Chapter 10

UNSUPERVISED MINING OF STATISTICAL TEMPORAL STRUCTURES IN VIDEO

Lexing Xie, Shih-Fu Chang
Department of Electrical Engineering, Columbia University, New York, NY
{xlx,sfchang}@ee.columbia.edu

Ajay Divakaran and Huifang Sun
Mitsubishi Electric Research Labs, Cambridge, MA
{ajayd,hsun}@merl.com

Abstract In this chapter we present algorithms for unsupervised mining of structures in video using multi-scale statistical models. Video structure are repetitive segments in a video stream with consistent statistical characteristics. Such structures can often be interpreted in relation to distinctive semantics, particularly in structured domains like sports. While much work in the literature explores the link between the observations and the semantics using supervised learning, we propose unsupervised structure mining algorithms that aim at alleviating the burden of labelling and training, as well as providing a scalable solution for generalizing video indexing techniques to heterogeneous content collections such as surveillance and consumer video. Existing unsupervised video structuring work primarily uses clustering techniques, while the rich statistical characteristics in the temporal dimension at different granularities remain unexplored. Automatically identifying structures from an unknown domain poses significant challenges when domain knowledge is not explicitly present to assist algorithm design, model selection, and feature selection. In this work we model multi-level statistical structures with hierarchical hidden Markov models based on a multi-level Markov dependency assumption. The parameters of the model are efficiently estimated using the EM algorithm. We have also developed a model structure learning algorithm that uses stochastic sampling techniques to find the optimal model structure, and a feature selection algorithm that automatically finds compact relevant feature sets using hybrid wrapper-filter methods. When tested on sports videos, the unsupervised learning scheme achieves very promising results: (1) The automatically selected

feature set for soccer and baseball videos matches sets that are manually selected with domain knowledge. (2) The system automatically discovers high-level structures that match the semantic events in the video. (3) The system achieves better accuracy in detecting semantic events in unlabelled soccer videos than a competing supervised approach designed and trained with domain knowledge.

Keywords: Multimedia mining, structure discovery, unsupervised learning, video indexing, statistical learning, model selection, automatic feature selection, hierarchical hidden Markov model (HHMM), hidden Markov model (HMM), Markov chain Monte-Carlo (MCMC), dynamic Bayesian network (DBN), Bayesian Information Criteria (BIC), maximum likelihood (ML), expectation maximization (EM).

Introduction

In this chapter, we present algorithms for jointly discovering statistical structures, using the appropriate model complexity, and finding informative low-level features from video in an unsupervised setting. These techniques address the challenges of automatically mining salient structures and patterns that exist in video streams from many practical domains. Effective solutions to video indexing require detection and recognition of *structure* and *event* in the video, where *structure* represents the syntactic level composition of the video content, and *event* represents the occurrences of certain semantic concepts. In specific domains, high-level syntactic structures may correspond well to distinctive semantic events. Our focus is on temporal structures, which is defined as the repetitive segments in a time sequence that possess consistent deterministic or statistical characteristics. This definition is general to various domains, and it is applicable at multiple levels of abstraction. At the lowest level for example, structure can be the frequent triples of symbols in a DNA sequence, or the repeating color schemes in a video; at the mid-level, the seasonal trends in web traffics, or the canonical camera movements in films; and at a higher level, the genetic functional regions in DNA sequences, or the game-specific temporal state transitions in sports video. Automatic detection of structures will help locate semantic events from low-level observations, and facilitate summarization and navigation of the content.

The structure discovery problem. The problem of identifying structure consists of two parts: finding a description of the structure (a.k.a *the model*), and locating segments that match the description. There are many successful cases where these two tasks are performed in separate steps. The former is usually referred to as *training*, while the

latter, *classification* or *segmentation*. Among various possible models, hidden Markov model (HMM) [Rabiner, 1989] is a discrete state-space stochastic model with efficient learning algorithms that work well for temporally correlated data streams. HMM has been successfully applied to many different domains such as speech recognition, handwriting recognition, motion analysis, or genome sequence analysis. For video analysis in particular, different genres in TV programs have been distinguished with HMMs trained for each genre in [Wang et al., 2000], and the high-level structure of soccer games (e.g. play versus break) was also delineated with a pool of HMMs trained for each category in [Xie et al., 2002b].

