Abstract

The availability of wearable and ambient sensors allows more information to be captured for human activity recognition. However, noises and signal disconnections are common in complicated environment. Resolving noise and variability from different inputs, as well as accounting for the human-object interactions is challenging under complex settings. To address the challenge, I present a novel Hidden Markov Model variant that includes both coupled and factorized states for estimation and learning problems.

In this thesis, I provide the detailed formulation of selective factorized coupled hidden markov model (SFCHMM), including its model definition, forward-backward procedure for conditional observation probabilities, optimal state path decoding and parameter estimation. In addition to the algorithmic discussion, I also test the model by simulating on synthetic CHMM processes and applying to a real world sensor-rich benchmark dataset that recorded human daily activities. The performance analysis based on the experiments demonstrates that this model is capable of consolidating the fuzzy information from a collective pool of sensors and improving human activity recognition in interactive context, which is highly applicable to real world settings such as surveillance, smart home and multimedia games.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Early works on human activity recognition are mostly done using explicit models where features are hard coded. An example is Hogg's early work of modeling human walking with spatio-temporal constraints on the movement patterns [14]. While the explicit models are useful for providing solution to particular applications, it might require way too much manual labor to conduct specification of model parameters for a task that involves a huge input, which is more often than not the common case. Human intervention such as designing example temporal, deciding what irrelevant information to leave out and what distance metric to choose is also required for exemplar-based models. Modeling a complex scene, the inherent structure and semantics of complex activities require higher level representation and methods.

Statistical approach has been applied to activity recognition [22] [35]. One advantage is that they scale up well for real world applications compared to methods that rely on spatio-temporal or sliding-window search. However this approach is mostly restricted to applications of discriminative models and anomaly detection.

The development of parametric machine learning models makes automatic learning possible. Furthermore, these models allow some knowledge of the complexity of the problems, such as dimensions of hidden state space to be built in and let the models to adapt to different types of inputs required by specific application. Roweis and Ghahramani provided an example list of such generative models with the relationships explained [31].

Table 1.1: Generative Model Relationships

Initial	Extension	Final
Gaussian	Mixture	GM
Gaussian	Reduced Dimension	PCA
GM	Dynamic	HMM
HMM	Coupling	CHMM
HMM	Variable Length	VLMM
HMM	Hierarchy	DBN
DBN	Utility	DDN

Its comparable counterpart, such as deep learning, shares some of these advantages and is in an even more automated fashion [1] [17] [18]. A disadvantage of deep learning though is that it normally suffers from poorer interpretability than the former. Probabilistic graphical models combine both probability theory and graph theory and are able to detect both complex human activities and simple human actions [27].

Most recently vision-based human activity recognition [29] has attracted a lot of research interest. This type of human activity tasks is very challenging in that it involves large visual inputs and activity categories. Unsupervised learning method such as deep learning proves its strong power in achieving good results in such tasks. Ji Shuiwang et al developed a novel 3D convolutional neural network that can act directly on raw 3D inputs and automatically perform feature construction for activity recognition [18]. Quoc V. Le et al presented an extension of Independent Subspace Analysis algorithm (ISA) for unsupervised motion feature learning from video [19]. This method performs surprisingly well when combined with deep learning techniques such as stacking and convolution to learn hierarchical representation.

We study the human activity recognition problem in a specific application context where there are human-object interactions involved. And the activities are captured by a rich pool of sensors targeted at both human and the object. Due to complicated environment setting, significant sensor information variance among same and different sources, or even failure inevitably happens. Our goal is to develop a learning algorithm that preserves as much temporal and human-object interaction information as possible. At the same time, it should be relatively resistant to the situation of partial information loss.

The basis of our algorithm is Hidden Markov Models. As we can see from the investigation that follows, this kind of model is not only good at recognition tasks that involve temporal sequences, but is flexible in taking different types of inputs as well. More importantly, it is powerful in modeling complicated activities by using some sophisticated structure derived from the basic form. Also when compared with discriminative models with proven excellence in performance, such as deep learning, this kind of model enjoys the advantage of being easily adaptive to different applications and requires much less time in training.

This thesis begins with an overview of various kinds of machine learning algorithms for human activity recognition in Chapter 2, where the trade-off between different approaches is also discussed. In addition, we look at existing variants of hidden Markov models and the motivation for our model. In Chapter 3, we present the formal definition and derivation of our model, followed by Chapter 4, where we illustrate how to solve inference and learning problems by using our selective factorized coupled hidden Markov model. The details of synthetic and empirical experiments are included in Chapter 5. Finally we conclude our discussion in Chapter 6.

Chapter 2

Background and Existing Methodologies

2.1 Hidden Markov Model

2.1.1 Standard Form and Conditional Independence

A Hidden Markov Models (HMM) is a generative model of the joint probability of a collection of random variables [30]. Graphically, it can be represented as an unfolded mixture model whose states at time t are conditioned on those at time $t-1$, also known as the first-order Markov property. The first-order Hidden Markov Models can be extended to n-th order, where n is greater than one.

A standard first-order HMM with N hidden states, M possible observations and length T is defined by the parameter set $\theta = \{A, B, \pi\}$. π is the initial state probability parameter set, which is also called prior. A is a set of state transition probabilities. And B is the set of observation probabilities conditioned on the hidden states. Suppose we have the observation and hidden state at time t denoted as o_t and q_t, the independence assumptions can be concluded as follows:

$$P(q_t \mid q_{t-1}, o_{t-1}, ..., q_1, o_1) = P(q_t \mid q_{t-1})$$

$$P(o_t \mid o_t, q_t, o_{t-1}, q_{t-1}, ..., o_1, q_1) = P(o_t \mid q_t)$$

where the t-th hidden state only depends on the previous hidden state, and the t-th observation is independent of other variables except for the current hidden state. Note that the observations or emissions can be either discrete or continuous. Meanwhile, the probability constraints also apply. The initial state probabilities and elements in A and B must satisfy:

$$\textstyle\sum_{i=1}^{N} \pi_i = 1, \quad \sum_{j=1}^{N} a_{ij} = 1, \quad \sum_{k=1}^{M} b_i(k) = 1$$

In the continuous case, the pdf's of observations o_t at time t is often parameterized by a multivariate Gaussian distribution [2]. A single multivariate Gaussian output distribution is:

$$b_i(k) = P(O|s_i) = \mathcal{N}(O; \mu^i, \Sigma^i)$$

And can be extended to the case of M-component Gaussian mixture model:

$$b_i(k) = P(O|s_i) = \sum_{m=1}^{M} c_{im}\mathcal{N}(O; \mu^{im}, \Sigma^{im})$$

2.1.2 Scope of Usage

There are mainly three types of problems to solve for HMM's [4]. The first one is to find the forward probability that the model generates a given sequence, i.e. to estimate $P(O \mid \theta)$ given an observation sequence $O = o_1, o_2, ..., o_T$ and a model parameter set $\theta = \{A, B, \pi\}$. The second one is to find the best state path given both observations and a model parameter set $\theta = \{A, B, \pi\}$. This problem can be solved efficiently by Viterbi algorithm, which is a type of dynamic programming algorithm. The last one

Table 2.1: Three Important Problems of HMM

Problem	Algorithm	Computation Complexity
estimation ($P(O \mid \theta)$)	forward, backward algorithm	$O(TN^2)$
inference($Q^* = argmax_Q P(Q \mid O)$)	Viterbi algorithm	$O(TN^2)$
learning ($\theta^* = argmax_\theta P(Q \mid \theta)$)	EM (Baum-Welch)	$O(TN^2)$

is to find the best $\theta = \{A, B, \pi\}$ that maximizes $P(O \mid \theta)$. This problem can be framed as a constrained optimization problem of finding $\theta^* = argmax_\theta P(O \mid \theta)$. It can be solved by using Baum-Welch, which is essentially a type of EM algorithm.

We can summarize the scope of applying HMM models along with typical asymptotic computation cost in the table.

2.2 Hidden Markov Model Variations

2.2.1 Forms of Variations

Variations of HMM's develop more sophisticated structure upon the most basic Hidden Markov Models. There can be temporal relaxation such as Hidden semi Markov Model (HSMM) [11] [15] [23] or hierarchical structure [13], such as Layered Hidden Markov Models (LHMMs) and Hierarchical Hidden Markov Models (HHMMs).

A dynamically multi-linked hidden Markov model with the structure determined by Bayesian information criterion was also proposed. Local and global features are used for representation and to have a uniform distribution for modeling the global activity, a duration state is added in the DBN model, which is similar to hidden semi Markov Model [10].

