figure 1).

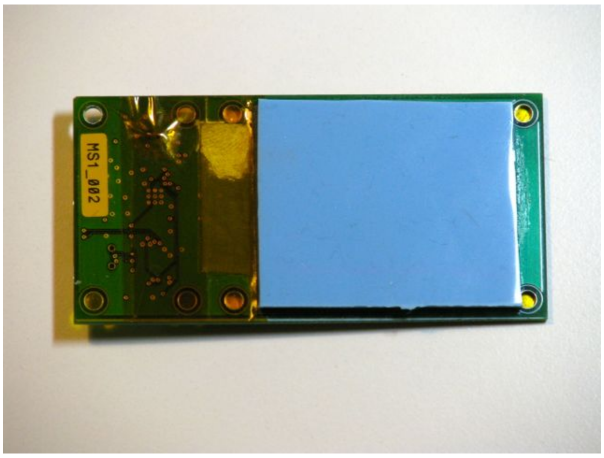


Figure 1: Skin sensor

Twelve 12 signals were recorded simultaneously from the device over USB at about 30Hz. Each taxel measures the normal component of the force applied to it over 16 bits. A silicone membrane cast on top of those taxels also spread the force across a larger surface and avoids creating discontinuities between any two taxels. This mechanical construct however puts a higher static load on the taxels such that the bias averages 32000 ticks across the array for the device used in this experiment. Data collected with no load on the sensor shows that the noise varies from 2 to 8 ticks. Unless very subtle pressure patterns would have be recognized, the corse sensor characterization performed in this study does not indicate that noise filtering would be necessary to achieve the intended goal.

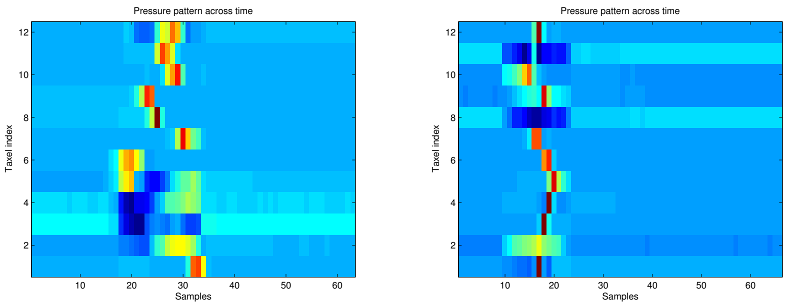
To avoid biasing the classifier towards a given taxel during training, normalization should be performed. An easy approach for this would be Min-Max normalization in the 0 to 1 range. Given a raw sample x from a taxel, it would be normalized by:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

x_{max} was chosen empirically to be 55000. Indeed, the theoretical saturation value of 65535 would not be attained realistically during normal interaction with the device. The measurements collected with no load were used for defining x_{min} as the average bias measured for every given taxel.

Although this normalization technique seemed reasonable, numerical issues were encountered in the implementation using two different HMM software packages : the Java library *Jahmm* as well as the Matlab probabilistic toolbox *pmtk3*. In both cases, the inverse of the covariance matrix used for EM during training would fail to be evaluated. This happened because the covariance matrix was not positive definite as expected by the Cholesky factorization algorithm used to find the inverse.

Because of this inability to execute the training phase, another normalization scheme was used. Instead of clamping the values to the 0-1 range by Min-Max normalization, z-scores were used instead. The taxel measurement x would then



(a) Cicular

(b) Diagonal

Figure 2: Training instances normalized by z-scores. Two types of gestures are shown : a single finger circle-like gesture and a diagonal swipe across the array

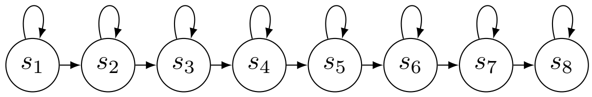


Figure 3: Left-to-right HMM used to model individual tactitle gestures

become:

$$x^* = \frac{x - \mu}{\sigma} \tag{2}$$

In the above expression, μ is the average value of a taxel taken over all the collected samples from a given instance of the training set. Similarly, σ is the empirical standard deviation of the recorded signal.

3.2 Hidden Markov Modelling

Based on the surveyed literature presented above, a left-to-right HMM with no skip between states was decided. It also appeared reasonable to use eight states for representing any given gesture (see figure 3). This choice was not only motived by the related studies, but also by the scale of the device : a relatively long gesture would not travel across a much longer distance than 8 taxels. This intuition is obviously very limited and finding the optimal number of states might be the subject of another study.