The structure detection methods above fall in the conventional category of supervised learning - the algorithm designers manually identify important structures, collect labelled data for training, and apply supervised learning tools to learn the classifiers. This methodology works for domain-specific problems at a small scale, yet it cannot be readily extended to diverse new domains at a large scale. In this chapter, we propose a new paradigm that uses fully unsupervised statistical techniques and aims at automatic discovery of salient structures and simultaneously recognizing such structures in unlabelled data without prior domain knowledge. Domain knowledge, if available, can be used to assign semantic meanings to the discovered structures in a post-processing stage. Although unsupervised clustering techniques date back to several decades ago [Jain et al., 1999], most of the data sets were treated as independent samples, while the temporal correlation between samples were largely unexplored. Classical time series analysis techniques has been widely used in many domains such as financial data and web stat analysis [Iyengar et al., 1999], where the problem of identifying seasonality reduces to the problem of parameter estimation with a known order ARMA model, where the order is determined with prior statistical tests. Yet this model does not readily adapt to domains with dynamically changing model characteristics, as is often the case with video. New statistical methods such as Monte Carlo sampling have also appeared in genome sequence analysis [Lawrence et al., 1993], where unknown short motifs were recovered by finding the best alignment among all protein sequences using Gibbs sampling techniques on a multinomial model, yet independence among amino acids in adjacent positions is still assumed. Only a few instances have been explored for video. Clustering techniques are used on the key frames of shots [Yeung and Yeo, 1996] or the principal components of color histogram of image frames [Sahouria and Zakhor, 1999], to detect the story units or scenes in the video, yet the temporal dependency of video has not been fully explored. In the inde-

pendent work in [Clarkson and Pentland, 1999; Naphade and Huang, 2002], several left-to-right HMMs were concatenated to identify temporally evolving events in ambulatory videos captured by wearable devices or in films. In the former, the resulting clusters correspond to different locations such as the lab or a restaurant; while in the latter, some of the clusters correspond to recurrent events such as explosion.

Unsupervised learning of statistical structures also involve automatic selection of features extracted from the audio-visual stream. The computational front end in many real-world scenarios extracts a large pool of observations (i.e. features) from the stream, and in the absence of expert knowledge, picking a subset of relevant and compact features becomes a bottleneck. Automatically identifying informative features, if done, will improve both the learning quality and computation efficiency. Prior work in feature selection for supervised learning mainly divides into *filter* and *wrapper* methods according to whether or not the classifier is in-the-loop [Koller and Sahami, 1996]. Many directions of existing work address the supervised learning scenario, and evaluate the fitness of a feature with regard to its information gain against training labels (*filter*) or the quality of learned classifiers (*wrapper*). For unsupervised learning on spatial data (i.e. assuming temporally adjacent samples are independent), [Xing and Karp, 2001] developed a method that iterated between cluster assignment and filter/wrapper methods under the scenario that the number of clusters is known; [Dy and Brodley, 2000] used scatter separability and maximum likelihood (ML) criteria to evaluate fitness of features. To the best of our knowledge, no prior work has been reported for our particular problem of interest: unsupervised learning on temporally dependent sequences with unknown cluster size.

Characteristics of Video Structure. Our main attention in this chapter is on the particular domain of video (i.e. audio-visual streams), where the structures have the following properties from our observations: (1) Video structure is in a discrete state-space, since we humans understand video in terms of concepts, and we assume there exist a small set of concepts in a given domain; (2) The features, i.e. observations from data,s are stochastic, as segments of video seldom have exactly the same raw features even if they are conceptually similar; (3) The sequence is highly correlated in time, since the videos are sampled at a rate much higher than that of the changes in the scene.