Each of the models has its own strengths and weaknesses in terms of performance in different applications and computation cost. If the underlying process does not have a geometrically distributed duration, hidden semi Markov model is more appropriate for modeling the process. However, statistical inference for hidden semi-Markov models is more difficult than in hidden Markov models since algorithms like the Baum-Welch algorithm are not directly applicable, and must be adapted requiring more resources. The hierarchical hidden Markov models utilize its structure to solve a subset of the problems more efficiently and can be used to extract higher-level semantic meanings. However, the methods for estimating the HHMM parameters and model structure are more complex than for the HMM.

For the interest of our investigation, coupled Hidden Marko Models (CHMM) [36], [5] is a more relevant variation. It has been compared to HMM to model activities that involve interactions among multi-agents, and has been shown to generate better recognition results [28]. In order to account for the varied sub-event duration and states, a coupled hidden semi Markov model (CHSMM) has been proposed which allows composition states for both of the channels in the CHMM model. The algorithmic complexity can be reduced to $O(C^2NT)$, where C is the number of channels, N is the number of states and T is the number of time steps. The CHSMM model has produced 20% to 30% increase in performance on simulated test and 50% increase in real data [24]. The main focus of our work is not on temporal variance in the hidden states, but rather to address the issues with observations variance or missing. Thus the model is developed upon CHMM, though it can be further adapted to semi-Markovian processes too.

2.2.2 An Architectural Perspective

Following is a collection of hidden Markov models that researchers have been using for multi-agent activities recognition. The hidden semi Markov models (HSMM's) allow the hidden state to extend more than a single time step and the parallel hidden markov models (PaHMM's) develop multi-chains for multiple agents. Parallel hidden semi Markov models (PaHSMM's) combine the merits of both HSMM's and PaHMM's. The hierarchical semi parallel hidden Markov models extract higher-level semantic meanings from observations whereas the hierarchical parallel hidden semi Markov models encode lower-level hidden state variables.

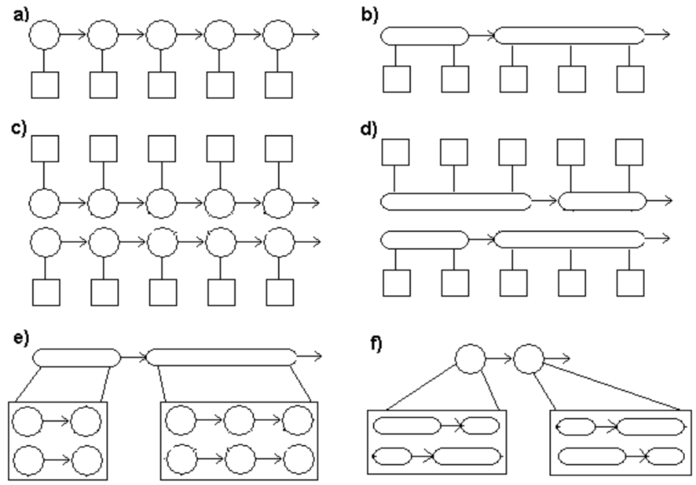

Figure 2-1: An Illustration of HMM Variants Structures [25]: a)HMM b)HSMM c)PaHMM d)PaHSMM e)HSPaHMM f)HPaHSMM

HSPaHMM and HPaHSMM [25] combine the features of all these 3 classes (hierarchical, semi, multi-channel) of extensions. And they are able to derive higher level sematic concepts from raw observation inputs, avoid significant variance in low-level feature variance and reduce the variance in observations where different people perform activities in slightly different style. Pradeep Natarajan et al also provided a graphical representation of these of models.

2.2.3 Coupled Hidden Markov Models

Coupled Hidden Markov Models (CHMM's) can be viewed as HMM extended to multi-dimensional forms [5]. Here the current state is dependent on the states of its own chain and that of the neighboring chain at the previous time step. In addition to the parameter set established in the hidden Markov Models, the cross-chain state transitional probabilities are introduced. The posterior state probability for CHMM's is expressed as:

$$P(S \mid O) = \frac{P_{s_1} p_{s_1}(o_1) P_{s'_1} p_{s'_1}(o'_1)}{P(O)} \times \prod_{t=2}^{T} P_{s_t|s_{t-1}} P_{s'_t|s'_{t-1}} P_{s'_t|s_{t-1}} P_{s_t|s'_{t-1}} p_{s_t}(o_t) p_{s'_t}(o'_t)$$

where s_t s'_t o_t $o_{t'}$ denote the state and observations for each of the Markov chains that the CHMM is consist of.

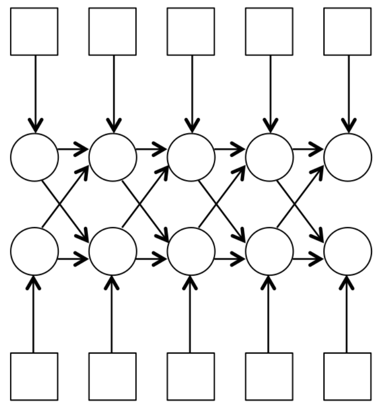

Figure 2-2: Graphical Representation of a CHMM

In a coupled hidden Markov model, the agents engaged in an activity are no longer isolated and a proper choice of CHMM can significantly improve the recognition effectiveness. Due to its nice property of being able to model interaction, CHMM has been combined with hierarchical or semi structure for complex activity recognition [33] [25]. Besides the area of human activity recognition, CHMM has also been applied to a wide range of applications including molecular sciences [33], tool-ware classifications [7] and speech recognition [20] [26] [9].

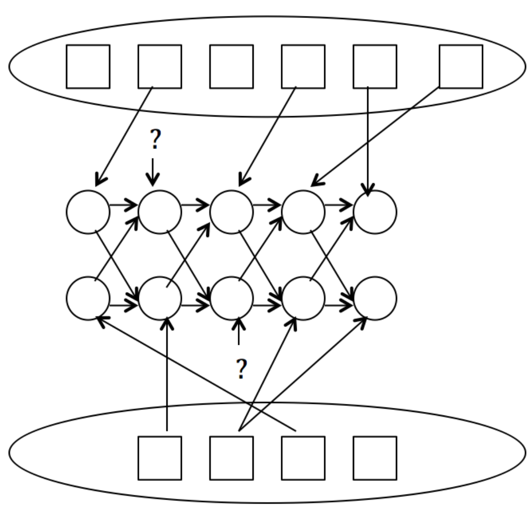

Figure 2-3: An example of a CHMM sequence with missing observations

However there's still an unaddressed gap. In situations where there are conditional dependencies between the agents in the activities, we need a model that considers the interactions across the chains. In another type of situations where partial observations are missing/of poor quality, or only one of the agents is engaged in the activity at certain time steps, we need a model that can factorize or collapse coupled states at certain transitions. This is where the selective factorized hidden markov model (SFCHMM) comes to fill in the gap.

Chapter 3

Selective Factorized Coupled Hidden Markov Models

The CHMM has already been shown as a powerful machine to model interactions among agents [5]. To further cater to situations where there should be partial information missing or sensor inputs quality deterioration, we present a HMM variant that has similar architecture as CHMM but includes both coupled as well as selective factorized states. It captures the interactions overall but also makes sure to utilize the available input set to its best.

3.1 Notations

The model definition will be extended based on the notations of standard hidden Markov models. In a standard HMM with a parameter set $\theta = \{A, B, \pi\}$, where there are N hidden states $S = \{s_1, s_2..., s_N\}$, and M observations $V = \{v_1, v_2..., v_M\}$. A and B are conditional state transition probabilities and conditional observation transitional probabilities respectively.

For any given sequence with observations $O = o_1, o_2, ..., o_T$ and hidden states $Q = q_1, q_2, ..., q_T$, the initial state distribution π, and each entry of A, B are calculated as follows:

$$\pi_i = P(q_1 = s_i)$$

$$a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i)$$

$$b_j(k) = P(o_t = v_k \mid q_t = s_j)$$

where $1 \leq i, j \leq N$,$1 \leq k \leq M$ and $t \leq T$

In a standard CHMM model, suppose the state space sizes of the two individual chains are $N^{(1)}$ and $N^{(2)}$, then the complete coupled state space is the Cartesian product of the two. Some of the coupled hidden states are likely to be obsolete or redundant, causing the computation cost to be unnecessarily high. Instead of exhausting all the possible coupled hidden state, we use a I parameter to indicate whether two states from the coupled chains are related to each other, thus reducing the size of the state space.

So now we have a new set of model parameters for our selective factorized coupled hidden Markov model $\theta = (A, B, \pi, I)$. Under the new definition of selective factorized coupled hidden markov model, the parameter set is $\theta = (A, B, \pi, I)$, where A, B, π are the transition probabilities, emission probabilities, initial probabilities and hidden state interaction set. In addition superscript of a variable indicates whether it belongs to a specific chain of the coupled model or it comes from a selective factorized term.