The input vectors considered for this model consisted of the z-score normalized values of the 12 taxels sampled at given instant. This representation is thus an instance of a template feature vector taken locally from the raw input feed under a window of 1 sample in width.

To avoid the inherent complexity that comes with vector quantization, the observation distribution was chosen to be modelled as a multivariate Gaussian. That way, the z-score vector can be used without further modification. A mixture of Gaussian representation was also attempted and yielded comparable results.

The initial state distribution π was set to such that the probability of starting in state 1 was maximal : a reasonable assumption considering that gestures are always started in the same manner. The state transitions probabilities were also initialized accordingly in matrix A (eq. 3)

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, A = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

3.3 Training

One hundred samples for two different gestures were collected (see figure 1) for a total of 200 training instances. Only 50 examples per class were used for training and the remaining examples were kept for testing.

Using the pmtk3 toolbox for Matlab, the Expectation Maximization (EM) algorithm was used to train two HMM's : one for a circular-like gesture and the other for a diagonal gesture across the array. A fixed limit of 10 training step was set for this phase.

3.4 Classification

In the training phase, the likelihood of generating the training dataset given its associated model was maximized. For the classification task, the different optimized models are all being evaluated for a given testing instance and the one with the most probable state sequence given the observations is retained. Viterbi algorithm was used to compute the log likelihood of the state sequences given the observations and the maximum value across all models was selected for labelling.

4 Results and analysis

Initial testing results indicated a perfect recognition score over a 100 different testing instances from the two classes. Suspecting that over fitting might had occurred, it was attempted to measure the effect of using a smaller training set. Nonetheless surprising results were obtained. A minimum of five examples for each class appeared to be necessary in order to get the EM algorithm to succeed numerically. From 5 to 50 examples per class however, a 100% recognition rate was still obtained.

Considering that the number of states might had been overestimated, the number of states was decreased from 8 down to only 2, and yet a perfect score was obtained. It was expected that all of the 8 states would be used to represent the gesture. To better analyze the situation, the generated Veterbi paths were examined. For the HMM associated with the first model, the most likely state sequence was $2, 2, ..., 2$ and similarly $8, 8, ..., 8$ for the second model. The same kind of observation could be noticed in the optimized π (see eq. 4).

$$\pi = \begin{bmatrix} 0.086 \\ 0.224 \\ 0.017 \\ 0.017 \\ 0.034 \\ 0.017 \\ 0.017 \\ 0.586 \end{bmatrix} \tag{4}$$

Clearly, each model seemed to have internalized only one characteristic of the training data. By visual inspection from the collected data, it appeared that taxels 2 and 8 were both not involved in one gesture or the other. Hence, the algorithm used this only distinguishing feature from the training set to base its decision on.

Due to the high similarly of the training examples, it can be then understood why 2-states model was sufficient for recognizing all testing examples perfectly.

5 Future work

Although an HMM model was shown to be sufficient for recognizing two different gesture types with varying temporal structure, the approach would most likely not be applicable to more general setups. Some potential improvements are proposed below.

5.1 Feature vector redefinition

Not only the feature vector definition used in this experiment does not scale to more taxels, it is also prone to systematic feature specialization during training. For this reason, it might be worth experimenting with a input representation closer to studies in video gesture recognition such as as in [9] or [1].

At each sample period, the index of the taxel with the highest normalized pressure value could be used as one component of the feature vector. Because of the particular type of sensor considered in the present study, it might be needed to also represent magnitude of the pressure values on neighbouring taxels. Indeed, because of the low sensor resolution, taking only the maximum taxel location might result in ambiguities between two different gestures producing the same maximum value. The neighbouring pressure values might then help resolve this uncertainty. This definition of the feature vector would then still belong to *templates* class but using a larger window width.

Since future versions of the skin sensor by Meka Robotics announced sampling rate close to 100Hz over USB, using the velocity information might also be added as another component of the feature vector. Following a similar approach from [10] could then be applied by processing the input signal in the frequency space using the Fast Fourier Transform (FFT). Note that due to the low sampling rate, such an approach was not attempted in the present experiment.

