JACHMM: A JAVA-BASED CONDITIONED HIDDEN MARKOV MODEL LIBRARY

Stefan Ultes, Robert ElChabb, Alexander Schmitt and Wolfgang Minker

Institute of Communications Technology
Ulm University
Albert-Einstein-Allee 43
89081 Ulm, Germany

ABSTRACT

We present JaCHMM, a Java implementation of a *conditioned* Hidden Markov Model (CHMM), which is made available under BSD license. It is based on the open source library "Jahmm" and provides implementations of the Viterbi, Forward-Backward, Baum-Welch and K-Means algorithms, all adapted for the CHMM. Like the Hidden Markov Model (HMM), the CHMM may be applied to a wide range of uni- and multimodal classification problems. The library is intended for academic and scientific purposes but may be also used in commercial systems. As a proof of concept, the JaCHMM library is successfully applied to speech-based emotion recognition outperforming HMM- and SVM-based approaches.

Index Terms— statistical machine learning library, emotion recognition, spoken dialogue systems

1. INTRODUCTION AND RELATED PRIOR WORK

Recent developments in the field of consumer electronics have brought speech-based human-computer interaction to a broad audience. Digital companions such as Apple's SIRI or TrueKnowledge's EVI have shown that speech and tactile interaction may easily go together. While these and other Spoken Dialogue Systems (SDSs) have made large progress in recent years, those systems are still mainly static in terms of what they prompt to the user and static in terms of how they treat the user notwithstanding the user-specific properties and the course of the previous conversation. The rising complexity of SDS demands for innovative techniques that help rendering future systems interaction-aware for enabling adaptivity. Ultimately, this will lead to more natural interactions, higher acceptance and raised usability of SDS. With the knowledge gained during the interaction, a dialog system would be capable of adapting its strategy, similar as a real, human dialogue partner.

Pattern recognition techniques are the driving force behind modeling and classifying human-machine interaction and human behavior. Therefore, powerful algorithms and software libraries are required to advance adaptivity in uni- and multimodal dialogue systems. They allow for recognizing (and finally adapting to) the user. With the help of trained data-driven, stochastic models, specific target variables may be predicted, which are part of static user properties, e.g., the gender and age of the user, or dynamic user properties like the emotional state, the intoxication level, or the user's current interest level in a task.

Discriminative classifiers are commonly used for classification of static feature vectors, such as Artificial Neural Networks (ANN), Support Vector Machines (SVM) (Vapnik [1]) or rule learners. For classifying sequential information, such as in speech or gesture recognition, Hidden Markov Models (HMMs) have shown superior results (e.g., Rabiner et al. [2]). For work on speech-based emotion recognition, which is, despite its sequential character, mainly based on static feature vector models (e.g., Schmitt et al. [3, 4]) with Hidden Markov Models playing only a minor role (e.g., Schuller et al. [5]), we explore the use of Conditioned Hidden Markov Models (CHMMs), originally published by Glodek et al. [6]. While CHMMs may be applied to the same types of tasks as classical HMMs, they offer essential advantages:

- **Class probabilities**: A CHMM directly provides a class probability $p(y|\mathrm{x}^{(n)}, \lambda)$ instead of an output probability $p(\mathrm{x}^{(n)}|\lambda)$. For the latter, multi-class recognition is usually performed by instantiating a model for each class separately. (Here, y are the class labels, $\mathrm{x}^{(n)}$ the observation sequence and λ the set of model parameters.)
- **Extra training data**: As all classes are combined in one single model in the CHMM, hidden states can be shared by several labels thus taking advantage of extra training data.
- **Label-observation-relation**: The CHMM may be used for data where a label is related to a whole sequence of observations as well as where a label is related to one single observation. Here, it is more flexible than the HMM which has limited options for the latter case. The only option in the classical HMM is to statically assign each hidden state to one of the labels.

- **Easy multimodal fusion**: As the CHMM directly provides class probabilities, lateral fusion becomes much easier. In the classical HMM, rations for each feature decision have to be computed to determine the most likely class. The models depend on the class and the modality and cannot differ in complexity. Therefore, no intuitive weighting of the classifiers is possible and the uncertainty does not necessarily sum up to one.

Exploration of the CHMM for speech-based emotion recognition was encouraged by previous work by Glodek et al. [6, 7], who showed that CHMM based approaches can outperform classical HMMs. While multimodal laughter detection on audio-visual data from the FreeTalk data set only resulted in about equal performance of both approaches [6], the HMM is outperformed by the CHMM for action-detection using a two-layered approach evaluated in [7]. The CHMM reached a *F1-score* of 0.53 compared to a score of 0.32 for classical HMMs, both applied to unseen, unsegmented data.