The prior probabilities and conditional observation transition probabilities become:

$$\pi_i^{(c)} = P(q_1^{(c)} = s_i)$$

$$b_i^{(c)}(k) = P(o_t^{(c)} = v_k \mid q_t^{(c)} = s_i)$$

where $1 \leq i, j \leq N^{(c)}$, $t \leq T$ and $c = 1, 2$. There are two types of transition probabilities. One type is for transitions on the same chain and the other is for transitions across the two chains:

$$a_{ij}^{(c,c')} = P(q_t^{(c')} = s_j \mid q_{t-1}^{(c)} = s_i)$$

where $1 \leq i \leq N^{(c)}$, $1 \leq j \leq N^{(c')}$ $t \leq T$ and $c, c' \in \{1, 2\}$.

In actual computation, we can use maximum-entropy factoring [5], [21] to project the coupled parameter back into its components:

$$\hat{a}_{ij}^{(c,c)} \propto \sqrt{\sum_l \sum_k a_{ik}^{(c,c')} a_{jl}^{(c,c')}}$$

$$\hat{a}_{ij}^{(c,c')} \propto \sqrt{\sum_j \sum_k a_{ik}^{(c,c')} a_{jl}^{(c,c')}}$$

where the $\hat{a}_{ij}^{(c,c)}$ is the transition probability along the same chain and $\hat{a}_{ij}^{(c,c')}$ is the transitional probability between the two chains. With the projected values, we can reduce the computation of using $(N^{(c)} \cdot N^{(c')}) \times (N^{(c)} \cdot N^{(c')})$ dimension matrix to $(N^{(c)} \times N^{(c)})$ and $(N^{(c)} \times N^{(c')})$ dimension matrices.

3.2 Forward and Backward Procedures

3.2.1 Forward and Backward Procedures in basic HMM's

To solve for the probability of an observation sequence give the model, i.e. $P(O \mid \theta)$, the naive way to compute is according to definition, which requires enumerating joint probability over all possible state sequence. It would lead to $O(TN^T)$ computation since for every time step there are N possible states. For a simple example of $N = 10$ and $T = 100$, the order of computation is 10^{101}. The forward-backward procedure helps solve this computation infeasibility of the naive approach.

We first define the forward probability as $\alpha_t(i) = P(o_1, o_2, ..., q_t = s_i \mid \theta)$. In other words, $\alpha_t(i)$ is the probability of observing o_1 till o_t and hidden state s_i at time t, given the model parameters θ.

The computation using induction is as follows:

- Initialization

  $\alpha_1 = \pi_i b_i(o_1)$ for $1 \leq i \leq N$, where α_1 is initialized as the joint probability of state S_i and initial observation o_1.

- Induction Step

  $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(o_{t+1})$, where $1 \leq t \leq T-1$ and $1 \leq j \leq N$.

- Termination

  $P(O \mid \theta) = \sum_{i=1}^{N} \alpha_T(i)$.

The complexity under this scheme is $O(N^2T)$, which is significantly less than the naive approach. Also the forward probability induction is sufficient to solve for our first question, which is finding out how likely a sequence of observations is generated by a given model.

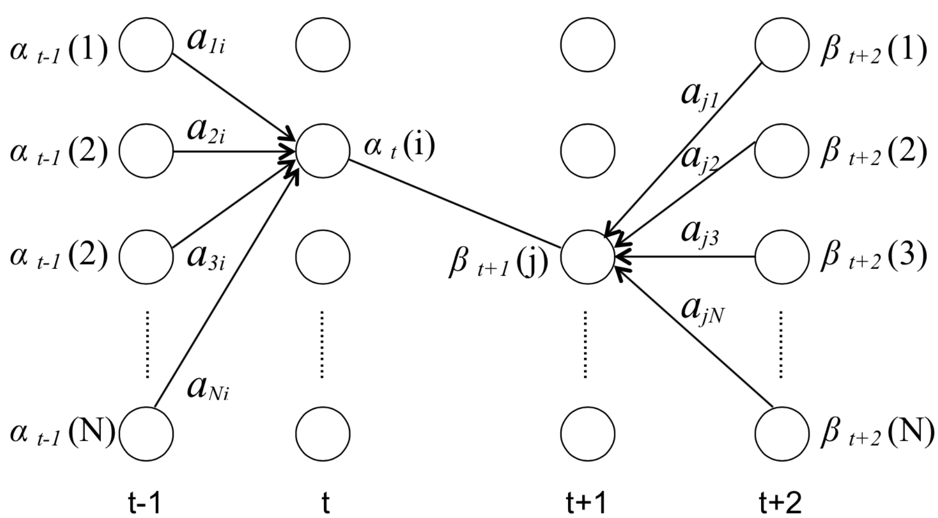

Figure 3-1: An Illustration of forward, backward procedures

In essence, the forward probabilities α_i's can be viewed as a trellis data structure, where the probabilities of previous sequence are summarized to N nodes at the current step.

Its counterpart backward probability $\beta_t(i)$ is calculated by induction in a similar fashion, except that it is derived from backwards. $\beta_t(i)$ is the probability of observing o_{t+1} till o_T, given the hidden state s_i at time t and the model parameters θ. As initialization values:

$$\beta_T(i) = 1$$

for all $\beta_t(i) = P(o_{t+1}, o_{t+2}, ...o_T \mid q_t = s_i, \theta)$, where $1 \leq i \leq N$. In the induction step, the backward probability of a sequence having hidden state s_i, the probabilities at time $t+1$ and the transition probabilities are considered:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

where $t = T-1, T-2, ..., 1$ and $1 \leq i \leq N$. Likewise, the complexity of computing $\beta_t(i)$ is $O(N^2 T)$.

A couple more useful variables are introduced here. The first one is $\gamma_t(i)$, representing the probability of being in state s_i at time t, given the observation sequence and model parameters. Following the previous derivation, $\gamma_t(i)$ can also be expressed in terms of $\alpha_t(i)$ and $\beta_t(i)$:

$$\begin{aligned}\gamma_t(i) &= P(q_t = s_i \mid O, \theta) \\ &= \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \theta)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}\end{aligned}$$

The other variable $\xi_t(i,j) = P(q_t = s_i, q_{t+1} \mid O, \theta)$ is used for representing the probability of being in state s_i at time t, and being in state s_j at time $t+1$, given observation O and model parameters θ. Again ξ can be expressed by the other terms introduced earlier:

$$
\begin{aligned}
\xi_t(i,j) &= \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O \mid \theta)} \\
&= \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}
\end{aligned}
$$

And according to the definitions, summation of $\xi_t(i,j)$ over j gives $\gamma_t(i)$

$$
\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)
$$

3.2.2 A New Formation for Selective Factorial CHMM

In our SFCHMM model, we define the forward probabilities $\alpha_t^{(c)}(i)$ and $\beta_t^{(c)}(i)$. Forward probability $\alpha_t^{(c)}(i)$ is the probability at chain c, observation $o_t^{(c)}$ is observed at time t given the model parameters θ. And $\beta_t^{(c)}(i)$ is the probability at chain c that observations $o_{t+1}^{(c)}$ through $o_T^{(c)}$ are observed, given the model parameters θ and the hidden state at time t. The formal notations for α and β are as follows:

$$
\alpha_t^{(c)}(i) = P(o_1^{(1)}, o_1^{(2)}, ..., o_t^{(1)}, o_t^{(2)}, q_t^{(c)} = s_i \mid \theta, I)
$$

$$
\beta_t^{(c)}(i) = P(o_{t+1}^{(1)}, o_{t+1}^{(2)}, ..., o_T^{(1)}, o_T^{(2)} \mid q_t^{(c)} = s_i, \theta, I)
$$

where $1 \le i \le N^{(c)}$ and $c = 1, 2$. Again rather than naively compute $P(O \mid \theta)$ according to definition, calculation by induction caches the result of previous iteration and significantly improves the computation efficiency. The asymptotic complexity reduces from $O(TN^T)$ to $O(N^2T)$, where hidden N is the state space size and the length of the sequence.

In addition, We impose triggers on the transitions so that the coupled state collapse to a single chain state when there is a missing observation at either one of the chain. The triggers are generated by a simple step function H:

$$H(o_t^{(c)}) = \begin{cases} 1 & o_t^{(c)} \text{is present} \\ 0 & o_t^{(c)} \text{otherwise} \end{cases}$$

When the current observation is missing from one party of the coupled chain, the model selectively factorize or merge to the hidden state of the other chain, creating a knot in the path.