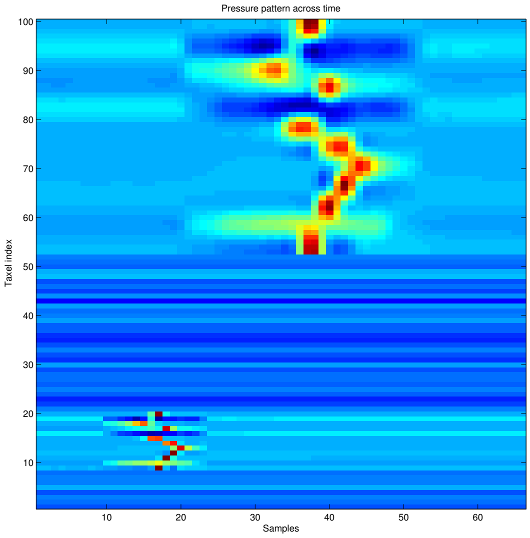


Figure 4: Test sample for an hypothetical sensor with 100 taxels. The same gesture is reproduced at different spatial scale.

5.2 Dimensionality reduction

Representing the full normalized input data into the feature vector was possible in this experiment due to the low number of taxels (only 12 per sensor). However, any realistic HRI setup would likely involve a much higher number of taxels. For example, to cover the forearm of Meka Robotics' flagship product, the M1 humanoid robot, 384 taxels were used. Clearly, a fully-covered humanoid robot with a comparable technology would easily need more than a thousand taxels.

As in [1] or [3], PCA could be applied to reduce the dimensionality of the input data. Figure 4 shows a sample collected over an hypothetical sensor array of 100 taxels. The same gesture was reproduced twice but at different scale and PCA was executed over it. Among all collected samples, and no matter the spatial scale, it appeared that only 6 components were sufficient to account for more that 98% of the variability. These preliminary results indicate that an aggressive dimensionality reduction could be obtained through PCA. Furthermore, the projection into the principal component space would be independent from the spatial scale.

5.3 Segmentation

Because multiple gestures might be executed simultaneously, but also to ensure continuous processing of the data stream, segmentation techniques might have to be applied in order to isolate portions of the input signal that might potentially contain gesture patterns. A multi-scale k-means approach as in [3] might be of interest for solving this problem. Other features might also be discovered by analyzing the frequency spectrum of the signal for systematic cues indicating the beginning and end of a gesture.

6 Conclusion

A tactile gesture recognition algorithm was presented based on Hidden Markov Models. Each gesture to be classified was represented by a corresponding 8 states left-to-right HMM previously trained over 50 examples using the Expectation Maximization algorithm. For evaluation, the Verterbi algorithm was executed over every HMM and the model with the highest likelihood over the state sequences given the input observations was chosen for the instance label. Due to the simple nature of the feature vector, but also because of the small number of gestures types and associated training examples, the EM algorithm resulted in models over-specialization for *weak* features of the data. Under those conditions, only 2 states were found to be necessary and thus the 6 states assumed in the initial model were not involved at all in the recognition process. Clearly, a better feature vector definition would have to be used in future work. Using the index taxel with the highest pressure value was proposed as a viable approach. Additional characteristics from the input signal such as the velocity of the gesture or even the amplitude coefficients obtained by FFT could be added to the vector. Tactile sensing has received little attention in Human Robot Interaction. With the advent of new technologies in this field, gesture recognition approaches such as the one proposed here will become of great interest.

7 Acknowledgement

I would like to thank Meka Robotics LLC for having provided me with one of their skin sensor unit for this project.

References

[1] Sylvain Calinon and Aude Billard. Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 105–112, New York, NY, USA, 2005. ACM.

[2] Simon Gunter and Horst Bunke. Hmm-based handwritten word recognition: on the optimization of the number of states, training iterations and gaussian components. *Pattern Recognition*, 37(10):2069 – 2079, 2004.

[3] Wu Hai and Alistair Sutherland. Dynamic gesture recognition using pca with multiscale theory and hmm. In *SPIE International Symposium on Multispectral Image Processing and Pattern Recognition, Vol. 4550*, pages 132–139, 2001.

[4] Jianying Hu, Michael K. Brown, and William Turin. Hmm based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1039–1045, October 1996.

[5] Xuedong Huang, Yasuo Ariki, and Mervyn Jack. *Hidden Markov Models for Speech Recognition.* Columbia University Press, New York, NY, USA, 1990.

[6] John Makhoul, Thad Starner, Richard Schwartz, and George Chou. On-line cursive handwriting recognition using speech recognition methods. In *In ICASSP*, pages 125–128, 1994.

[7] R. Nag, K. H. Wong, and F. Fallside. Script recognition using hidden Markov models. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pages 2071–2074, 1986.

[8] Lawrence Rabiner. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[9] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[10] Jie Yang and Yangsheng Xu. *Hidden Markov Model for Gesture Recognition.* May 1994.