Consequently, we publish the JaCHMM – a Java implementation of the Conditioned Hidden Markov Model – publicly available under the BSD license. In Section 2, a complete formal description of the CHMM and all implemented equations is presented. For clarity and readability reasons, it not only contains previously unpublished equations but also a summary of previously published equations by Glodek et al. [6]. Section 3 lists the prominent features of the JaCHMM and gives details on how to obtain the library. We present an example application of the JaCHMM in the task of speech-based emotion recognition along with a brief performance analysis in Section 4 before concluding in Section 5.

2. CONDITIONED HMM

Conditioned Hidden Markov Models are an extension of classical HMM, originally published by Glodek et al. [6]. The principle operation method of the CHMM in the time domain is illustrated by a sequence diagram in Figure 1.

2.1. Model Description

Like the classical HMM, the CHMM also consists of a discrete set of hidden states $w_i \in W$ and a vector space of observations $\mathbb{X} \subseteq \mathbb{R}^n$. A separate emission probability $b_i(\mathrm{x}^{(t)})$ is linked to each state defining the likelihood of observation $\mathrm{x}^{(t)} \in \mathbb{X}$ at time t while being in state w_i. Further, $a_{ij,y} = p(\mathrm{w}^{(t)} = w_j | \mathrm{w}^{(t-1)} = w_i, \mathrm{y}^{(t)} = y)$ defines the transition probability of transitioning from state w_i to w_j. In contrast to the classical HMM, the transition probability distribution also depends on the class label $y \in Y$. This results in the transition matrix $\mathbf{A} \in \mathbb{R}^{|W| \times |W| \times |Y|}$.

Furthermore, the meaning of the initial probability $\pi_{i,y} = p(\mathrm{w}^{(1)} = w_i | \mathrm{y}^{(1)} = y)$ for state w_i is altered. It additionally represents the label probability for label y at any time with the corresponding matrix $\boldsymbol{\pi} \in \mathbb{R}^{|W| \times |y|}$.

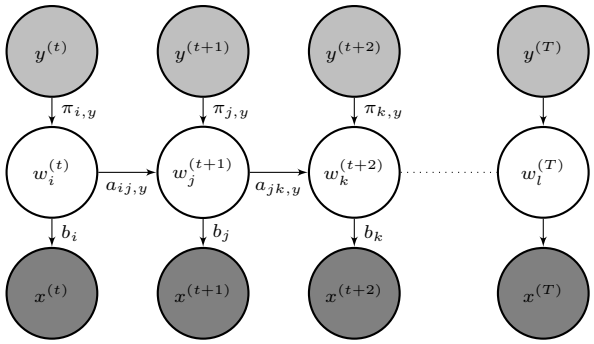

Fig. 1. General graphical representation of the CHMM model in the discrete time domain. For each time step t, $y^{(t)}$ represents the most likely label and $w_i^{(t)}$ the most likely hidden state given observation $x^{(t)}$. b_i represents the probability for the observation and $\pi_{i,y}$ the label probability. $a_{ij,y}$ defines the probability of transitioning from state $w_i^{(t)}$ to state $w_j^{(t+1)}$.

According to Glodek et al. [6], the likelihood of an observation sequence $\mathrm{x}^{(n)}$ with corresponding label sequence $\mathrm{y}^{(n)}$ is given by

$$
\begin{aligned}
&p(\mathrm{x}^{(n)}, \mathrm{w}^{(n)} | \mathrm{y}^{(n)}, \lambda) \\
&\quad = \sum_{w \in W} p(\mathrm{w}^{(1)} = w | \mathrm{y}^{(1)}, \boldsymbol{\pi}) \\
&\qquad \cdot \prod_{t=2}^{T} p(\mathrm{w}^{(t)} = w_j | \mathrm{w}^{(t-1)} = w_i, \mathrm{y}^{(t)}, \mathbf{A}) \\
&\qquad \cdot \prod_{t=1}^{T} p(\mathrm{x}^{(t)} | \mathrm{w}^{(t)} = w_j, \theta),
\end{aligned} \tag{1}
$$

where $\mathrm{w}^{(n)}$ denotes the sequence of the hidden states.

The emission probability $b_j(\mathrm{x}^{(t)}) = p(\mathrm{x}^{(t)} | \mathrm{w}^{(t)} = w_j, \theta)$ may be either modeled with discrete probabilities, or with continuous probabilities using a one- or multidimensional Gaussian Mixture Model (GMM) with the parameter set $\theta = \{\{\phi_{j,k}\}_k^K, \{\mu_{j,k}\}_k^K, \{\Sigma_{j,k}\}_k^K\}$. The parameter set λ describing the complete CHMM is defined as $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \theta\}$.