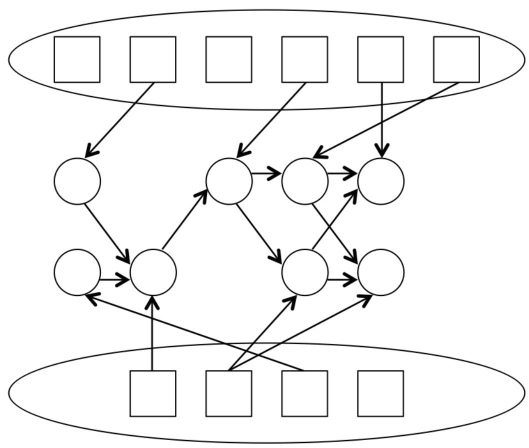

Figure 3-2: Selective Factorized CHMM

Now we will go through the derivation of forward procedure under the new formation.

- Initialization

$$\alpha_1^{(c)}(i) = \pi_i^{(c)} b_i^{(c)}(o_1)$$

- Induction Step

$$\alpha_{t+1}^{(c)}(j) = b_j^{(c)}(o_{t+1}) H(o_{t+1}^{(c)}) \sum_{c'=1,2} [I_{(c,c')} \sum_{i=1}^{N^{(c')}} (\alpha_t^{(c')}(i) \cdot a_{ij}^{(c,c')})] + (1 - H(o_{t+1}^{(c)})) \cdot w^{(c)}$$

- Termination

$$
\begin{aligned}
P(O \mid \theta) &= \prod_{c=1,2} \big(\sum_{i=1}^{N^{(c)}} \alpha_T^{(c)}(i) \big) \\
&= \prod_{c=1,2} \big(\sum_{i=1}^{N^{(c)}} \alpha_t^{(c)}(i) \cdot \beta_t^{(c)}(i) \big)
\end{aligned}
$$

More specifically in the induction step, we not only consider within-chain and cross-chain transition probabilities, but also factor in the H function. When an absence of an observation from one chain is detected, the H function causes the α and β at current iteration to be factorized with $w^{(c)}$. Weight $w^{(c)}$ is of the same dimension of $\alpha^{(c)}$. And $w^{(c)}$ can be randomly sampled from a normal distribution or be set to zeroes. If set to be zero, the model collapses to a single chain, creating a knot at the chain at that time step. In fact, the choice of $w^{(c)}$ allows us to choose from strict factorization or some random guesses to be included.

Alternatively, we can use the backward probabilities, or forward-backward probabilities for calculation:

$$
\begin{aligned}
P(O \mid \theta) &= \prod_{c=1,2} \big(\sum_{i=1}^{N^{(c)}} \alpha_T^{(c)}(i) \big) \\
&= \prod_{c=1,2} \big(\sum_{i=1}^{N^{(c)}} \alpha_t^{(c)}(i) \cdot \beta_t^{(c)}(i) \big) \\
&= \prod_{c=1,2} \big(\sum_{j=1}^{N^{(c)}} \beta_T^{(c)}(j) \pi_j b_j(o_1) \big)
\end{aligned}
$$

In addition, if we are interested in knowing the state probability at time t give the observations and model parameters, we can introduce an extra variable $\gamma_t(i)$ and calculate it with $\alpha_t(i)$ and $\beta_t(i)$:

$$
\begin{aligned}
\gamma_t(i) &= P(q_t^{(c)} = s_i \mid O, \theta) \\
&= \frac{\alpha_t(i)^{(c)} \beta_t(i)^{(c)}}{P(O \mid \theta)} \\
&= \frac{\alpha_t(i)^{(c)} \beta_t(i)^{(c)}}{\prod_{c=1,2} \left(\sum_{i=1}^{Nc} \alpha_t^{(c)}(i) \cdot \beta_t^{(c)}(i) \right)}
\end{aligned}
$$

An obvious use of the modified forward, backward and forward-backward procedures is to accommodate for our new model itself. But it can also be used as an approximation technique for standard CHMM. Suppose we have an observance sequence coming in and we would like to calculate $P(O \mid \theta)$ given an underlying model. Without a complete observation sequence, the forward and backward procedures have to fill in guesses or skip the calculation, whereas the selective factorization strategy can be used to approach the problem with computation efficiency. We will see in the experiment chapter how the estimation performs.

Chapter 4

Decoding and Parameter Estimation

4.1 Decoding Optimal Path for SFCHMM

4.1.1 Viterbi Algorithm

Viterbi [34] algorithm is widely used to find the best hidden state sequence, given both observations and the model parameters θ. In order to find the optimal hidden state path that optimize $P(Q', O \mid \theta)$, where Q' is our estimated hidden state, we first define the highest probability of a sequence up till time t and ending at hidden state s_i as $\delta_t(i)$:

$$\delta_t(i) = max_{q_1,q_2,...,q_{t-1}} P(q_1, q_2, ..., q_{t-1}, q_t = s_i, o_1, o_2, ..., o_t \mid \theta)$$

By induction, the best path merged at each possible hidden state is dependent only on the previous best state path and the conditional probabilities:

$$\delta_{t+1}(i) = max_i[\theta_t(i)a_{ij}]b_j(o_{t+1})$$

The efficiency of Viterbi lies in its dynamic programming strategy for keeping track of the best $\delta_t(i)$ at every time step t for all possible state s_i's.

An extra array $\psi_t(j)$ is used for this purpose. The steps of implementing the algorithm can be summarized as follows:

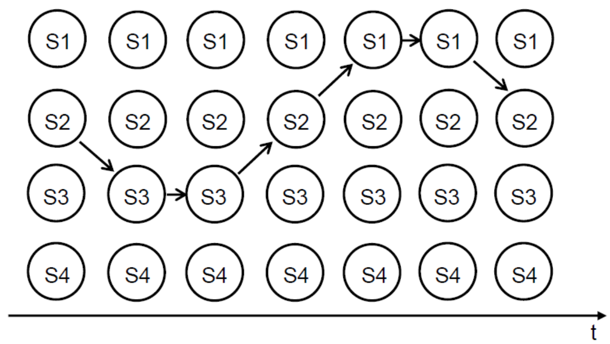

Figure 4-1: A simple Viterbi path

- Initialization

  $\delta_1(i) = \pi_i b_i(O_1)$,

  $\psi_1(i) = (0)_{N\times 1}$, where $1 \le i \le N$

- Recursion

  $\delta_t(j) = max_{i=1}^{N}[\delta_{t-1}(i)a_{ij}]b_j(o_t)$,

  $\psi_{t+1}(j) = argmax_{i=1}^{N}[\theta_{t-1}(i)a_{ij}]$, where $2 \le t \le T$ and $1 \le j \le N$

- Termination

  $P^* = max_{i=1}^{N}\delta_T(i)$,

  $q_T^* = argmax_{i=1}^{N}\delta_T(i)$

- Backtracking

  $q_t^* = \psi_{t+1}q_{t+1}^*$, where $t = T-1, T-2, ..., 1$

4.1.2 A Modified Viterbi Algorithm for SFCHMM

Now we will present the new formation of best state path finding algorithm for selective factorized CHMM. To adapt Viterbi to our model, we need to track best path according to transitional probabilities not only on a single chain but on cross-chain as well.

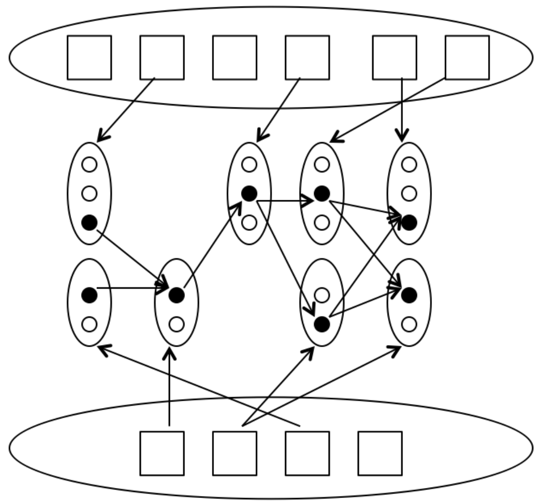

Figure 4-2: Modified Viterbi Path for Selective Factorized CHMM

We first define $\delta_t(i,j)$, a joint path probability till time t ending at hidden state $s_i^{(1)}$, $s_j^{(2)}$ given the model parameters. Another variable ψ is used keep track of current best state. Note that the new state space ${s'}_i^{(1)}$, ${s'}_j^{(2)}$ are mapped from $s_i^{(1)}$, $s_j^{(2)}$, but each has an extra null state added to account for the coupled state that selective factorization happens. The memory space of ψ is in accordance with ${s'}_i^{(1)}$, ${s'}_j^{(2)}$. The basic initialization, induction and termination scheme is similar but with the dimension expanded. In each of the iterations, we maximize the path probabilities to each of the possible state at current time step. However, the augmented Viterbi algorithm implementation also takes into consideration the coupled transitional probabilities. The final result, i.e. our optimal hidden state path is collected by backtracking ψ.