In this chapter, several terms are used without explicit distinction in referring to the *video structures*, despite the differences in their original meanings: by *structure* we emphasize the statistical characteristics in raw features. Given specific domains, such statistic structures often

correspond to *events*, which represent occurrences of objects, or changes of the objects or the current scene.

In particular, we will focus on *dense* structures in this chapter. By *dense* we refer to the cases where constituent structures can be modelled as a common parametric class, and representing their alternation would be sufficient for describing the whole data stream. In this case, there is no need for an explicit *background* class, which may or may not be of the same parametric form, to delineate *sparse* events from the majority of the background.

Based on the observations above, we model stochastic observations in a temporally correlated discrete state space, and adopt a few weak assumptions to facilitate efficient computation. We assume that within each *event*, states are discrete and Markov, and observations are associated with states under a fixed parametric form, usually Gaussian. Such assumptions are justified based on the satisfactory results from previous work using supervised HMM to classify video events or genre [Wang et al., 2000; Xie et al., 2002b]. We also model the transitions of events as a Markov chain at a higher level; this simplification will enable efficient computation at a minor cost of modelling power.

Our approach. In this chapter, we model the temporal dependencies in video and the generic structure of events in a unified statistical framework. Adopting the multi-level Markov dependency assumptions above for computational efficiency in modelling temporally structures, we model the recurring events in each video as HMMs, and the higher-level transitions between these events as another level of Markov chain. This hierarchy of HMMs forms a Hierarchical Hidden Markov Model (HHMM); its hidden state inference and parameter estimation can be efficiently learned in $O(T)$ using the expectation-maximization (EM) algorithm. This framework is general in that it is scalable to events of different complexity; yet it is also flexible in that prior domain knowledge can be incorporated in terms of state connectivity, number of levels of Markov chains, and the time scale of the states.

We have also developed algorithms to address model selection and feature selection problems that are necessary in unsupervised settings when domain knowledge is not used. Bayesian learning techniques are used to learn the model complexity automatically, where the search over model space is done with reverse-jump Markov chain Monte Carlo, and Bayesian Information Criteria (BIC) is used as model posterior. We use an iterative filter-wrapper methods for feature selection, where the wrapper step partitions the feature pool into consistent groups that agree with each other with a mutual information gain criterion, and the filter step

eliminates redundant dimensions in each group by finding an approximate Markov blanket, and finally the resulting groups are ranked with modified BIC with respect to their *a posteriori* fitness. The approach is elegant in that maximum likelihood parameter estimation, model and feature selection, structure decoding, and content segmentation are done in a single unified process.

Evaluation on real video data shows very promising results. We tested the algorithm on multiple sports videos, and our unsupervised approach automatically discovers the high-level structures, namely, *plays* and *breaks* in soccer and baseball. The feature selection method also automatically discovered a compact relevant feature set, which matched the features manually selected using domain knowledge. The new unsupervised method discovers the statistical descriptions of high-level structure from unlabelled video, yet it achieves even slightly higher accuracy (75.7% and 75.2% for unsupervised vs. 75.0% for supervised, Section 10.5.1) when compared to our previous results using supervised classification with domain knowledge and similar HMM models. We have also compared the proposed HHMM model with left-to-right models with single entry/exit states as in [Clarkson and Pentland, 1999; Naphade and Huang, 2002], and the average accuracy of the HHMM is 2.3% better than that of the constrained models. So the additional hierarchical structure imposed by HHMM over a more constrained model introduces more modelling power on our test domain.

The rest of this chapter is organized as follows: Section 10.1 presents the structure and semantics of the HHMM model; Section 10.2 presents the inference and parameter learning algorithms for HHMM; Section 10.3 presents algorithms for learning HHMM strucutre; Section 10.4 presents our feature selection algorithm for unsupervised learning over temporal sequences; Section 10.5 evaluates the results of learning with HHMM on sports video data; Section 10.6 summarizes the work and discusses open issues.