2.2. Learning

The learning phase consists of two parts: *initialization* and *training*.

For *initialization*, we adapted the k-means algorithm [8] for the use with CHMMs. Here, the number of clusters k corresponds to the number of hidden states. After clustering initial observation sequences with their corresponding label sequences, the transition probabilities are updated according to the transitions between the clusters, given the labels. The initial probabilities are updated according to the cluster and the corresponding label that each element belongs to.

Training may be either performed also by applying k-means or by using the Baum-Welch algorithm. The perfor-

mance of the latter is heavily dependent on the initialization. When comparing the HMM explained by Rabiner et al. [2] to the CHMM, several changes[1] must be applied to the Baum-Welch algorithm.

The αs and βs of the Forward-Backward algorithm in accordance with Glodek et al. [6] are

$$\alpha_{t,y}(j) = b_j(\mathrm{x}^{(t)}) \cdot \sum_{i \in W} a_{ij,y} \cdot \alpha_{t-1,y}(i) \tag{2a}$$

$$\alpha_{1,y}(j) = b_j(\mathrm{x}^{(1)}) \cdot \pi_{j,y} \tag{2b}$$

$$\beta_{t,y}(i) = \sum_{j \in W} a_{ij,y} \cdot b_j(\mathrm{x}^{(t+1)}) \cdot \beta_{t+1,y}(j) \tag{3a}$$

$$\beta_{0,y}(i) = \sum_{j \in W} \pi_{j,y} \cdot b_j(\mathrm{x}^{(1)}) \cdot \beta_{1,y}(j),\ \beta_{T,y}(i) = 1 \tag{3b}$$

Computation of the state beliefs $\gamma_{t,y}(j)$ and the transition beliefs $\xi_{t-1,t,y}(i,j)$ is then performed by using

$$\gamma_{t,y}(j) = \frac{\alpha_{t,y}(j) \cdot \beta_{t,y}(j)}{p(X)}\ , \tag{4}$$

$$\xi_{t-1,t,y}(i,j) = \frac{\alpha_{t-1,y}(i) \cdot b_j(\mathrm{x}^{(t)}) \cdot a_{ij,y} \cdot \beta_{t,y}(j)}{p(X)}\ , \tag{5}$$

where $\sum_{t=1}^{T-1} \gamma_{t,y}(i)$ is the expected number of transitions starting from w_i given y and $\sum_{t=1}^{T-1} \xi_{t-1,t,y}(i,j)$ is the expected number of transitions from w_i to w_j given y.

Parameter learning is performed after evaluation of N sequences, updating the initial probabilities using the following formula

$$\begin{aligned}\pi_{i,y} &= \frac{\textit{expected number of times being in } w_i \textit{ at } t=1 \textit{ given } y}{\textit{expected number of times being in all } w \textit{ at } t=1 \textit{ given } y} \\ &= \frac{\sum_{l=1}^{N} \delta_{\mathrm{y}^{(1)(l)}=y} \gamma_{1,\mathrm{y}^{(n)(l)}}(i)}{\sum_{l=1}^{N} \sum_{j \in w_1} \delta_{\mathrm{y}^{(1)(l)}=y} \gamma_{1,\mathrm{y}^{(n)(l)}}(j)}\end{aligned} \tag{6}$$

where $\sum_{i=1}^{n} \pi_{i,y} = 1$ and δ is the Kronecker delta.

The update for the transition probabilities after evaluating N sequences is

$$\begin{aligned}a_{ij,y} &= \frac{\textit{expected number of transitions from } w_i \textit{ to } w_j \textit{ given } y}{\textit{expected number of transitions from } w_i \textit{ given } y} \\ &= \frac{\sum_{l=1}^{N} \sum_{t=0}^{T-1} \xi_{t-1,t,\mathrm{y}^{(n)(l)}}(i,j) \delta_{\mathrm{y}^{(t)(l)}=y}}{\sum_{l=1}^{N} \sum_{t=0}^{T-1} \gamma_{t,\mathrm{y}^{(n)(l)}}(j) \delta_{\mathrm{y}^{(t)(l)}=y}}\end{aligned} \tag{7}$$

where

$$\forall_{y \in Y} \sum_{j=1}^{n} a_{ij,y} = 1.$$

The emission probabilities can be computed in accordance with the methods presented by Rabiner et al. [2]. As the state beliefs depend on y, a sum over all labels has to be applied in order to create label independent emission probabilities.