- Initialization

  $\delta_1^{(c,c')}(i) = \delta_1^{(c,c)}(i) = \pi_i^{(c)} b_i^{(c)}(o_1)$

  $\psi_1^{(c,c)}(i) = (0)_{(N^{(c)}+1)\times 1} \qquad \text{where } 1 \leq i \leq N^{(c)}$

  $\psi_1^{(c,c')}(i) = (0)_{(N^{(c')}+1)\times 1} \qquad \text{where } 1 \leq i \leq N^{(c')}$

- Recursion

  $\delta_t^{(c,c)}(j) = max_{i=1}^{N^{(c)}} \; [\delta_{t-1}(i) a_{ij}^{(c,c)}] b_j(o_t)$

  $\delta_t^{(c,c')}(j) = max_{i=1}^{N^{(c')}} \; [\delta_{t-1}(i) a_{ij}^{(c,c')}] b_j(o_t)$

$$\psi_{t+1}^{(c,c)}(j) = argmax_{i=1}^{N^{(c)}} \ [\delta_{t-1}^{(c,c)}(i) a_{ij}^{(c,c)}]$$

$$\psi_{t+1}^{(c,c')}(j) = argmax_{i=1}^{N^{(c)}} \ [\delta_{t-1}^{(c,c')}(i) a_{ij}^{(c,c')}] \qquad \text{where } 2 \leq t \leq T \text{ and } 1 \leq j \leq N$$

- Termination

$$P^{(c,c)*} = max_{i=1}^{N^{(c)}} \ \delta_T^{(c,c)}(i)$$

$$P^{(c,c')*} = max_{i=1}^{N^{(c')}} \ \delta_T^{(c,c')}(i)$$

$$q_T^{(c,c)*} = argmax_{i=1}^{N^{(c)}} \ \delta_T^{(c,c)}(i)$$

$$q_T^{(c,c')*} = argmax_{i=1}^{N^{(c')}} \ \delta_T^{(c,c')}(i)$$

- Backtracking

$$q_t^{(c,c)*} = \psi_{t+1}^{(c,c)} \cdot q_{t+1}^{(c,c)*}$$

$$q_t^{(c,c')*} = \psi_{t+1}^{(c,c')} \cdot q_{t+1}^{(c,c')*} \qquad \text{where } t = T-1, T-2, ..., 1$$

Notice that when an observation is missing from one chain, the hidden state at that time step at that chat is decoded to be null state. The forward induction and backtracking both take linear time and now the memory requirement more than double the case with single chain hidden Markov model.

4.2 Learning Model Parameters

Learning the model parameters $\theta = \{A, B, \pi\}$ is the more difficult problem related to HMM's. Most recently, there has been a breakthrough in developing analytic solver using spectral method [16]. But we will focus on the well-established iterative methods such as Baum-Welch method (a type of EM algorithm), which is guaranteed to generate locally-optimized solution.

4.2.1 EM and GEM Algorithm

EM and generalized EM (GEM) algorithm has been the most popular method for estimating HMM [12], [3]. Since the Baum-Welch is a form of EM algorithm, we will first provide a description of the algorithm listed as below:

- Estimation step

  Given observation O, parameters to estimate θ and the objective function $L(\theta; O, S)$, an auxiliary function is constructed:

$$Q(\theta, \theta') = E[L(\theta; O, S) \mid O, \theta]$$

  which is the expectation of the objective over all possible state sequences, give observations O and the current estimate of the parameter θ'. Note that in Baum-Welch the objective function is logarithmic form $L = logP(O, S \mid \theta)$

- Maximization step

  The new estimate is solve by:

$$\theta = argmaxQ(\theta; \theta')$$

  Solve for θ that maximizes $Q(\theta; \theta')$ is hard. More often, we use self-mapping transformation defined τ as $\theta_{new} = \tau(\theta')$ such that:

$$Q(\tau(\theta'); \theta) \geq Q(\theta; \theta')$$

4.2.2 Baum-Welch Algorithm

For an HMM with discrete observations, the model parameter estimations can be summarized as follows:

$$\pi_i = \gamma_1(i)$$

$$a_{ij} = \frac{\sum_{i=1}^{T-1} \xi_t(i, j)}{\sum_{i=1}^{T-1} \gamma_t(i)}$$

$$b_j(k) = \frac{\sum_{t=1}^{N} \gamma_{t_{O_t=v_k}}(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

where π_i is the expected frequency in state s_i at time $t = 1$, a_{ij} is the expected number of transition from state s_i to s_j with respect to the total number of transitions away from s_i, and $b_j(k)$ is the expected number of occurrences of state s_j observing v_k with respect to the total number of times in state s_j.

We further define an objective function L for our maximization step:

$$L(\theta, \theta') = \sum_{q \in Q} \log P(O, q \mid \theta) P(O, q \mid \theta')$$

where θ' is the estimated model parameter set and θ' is the true model parameter set. And L can be rewritten as with substitution of $P(O, q, \mid \theta)$:

$$P(O, q, \mid \theta) = \pi_i \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t)$$

$$\begin{aligned} L(\theta, \theta') = \sum_{i=1}^{N} \log \pi_i P(O, q_0 = s_i, \mid \theta') &+ \sum_{q \in Q} (\sum_{t=1}^{T} \log a_{q_{t-1}q_t}) P(O, q, \mid \theta') \\ &+ \sum_{q \in Q} (\sum_{t=1}^{T} \log b_{q_t}(o_t)) P(O, q, \mid \theta') \end{aligned}$$

The new estimates will be calculated from the three independent parts of the summation in $L(\theta, \theta')$. We can take the derivative, add a Lagrange multiplier for each, and compute the result with derivative set to zero for each part. We get the new estimations are as follows:

$$\pi_i = \frac{P(O, q_0 = s_i \mid \theta')}{P(O \mid \theta')}$$

$$a_{ij} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = s_i, q_t = s_j \mid \theta')}{\sum_{t=1}^{T} P(O, q_{t-1} = s_i \mid \theta')}$$

$$b_i(k) = \frac{\sum_{t=1}^{T} P(O, q_t \mid \theta') I(o_t = v_k)}{\sum_{t=1}^{T} P(O, q_t \mid \theta')}$$

Note that $I(o_t = v_k)$ is just an indicator that only observations equal to v_k contribute to the backward probability $b_i(k)$.

Now we will extend the Baum-Welch algorithm for SHCHMM. For selective factorized coupled hidden Markov models, the objective function is defined as:

$$L(\theta, \theta') = \sum_{q \in Q} \log P(O, q \mid \theta) P(O, q \mid \theta')$$

where θ is the set of true values of model parameters, θ' is the estimation. The constraints now become:

$$\sum \pi_i^{(c)} = 1$$

$$\sum a_{ij}^{(c,c')} = 1$$

$$\sum b_j^{(c)}(k) = 1$$

.

In each of the iteration, we solve it as a constrained optimization problem by adding the constraint multiplied by the Lagrange multiplier λ and taking the derivative.

The estimation of parameters gets updated until they converge according to a threshold. The following calculation is used in both estimation and optimization steps:

$$\pi_i^{(c)} = \frac{\pi_i^{(c)} \frac{\partial L}{\partial \pi_i^{(c)}}}{\sum_{k=1}^{N^{(c)}} \pi_k \frac{\partial L}{\partial \pi_k^{(c)}}}$$

$$a_{ij}^{(c,c')} = \frac{a_{ij}^{(c,c')} \frac{\partial L}{\partial a_{ij}^{(c,c')}}}{\sum_{k=1}^{N^{(c')}} a_{ik}^{(c,c')} \frac{\partial L}{\partial a_{ik}^{(c,c')}}}$$

$$b_j^{(c)}(k) = \frac{b_j^{(c)}(k) \frac{\partial L}{\partial b_j^{(c)}(k)}}{\sum_{v=1}^{M^{(c)}} b_j^{(c)}(v) \frac{\partial L}{\partial b_j^{(c)}(v)}}$$

It is worth noting that Baum-Welch algorithm, like all other EM algorithms, we are guaranteed to find the local optimal but not guaranteed to hit the global optimal. One way to avoid to be stuck at a bad local optimal solution is using multiple initializations.

Chapter 5

Synthetic and Empirical Experiment

The experiments consist of two parts, including synthetic simulations and an empirical test on a real world dataset. The results demonstrate the usefulness of selective factorization in three cases. One is as an approximation method of learning a discrete CHMM sequences when partial observations are missing. Another is as a method to address issues with unknown or missing observations in a continuous CHMM. And the last one is as an original learning model for human activity recognition.