10.1 Hierarchical hidden Markov models

Based on the two-level Markov setup described above, we use a two-level hierarchical hidden Markov model to model structures in video. In this model, the higher-level structure elements usually correspond to semantic events, while the lower-level states represent variations that can occur within the same event, and these lower-level states in turn produce the observations, i.e., measurements taken from the raw video, with mixture-of-Gaussian distribution. Note the HHMM model is a special case of Dynamic Bayesian Network (DBN); also note the model

can be easily extended to more than two levels, and feature distribution is not constrained to mixture-of-Gaussians. In the sections that follow, we will present algorithms that address the inference, parameter learning, and structure learning problems for general D-level HHMMs.

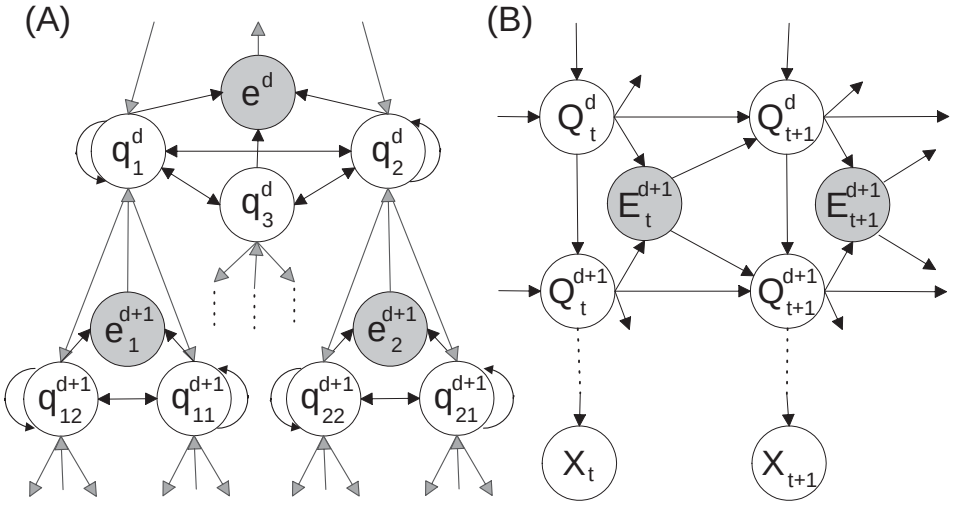

Figure 10.1. Graphical HHMM representation at level d and $d+1$ (A) Tree-structured representation; (B) DBN representations, with observations X_t drawn at the bottom. Uppercase letters denote the states as random variables in time t, lowercase letters denote the state-space of HHMM, i.e. values these random variables can take in any time slice. Shaded nodes are auxiliary *exit* nodes that turn on the transition at a higher level - a state at level d is not allowed to change unless the exiting states in the levels below are *on* ($E^{d+1} = 1$).

10.1.1 Structure of HHMM

Hierarchical hidden Markov modeling was first introduced in [Fine et al., 1998] as a natural generalization to HMM with a hierarchical control structure. As shown in Figure 10.1(A), every higher-level state symbol corresponds to a stream of symbols produced by a lower-level sub-HMM; a transition at the high level model is invoked only when the lower-level model enters an *exit* state (shaded nodes in Figure 10.1(A)); observations are only produced at the lowest level states.

This bottom-up structure is general in that it includes several other hierarchical schemes as special cases. Examples include the stacking of left-right HMMs [Clarkson and Pentland, 1999; Naphade and Huang, 2002], where across-level transitions can only happen at the first or the last state of a lower-level model; or the discrete counterpart of the jump Markov model [Doucet and Andrieu, 2001] with top-down(rather than

bottom-up) control structure, where the level-transition probabilities are identical for each state that belongs to the same *parent* state at a higher level.