[1] Changes in Eq.: 19, 20, 24, 25, 27, 37, 40a, 40b, and 40c from [2]

2.3. Evaluation

For evaluation, the Viterbi algorithm is applied generating the most likely label sequence. The label probability $p(y|\mathrm{x}^{(n)})$ for label y and observation sequence $\mathrm{x}^{(n)}$ is calculated by

$$p(y|\mathrm{x}^{(n)}) = \frac{p(\mathrm{x}^{(n)}, y)}{\sum_y p(\mathrm{x}^{(n)}, y)}, \tag{8}$$

where $p(\mathrm{x}^{(n)}, y) = p(\mathrm{x}^{(n)}|y)p(y)$, $p(y)$ is the prior probability over all labels and

$$p(\mathrm{x}^{(n)}|y) = \sum_{i \in \mathrm{w}^{(T)}} \alpha_{T,y}(i) \tag{9}$$

the probability for observation sequence $\mathrm{w}^{(T)}$ given y.

3. JACHMM

Based on the Jahmm library – a HMM library based on Java by Francois [9] – the JaCHMM library has been created implementing the Conditioned Hidden Markov Model. Here, some important features are outlined.

Labels JaCHMM may be used for data where a label is related to a whole sequence of observations as well as where a label is related to one single observation.

Observations Observations may either be discrete or continuous. Therefore, different types of observation probability distributions have been implemented, i.e., discrete probabilities and Gaussian mixture models for continuous observations.

Initialization and Training Initialization is implemented using the k-means algorithm, which can also be used for training. Additionally, training can be performed by using the traditional Baum-Welch algorithm.

Computational Efficiency In order to increase the computational efficiency of JaCHMM, reasonable independence assumptions regarding the transition probability $a_{ij,y} = p(\mathrm{w}^{(t)} = w_j|\mathrm{w}^{(t-1)} = w_i, \mathrm{y}^{(t)} = y)$ have been introduced, resulting in the simplified version $a_{ij,y} = p(\mathrm{w}^{(t)} = w_j|\mathrm{w}^{(t-1)} = w_i) \cdot p(\mathrm{w}^{(t)} = w_j|\mathrm{y}^{(t)} = y)$. Within the JaCHMM library, both variants are implemented.

Algorithms For their application in the Conditioned Hidden Markov Model, prominent algorithms known from HMMs were adapted, e.g., the Viterbi, Forward-Backward, Baum-Welch, or K-Means algorithm.

Both an application programming interface (API) and a command-line interface is provided by the JaCHMM library to ensure its flexible application. The library is organized by several packages encapsulating the core components (model

Table 1. *Results of anger recognition of CHMM, HMM, and SVM.*

	window	step	UAR
HMM	40ms	20ms	0.66
CHMM	40ms	20ms	**0.67**
SVM	whole file		0.59

structure and basic algorithms), the probability distributions of the observations, the learning algorithms, input and output methods, and the command-line interface. Further, a toolbox for creating observations and for evaluation of sequences given a model exists. This packet structure allows for easy integration into software projects of the whole library as well as solely using parts of it. The command-line interface offers commands for creating, initializing, learning, and evaluating CHMMs enabling the library to work as stand-alone software.

Like the JaHMM, the JaCHMM is designed to achieve reasonable performance without making the code unreadable. Consequently, it offers a good way of applying the Conditioned Hidden Markov Model in various tasks, e.g., for scientific or teaching purposes.

The JaCHMM library is available online under the BSD license at `http://nt.uni-ulm.de/ds-jachmm`.

4. APPLICATION

For a proof-of-concept application of the JaCHMM library, we chose the task of speech-based emotion recognition based on the LEGO corpus [10] distinguishing between the classes "angry", "neutral" and "garbage". The LEGO corpus contains annotated audio files of calls to the Lets Go bus information system (cf. Raux et al. [11]). For this evaluation, features have been derived from 4832 audio files with an average length of $1.65s$ ($\pm 1.46s$) using a windowing approach resulting in a total of 25 acoustic features per window with a window length of $40ms$ and a step width of $20ms$ ($20ms$ overlap). Features were, among others, Mel-Frequency-Cepstral-Coefficients (MFCCs), power, intensity, pitch, jitter, shimmer, and formants. For measuring the recognition performance, the *Unweighted Average Recall (UAR)* is used which is the arithmetic average of all class-wise recalls.