5.1 Synthetic Simulation

5.1.1 Simulation Description

In the first synthetic experiment, we generate random CHMM processes with 5×5 hidden states for the coupled chains, 4 and 6 observations associated with each chain respectively. We assume equal initial probabilities for each hidden state as prior. The observational probabilities and transitional probabilities are randomly generated for every instance in our simulations.

In a graphical visualization of one simulated sequence, the sequence of the couple states are represented by two step functions, and the observations are represented by nodes of two regular polygons, the weighted connections of which indicate the occurrences of observational transitions.

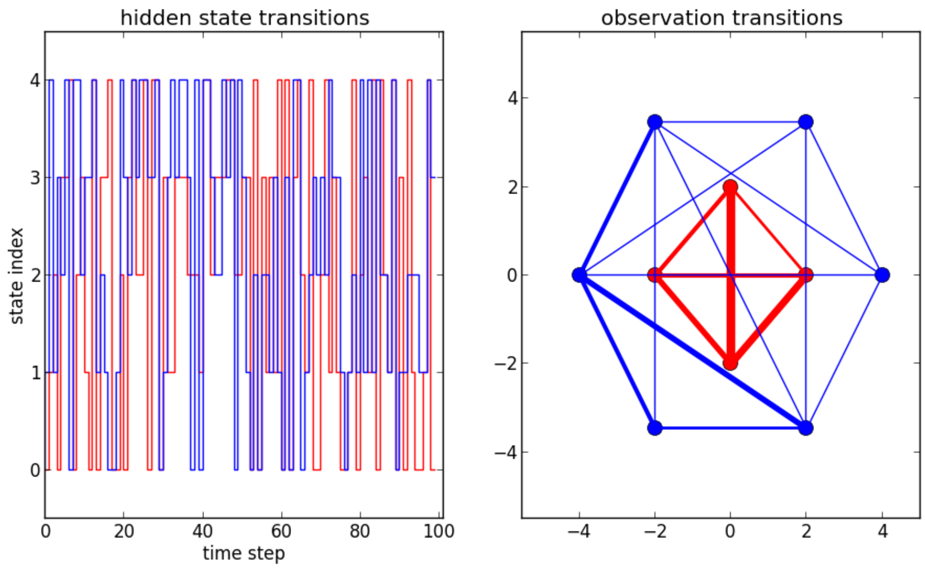

Figure 5-1: A sample of 5×5 hidden state CHMM process simulation

We mask 1% up to 50% of the observations of the simulated sequences to be unknown, keeping the non-masked sequence for base-line comparison. And then we use selective factorization in forward, backward procedures introduced earlier to estimate the $P(O' \mid \theta)$, where O' is the observations transformed after omission. We repeatedly test it on 30 independent instances of CHMM processes of length 1000 at different unknown observation percentage levels. The process is repeated for 4×4 -hidden-state and 5×5 -hidden-state CHMM as well for additional comparisons.

In the second synthetic experiment, we generate coupled models with continuous observations. The hidden state space is 3×3. For each of the chains, there are two sets of independent continuous observations. Each set is parameterized by 2-component multivariate normal distributions. We generate noises by randomly sampling from a uniform distribution. In order to compare different strategies to address unknown/missing observations, a random subset of observations generated by a CHMM is replaced by noisy observations. And the noise is randomly sampled from a uniform distribution. As a convention of solving for outlier detection of

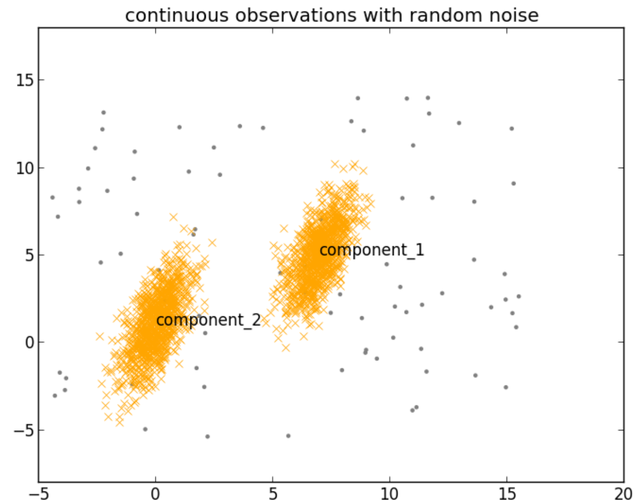

Figure 5-2: An example noisy observation simulation on one of the coupled chains.

multivariate Gaussian distributions, Mahalanobis distance, which takes into account the covariance among the variables, will be used. An observations is classified as unknown/missing when the Mahalanobis distance exceeds a threshold.

In the second experiment, some methods will consider unknown/missing observations as a special observation and recalculate the emissions accordingly whereas others won't anticipate the occurrences of noise thus using the original CHMM model parameters for estimations. As for the case of recalculating emission probabilities, we assume equal conditional probabilities of current unknown/missing state depending on hidden state $q_i^{(c)}$ to be equal for all components of the parameterized observation components.

5.1.2 Method and Result Analysis

In the first experiment, we generated 30 randomly simulated CHMM models. For each of the CHMM model, a complete observation sequence was also simulated. We used forward-backward procedure to estimate $P(O \mid \theta)$ with complete observation sequence. And then 1% up to 50% of observation sequences were randomly removed. We estimate the new $P(O \mid \theta)$'s with the forward procedure defined under selective factorized coupled hidden markov models. The result shown in the graph was aggregated over the 30 samples.

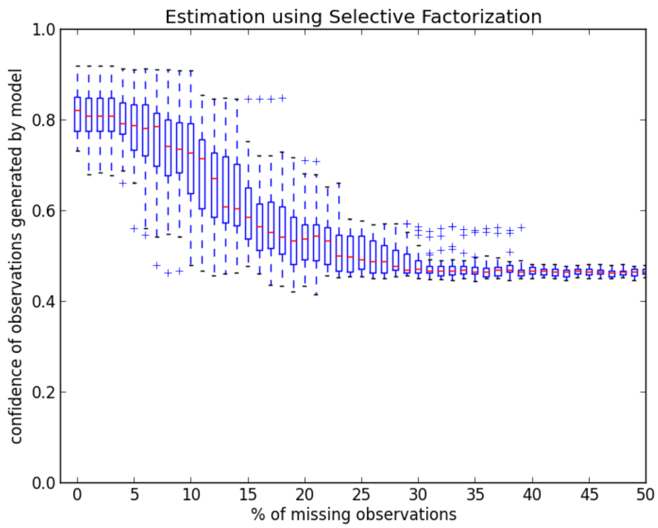

Figure 5-3: Estimation results using selective factorization on 30 samples of 5×5 hidden state CHMM with missing observations

The probability of observing the sequence given the underlying model with missing observations drops, but the negative relationship is non-linear. As we can see from the graph, if less than 7.5% of the observations are missing, selective factorization helps approximate the result well. They are close to the results generated by the normal CHMM estimation methods when there is no missing observation. However, from that point on, the drop of probabilities are steep and have greater variance, until the unstable curve flattens out at around 30%. We repeated the process for different CHMM's with different hidden state spaces and get similar results in terms of degradation trend.

To conclude the findings based on synthetic data simulation of using SFCHMM for approximating CHMM sequence, the performance of selective factorization is affected by the size of missing observations with respect to the total supposedly observed population. As long as it is below a critical value, selective factorization is a close estimation to the true model. The result becomes unstable and deteriorates beyond that point.

In the second part of synthetic experiment, we compare the performance of using

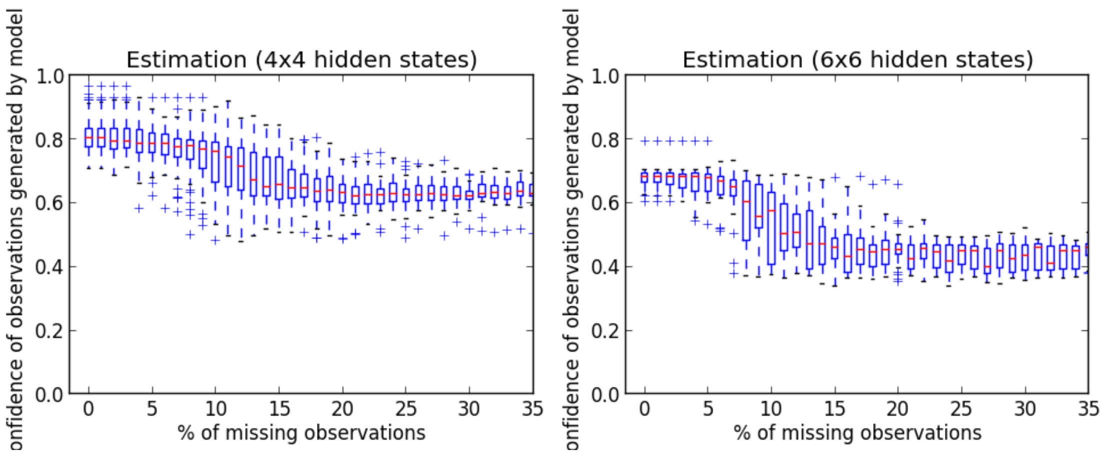

Figure 5-4: Estimation results using selective factorization on 30 samples of 4×4 and 6×6 hidden state CHMM with missing observations

selective factorization against other possible ways of dealing with unknown/missing observation for CHMM.