Prior applications of HHMM falls into three categories: (1) Supervised learning where manually segmented training data is available, hence each sub-HMM is learned separately on the segmented sub-sequences, and cross-level transitions are learned using the transition statistics across the subsequences. Examples include extron/intron recognition in DNA sequences [Hu et al., 2000], action recognition [Ivanov and Bobick, 2000]; more examples summarized in [Murphy, 2001] fall into this category. (2) Unsupervised learning, where segmented data at any level are not available for training, and parameters of different levels are jointly learned; (3) A mixture of the above, where state labels at the high level is given (with or without sub-model boundary), yet parameters still need to be estimated across several levels. Few instances of (2) can be found in the literature, while examples of (3), as a combination of (1) and (2), abound: the celebrated application of speech recognition systems with word-level annotation [The HTK Team, 2000], text parsing and handwriting recognition [Fine et al., 1998].

10.1.2 Complexity of Inferencing and Learning with HHMM

Fine et. al. have shown that multi-level hidden state inference with HHMM can be done in $O(T^3)$ by looping over all possible lengths of subsequences generated by each Markov model at each level, where T is the sequence length [Fine et al., 1998]. This algorithm is not optimal, however, an $O(T)$ algorithm has later been shown in [Murphy and Paskin, 2001] with an equivalent DBN representation by unrolling the multi-level states in time (Figure 10.1(B)). In this DBN representation, the hidden states Q_t^d at each level $d = 1, \ldots D$, the observation sequence X_t, and the auxiliary *level-exiting* variables E_t^d completely specify the state of the model at time t. Note E_t^d can be turned on only if all lower levels of $E_T^{d+1:D}$ are on. The inference scheme used in [Murphy and Paskin, 2001] is the generic junction tree algorithm for DBNs, and the empirical complexity is $O(DT \cdot |Q|^{1.5D})$,[1] where D is the number of levels in the hierarchy, and $|Q|$ is the maximum number of distinct discrete values of any variable $Q_t^d, d = 1, \ldots, D$.

For simplicity, we use a generalized forward-backward algorithm for hidden state inference, and a generalized EM algorithm for parameter

[1] More accurately, $O(DT \cdot |Q|^{\lceil 1.5D \rceil} 2^{\lceil 0.5D \rceil})$

estimation based on the forward-backward iterations. The algorithms is outlined in Section 10.2, and details can be found in [Xie et al., 2002a]. Note the complexity of this algorithm is $O(DT \cdot |Q|^{2D})$, with a similar running time as [Murphy and Paskin, 2001] for small D and modest Q.

10.2 Learning HHMM parameters with EM

In this section, we define notations to represent the states and parameter set of an HHMM, followed by a brief overview on deriving the EM algorithm for HHMMs. Details of the forward-backward algorithm for multi-level hidden state inference, and the EM update algorithms for parameter estimation are found in [Xie et al., 2002a]. The scope of the EM algorithm is the basic parameter estimation; we will assume that the size of the model is given, and the model is learned over a pre-defined feature set. These two assumptions are relaxed using the proposed model selection algorithms described in Section 10.3, and feature selection criteria in Section 10.4.

10.2.1 Representing an HHMM

Denote the maximum state-space size of any sub-HMM as N, we use the *bar notation* (Equation10.1) to write the entire configuration of the hierarchical states from the top (level 1) to the bottom (level D) with a N-ary D-digit integer, with the lowest-level states at the least significant digit:

$$k^{(D)} = q_{1:D} = \overline{(q_1 q_2 \dots q_D)} = \sum_{i=1}^{D} q_i \cdot N^{D-i} \tag{10.1}$$

Here $1 \le q_i \le N; i = 1, \dots, D$. We drop the superscript of k where there is no confusion, the whole parameter set Θ of an HHMM then consists of (1) Markov chain parameters λ^d in level d indexed by the state configuration $k^{(d-1)}$, i.e., transition probabilities A_k^d, prior probabilities π_k^d, and exiting probabilities from the current level e_k^d; (2) emission parameters B that specify the distribution of observations conditioned on the state configuration, i.e., the means μ_k and covariances σ_k when emission distributions are Gaussian.