Initialization and training of the CHMM was performed using k-Means as it has been shown to be more robust in case of limited data. The tests were performed using 6-fold cross-validation in order to guarantee for generalizable results. To determine the optimal performance with respect to the number of hidden states, window width, and step width, simple linear exploration of the search space was applied. Results can be seen in Table 1.

For comparison, the same experiment was conducted with classical HMMs using Jahmm. One separate HMM was created for each class. The results for *UAR* show that the CHMM outperforms the baseline of HMM classification as well as a previous approach for emotion recognition of simple SVM classification performed in accordance with Schmitt et al. [4].

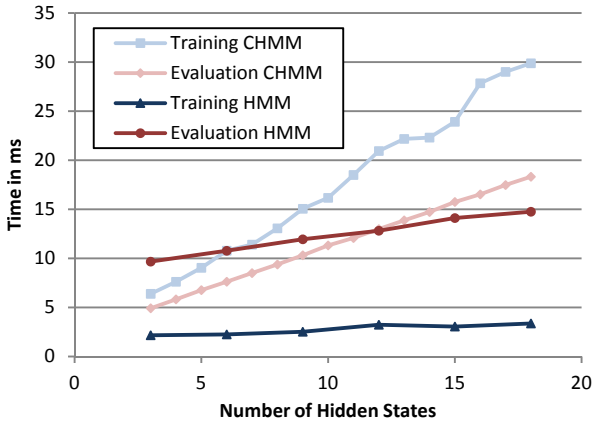

Fig. 2. *Average time needed for training and evaluation of the CHMM and HMM per observation sequence. For the HMM, the total number of hidden states and the time for training and evaluating summed up over all HMMs is used.*

Figure 2 shows the average time needed for training and evaluating the CHMM and HMM per observation sequence in ms with respect to the number of hidden states. While training of the HMM was much faster for all configurations, for a small number of hidden states, evaluation of the CHMM was faster than evaluating three HMMs. Note that for HMM performance, the total number of hidden states as well as the training and evaluation time was summed up over all HMMs.

5. CONCLUSION

In this work, we present the JaCHMM library publicly available under the BSD license, implementing a Conditioned Hidden Markov Model. The CHMM offers an easy means for classification tasks regarding sequential information for multi-class recognition tasks. The library was successfully applied to speech-based emotion recognition outperforming approaches using Support Vector Machines as well as Hidden Markov Models. Both, the successful application of the JaCHMM library and work by Glodek et al. show the high potential of the CHMM. While the application showed that, for a high number of hidden states, the CHMM needs more time for training and evaluation than classical HMMs do, the total time needed for applying CHMMs with a sufficient number of hidden states to achieve good performance may still be regarded as good.

6. ACKNOWLEDGMENTS

This work was supported by the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" which is funded by the German Research Foundation (DFG).

7. REFERENCES

[1] Vladimir N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[2] Lawrence R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.

[3] Alexander Schmitt, Tobias Heinroth, and Jackson Liscombe, "On nomatchs, noinputs and bargeins: Do non-acoustic features support anger detection?," in *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue, SigDial Conference 2009*, London (UK), Sept. 2009, Association for Computational Linguistics.

[4] Alexander Schmitt, Roberto Pieraccini, and Tim Polzehl, *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*, chapter 'For Heavens sake, gimme a live person!' Designing Emotion-Detection Customer Care Voice Applications in Automated Call Cent, Springer, Sept. 2010.

[5] Björn Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 2, pp. II–1.

[6] Michael Glodek, Stefan Scherer, and Friedhelm Schwenker, "Conditioned hidden markov model fusion for multimodal classification," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*. Aug. 2011, pp. 2269–2272, International Speech Communication Association.

[7] Michael Glodek, L. Bigalke, G. Palm, and Friedhelm Schwenker, "Recognizing human activities using a layered hmm architecture," in *Workshop New Challenges in Neural Computation 2011*, 2011, pp. 38–41.

[8] Vance Faber, "Clustering and the continuous k-means algorithm," *Los Alamos Science*, , no. 22, pp. 138–144, 1994.

[9] Jean-Marc Francois, "Jahmm - An implementation of HMM in Java," 2006.

[10] Alexander Schmitt, Stefan Ultes, and Wolfgang Minker, "A parameterized and annotated corpus of the cmu let's go bus information system," in *International Conference on Language Resources and Evaluation (LREC)*, 2012.

[11] Antoine Raux, Dan Bohus, Brian Langner, Alan W. Black, and Maxine Eskenazi, "Doing research on a deployed spoken dialogue system: One year of lets go! experience," in *Proc. of the International Conference on Speech and Language Processing (ICSLP)*, Sept. 2006.