The result is based on simulations of 30 independent CHMM sequence of length 1000 with continuous observations and partial observations replaced by random noises sampled from a uniform distribution. There are two type of observations generated by the simulations, one type is observations identified as belonging to one of the component of continuous observations parameterized by multivariate normal distributions, the other type is outliers to any of the observation components.

We use selective factorization for continuous sequence estimation in a similar fashion as we do in the discrete case. We deploy selective factorizaton and adjust the probability dependency whenever an observation is classified as unknown/missing.

There are four other alternatives we could possibly use to estimate the sequence in addition to selective factorization.

In the first one, without modification on the existing model, we infer the most likely state from transition probabilities at the time step where the observation on one of the chain is classified as unknown/missing. In the second one, we adjust the model to include an extra state (null) associated with missing observation, where we include additional transition probabilities related to null state and the transitional probabilities of the original model are therefore diffused.

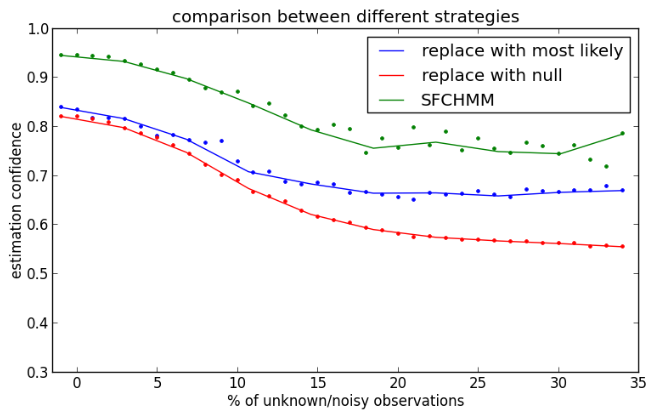

Figure 5-5: comparison between different strategies of dealing with unknown/missing observations

The first two alternatives and SFCHMM don't change the emission probabilities. And they displays a common trend of degrading estimation result after the percentage of unknown/missing observations hits below certain level, which is illustrated on drop in the curves. Among the three approaches, SFCHMM shows the best estimation result regardless to the percentage of unknown/missing observations. The replacement with most likely state yields better result than including an extra null hidden state.

In the third and fourth approaches, we will include anticipations of the occurrences of noises in the adjusted models. In the third one, we expand the emission probabilities to include missing observation probabilities proportional to the number of unknown/missing observations. And in the last model, we include both the unknown/missing emission probabilities and the transition probabilities of an extra hidden state.

Now that we've adjusted the emission probabilities, the third and fourth approaches have different trending properties with regard to percentage of unknown/missing observations. The curves stay relatively stable around the same value at different percentages.

In conclusion for the second part of synthetic experiment, when we take into account unknown/missing observations in the emission probabilities, the results don't correlate to the percentage of known/missing observations. However, for selective factorization and the two alternatives that don't adjust the emissions, they display a

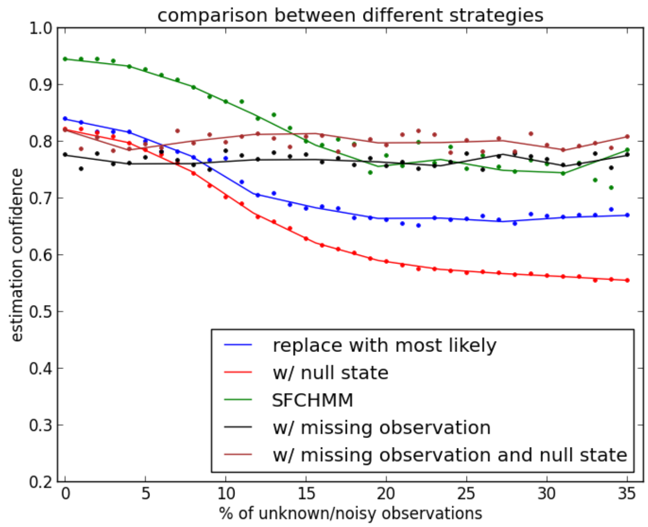

Figure 5-6: comparison between different strategies of dealing with unknown/missing observations

degrading feature after certain threshold.

As for the overall result, when the percentage of unknown/missing value exceeds around 15%, SFCHMM isn't as competitive as the fourth alternative. That is the case where we use a model that has well estimated emissions, and adjusted transition probabilities associated with unknown/missing cases. But in real applications, it is hard to anticipate the actual occurrence of unknown/missing observations and generating such a model with extra hidden state transitions and emissions is hard.

5.2 Experiment on a Real World Problem

5.2.1 Dataset Description and Measurement

To empirically test our method, we use the Opportunity human activity recognition dataset generated by researchers from Wearable Computing Laboratory ETH Zurich [32]. The dataset was acquired from 12 subjects while they were performing morning activities and included 72 sensors of 10 modalities in 15 wireless and wired networked sensor systems in the environment, objects and the body. For each subject there are 5 daily activity sessions and one drill session which has about 20 repetitions

Table 5.1: Benchmark Result of Modes of Locomotion

Classifier	Accuracy	F-measure	F-measure (without null class)
Linear discriminant analysis	0.60	0.60	0.68
Quadratic discriminant analysis	0.64	0.62	0.73
1-Nearest neighbors	0.82	0.82	0.83
3-Nearest neighbors classifier	0.83	0.83	0.84
Nearest cluster classifier	0.54	0.54	0.62

of some pre-defined actions. Data was manually labeled for modes of locomotion, gestures and high-level activities by at least two different persons.

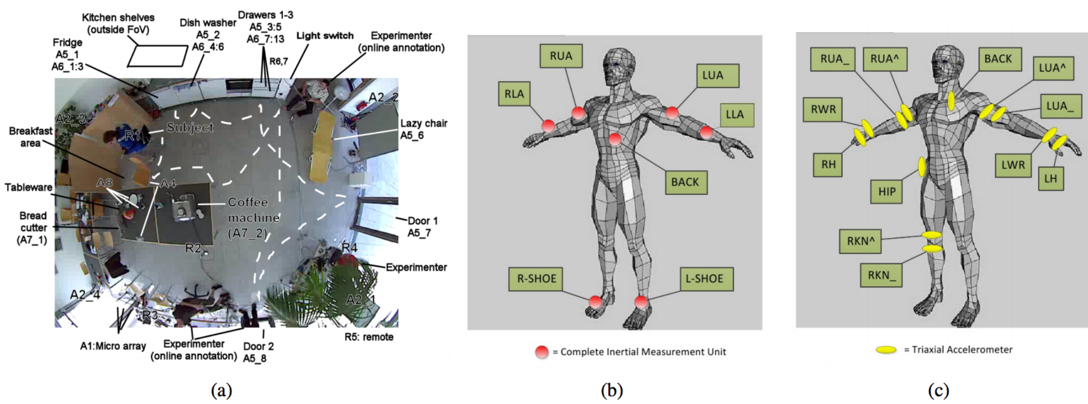

Figure 5-7: (a) Recording environment of the Opportunity dataset. (b) Location of the on-body IMU sensors. (c) Location of the bluetooth accelerometers. [32]

The Opportunity dataset is relatively ideal for the problem of investigation, for it incorporates human-object interaction as well as multi-channel sensor inputs. In addition, the benchmarking results using other methods including both discriminant models and generative models are publically available to compare against. There are two types of activity recognition tasks, the lower-level locomotion classification and the higher-level gesture classification. We are targeting at the latter.

As for the winning team of the Opportunity challenge, Hong Cao et al. proposed an integrated framework for human recognition [6], where preprocessing, balanced sampling are defined in addition to using non-sequential classifier such as Support Vector Machine (SVM) or K-Nearest Neighbors (KNN). They proved on a couple real world recognition problems that by using state-of-the-art non-sequential combined

Table 5.2: Benchmark Result of Gestures Recognition

Classifier	Accuracy	F-measure	F-measure (without null class)
Linear discriminant analysis	0.53	0.62	0.28
Quadratic discriminant analysis	0.49	0.56	0.29
1-Nearest neighbors classifier	0.83	0.83	0.52
3-Nearest neighbors classifier	0.84	0.83	0.53
Nearest cluster classifier	0.39	0.46	0.24

with pre- and post- classification techniques, this framework is able to achieve good performance for activity recognition.

As for measurement, we will use the performance measures suggested by the benchmarks, namely the weighted F-measure [8]:

$$F_1 = \sum_i 2w_i \frac{precision_i * recall_i}{precision_i + recall_i},$$

where i is the index for activity class c_i and weights i's are calculated from the proportion of class c_i samples out of all N samples, $w_i = n_i/N$. More specifically, precision is defined as $\frac{TP}{TP+FP}$ and recall is defined as $\frac{TP}{TP+FN}$ and can be calculated from the confusion matrix. Due to ambiguity of onset and offset of an action within a continuous activity sequence, misalignment measures are also considered.

5.2.2 Method and Result Analysis

Our data preprocessing includes removing rows that include only null values and discretizing input features. Then a selectively factorized CHMM is generated for each of the 5 daily activity categories, excluding the null activity class. We used locomotion and middle-level upper-arm gesture labels as our hidden states for the coupled chains. The associated observations are encoded by lower-level gestures and object labels. One observation after constructing the models is that in activity categories where the upper-body is mostly idle and the human object is not actively interacting with objects, there are more factorized states. The opposite is true for activities that engage both human body and objects, for instance drinking from coffee cups, eating salami, and closing dish washers.

Table 5.3: Summary of Training Result

high-level activity label	% of samples	% of selective factorized states
relaxing	7.7%	46.2%
clean_up	13.8%	34.2%
sandwich_time	29.3%	26.7%
early_morning	29.6%	34.2%
coffee_time	19.6%	18.2%

We set aside about two-third of the total dataset as our training data to calculate model parameter estimations and then apply on the rest for gesture recognition test. The test data is first segmented to smaller sequences according to the high-level activity categories. We use the augmented Viterbi algorithm modified based on our selective factorized CHMM model to learn the optimal hidden state path, which consists our gesture labels.

With discretization, the consecutive samples with same set of observations and state labels are combined to a single one and thus the number of sequence doesn't correspond to the exact length of the original dataset. The overall weighted F-measure excluding the null class is 0.69. The result is comparable to other top models such as KNN, which is between 0.53 and 0.58 and mixture of SVM and KNN, which is between 0.72 and 0.80, depending on the subset the model is tested [32]. Although the result doesn't stand out as the best performer in terms of F-measure, it has an advantage of preserving temporal information, which will be very useful in more complicated problems where transitions between gestures and activities need to be investigated.

Table 5.4: Summary of Test Result

gesture (# of samples tested)	Individual F-measure
Open Door 1 (25)	0.62
Open Door 2 (23)	0.55
Close Door 1 (25)	0.64
Close Door 2 (22)	0.56
Open Fridge (41)	0.78
Close Fridge (41)	0.79
Open Dishwasher (20)	0.68
Close Dishwasher (20)	0.62
Open Drawer 1 (17)	0.66
Close Drawer 1 (17)	0.63
Open Drawer 2 (26)	0.65
Close Drawer 2 (26)	0.62
Open Drawer 3 (32)	0.58
Close Drawer 3 (32)	0.63
Clean Table (10)	0.86
Drink from Cup (43)	0.88
Toggle Switch (21)	0.91

Chapter 6

Conclusion and Future Work

In conclusion, we introduce selective factorization as a complimentary technique for coupled hidden markov models and demonstrate how it can be incorporated to CHMM to solve inference and prediction problems. Both the theoretical derivations and experimental investigation are presented.

In cases where there are observational features absent from a standard coupled hidden markov process, we can apply selective factorization as an approximation method. We've shown through our synthetic experiment that the result is close to what one would obtain with the complete observations, as long as the portion of missing features is below certain limit. We've also compared selective factorization against other methods used to dealing with unknown or missing observations in CHMM. Selective factorized coupled hidden markov model can also be used to solve activity recognition problems that involve human-object interaction and noisy, multi-channel sensor inputs. The model is able to both produce competitive result and preserve important temporal information. It would be interesting to see our method to be applied to more real-world datasets to see how it performs under different conditions.

Currently the factorization trigger function H is defined according to the presence of observations. However, if H can be generalized and embed a mechanism to select observation instance for selective factorization, this method be more widely used. The other aspect of future investigation is to develop unsupervised learning that can automatically generate an optimal selective factorized coupled Markov models. And the learning model should also be able to address the complexity due to large state space and factorization conditions.

Bibliography

[1] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer, 2011.

[2] Yoshua Bengio. Markovian models for sequential data. *Neural computing surveys*, 2:129–162, 1999.

[3] Yoshua Bengio and Paolo Frasconi. Input-output hmms for sequence processing. *Neural Networks, IEEE Transactions on*, 7(5):1231–1249, 1996.

[4] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[5] Matthew Brand. Coupled hidden markov models for modeling interacting processes, 1997.

[6] Hong Cao, Minh Nhut Nguyen, Clifton Phua, Shonali Krishnaswamy, and Xiao-Li Li. An integrated framework for human activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 621–622. ACM, 2012.

[7] Ozgür Cetin, Mari Ostendorf, and Gary D Bernard. Multirate coupled hidden markov models and their application to machining tool-wear classification. *Signal Processing, IEEE Transactions on*, 55(6):2885–2896, 2007.

[8] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

[9] Stephen M Chu and Thomas S Huang. Audio-visual speech modeling using coupled hidden markov models. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–2009. IEEE, 2002.

[10] Youtian Du, Feng Chen, Wenli Xu, and Yongbin Li. Recognizing interaction activities using dynamic bayesian network. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 618–621. IEEE, 2006.

[11] Thi V Duong, Hung Hai Bui, Dinh Q Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845. IEEE, 2005.

[12] Yariv Ephraim, Amir Dembo, and L Rabiner. A minimum discrimination information approach for hidden markov modeling. *Information Theory, IEEE Transactions on*, 35(5):1001–1013, 1989.

[13] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[14] David Hogg. Model-based vision: a program to see a walking person. *Image and vision computing*, 1(1):5–20, 1983.

[15] Somboon Hongeng and Ramakant Nevatia. Large-scale event detection using semi-hidden markov models. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1455–1462. IEEE, 2003.

[16] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

[17] Weiming Hu, Dan Xie, Tieniu Tan, and Steve Maybank. Learning activity patterns using fuzzy self-organizing neural network. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(3):1618–1626, 2004.

[18] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. 2010.

[19] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[20] Xiaoxing Liu, Yibao Zhao, Xiaobo Pi, Luhong Liang, and Ara V Nefian. Audio-visual continuous speech recognition using a coupled hidden markov model. In *INTERSPEECH*, 2002.

[21] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, pages 591–598, 2000.

[22] Christian Micheloni, Lauro Snidaro, and Gian Luca Foresti. Exploiting temporal statistics for events analysis and understanding. *Image and Vision Computing*, 27(10):1459–1469, 2009.

[23] Kevin P Murphy. Hidden semi-markov models (hsmms). *unpublished notes*, 2002.

[24] Pradeep Natarajan and Ramakant Nevatia. Coupled hidden semi markov models for activity recognition. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 10–10. IEEE, 2007.

[25] Pradeep Natarajan and Ramakant Nevatia. Hierarchical multi-channel hidden semi markov graphical models for activity recognition. *Computer Vision and Image Understanding*, 117(10):1329–1344, 2013.

[26] Ara V Nefian, Luhong Liang, Xiaobo Pi, Liu Xiaoxiang, Crusoe Mao, and Kevin Murphy. A coupled hmm for audio-visual speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–2013. IEEE, 2002.

[27] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Computer Vision–ECCV 2010*, pages 392–405. Springer, 2010.

[28] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.

[29] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[30] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[31] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[32] Hesam Sagha, Sundara Tejaswi Digumarti, J del R Millan, Ricardo Chavarriaga, Alberto Calatroni, Daniel Roggen, and G Troster. Benchmarking classification techniques using the opportunity human activity dataset. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 36–40. IEEE, 2011.

[33] Jan-Willem van de Meent, Jonathan E Bronson, Frank Wood, Ruben L Gonzalez Jr, and Chris H Wiggins. Hierarchically-coupled hidden markov models for learning kinetic rates from single-molecule data. *arXiv preprint arXiv:1305.3640*, 2013.

[34] Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[35] Ming Yang, Fengjun Lv, Wei Xu, Kai Yu, and Yihong Gong. Human action detection by boosting efficient motion features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 522–529. IEEE, 2009.

[36] Shi Zhong and Joydeep Ghosh. A new formulation of coupled hidden markov models. *Department of Electrical and Computer Engineering, The University of Texas at Austin*, 2001.