$$\begin{aligned} \Theta &= (\bigcup_{d=1}^{D} \{\lambda^d\}) \bigcup \{B\} \\ &= (\bigcup_{d=1}^{D} \bigcup_{i=1}^{N^{d-1}} \{A_i^d, \pi_i^d, e_i^d\}) \bigcup (\bigcup_{i=1}^{N^D} \{\mu_i, \sigma_i\}) \end{aligned} \tag{10.2}$$

10.2.2 Overview of the EM algorithm

Denote Θ the old parameter set, $\hat{\Theta}$ the new (updated) parameter set, then maximizing the data likelihood L is equivalent to iteratively maximizing the expected value of the complete-data log-likelihood function $\Omega(\cdot, \Theta)$ as in Equation (10.3), for the observation sequence $X_{1:T}$ and the D-level hidden state sequence $Q_{1:T}$, according to the general EM presented in [Dempster et al., 1977]. Here we adopt the Matlab-like notation to write a temporal sequence of length T as $(\cdot)_{1:T}$, and its element at time t is simply $(\cdot)_t$.

$$\begin{aligned} \Omega(\hat{\Theta}, \Theta) &= E[\log(P(Q_{1:T}, X_{1:T}|\hat{\Theta}))|X_{1:T}, \Theta] \quad (10.3) \\ &= \sum_{Q_{1:T}} P(Q_{1:T}|X_{1:T}, \Theta) \log(P(Q_{1:T}, X_{1:T}|\hat{\Theta})) \\ &= L^{-1} \sum_{Q_{1:T}} P(Q_{1:T}, X_{1:T}|\Theta) \log(P(Q_{1:T}, X_{1:T}|\hat{\Theta})) \quad (10.4) \end{aligned}$$

Generally speaking, the "E" step evaluates this expectation based on the current parameter set Θ, and the "M" step finds the value of $\hat{\Theta}$ that maximizes this expectation. Special care must be taken in choosing a proper hidden state space for the "M" step of (10.4) to have a closed-form solution. Since all the unknowns lie inside the $\log(\cdot)$, it can be easily seen that if the complete-data probability $P(Q_{1:T}, X_{1:T}|\hat{\Theta})$ takes the form of product-of-unknown-parameters, we would get summation-of-individual-parameters in $\Omega(\hat{\Theta}, \Theta)$, hence each unknown can be solved separately in maximization and a closed-form solution is possible.

10.3 Bayesian model adaptation

Parameter learning for HHMM using EM is known to converge to a local maximum of the data likelihood since EM is an hill-climbing algorithm, and it is also known that searching for a global maximum in the likelihood landscape is intractable. Moreover, this optimization for data likelihood is only carried out over a predefined model structure, and in order to enable the comparison and search over a set of model structures, we will need not only a new optimality criterion, but also an alternative search strategy since exhausting all model topologies is super-exponential in complexity.

In this work, we adopt randomized search strategies to address the intractability problem on the parameter and model structure space; and the optimality criterion is generalized to maximum posterior from maximum likelihood, thus incorporating *Bayesian* prior belief on the model

structure. Specifically, we use a Markov chain Monte Carlo (MCMC) method to maximize Bayesian information criterion (BIC) [Schwarz, 1978], and the motivation and basics structure of this algorithm are presented in the following subsections.

We are aware that alternatives for structure learning exist, such as the deterministic parameter trimming algorithm with entropy prior [Brand, 1999], which ensures the monotonic increasing of model priors throughout the trimming process. But we would have to start with a sufficiently large model in order to apply this trimming algorithm, which is not preferable for computational complexity purposes, and also impossible if we do not know a bound of the model complexity beforehand.

10.3.1 Overview of MCMC

MCMC is a class of algorithms that can solve high-dimensional optimization problems, which is seeing much recent success in Bayesian learning of statistical models [Andrieu et al., 2003]. In general, MCMC for Bayesian learning iterates between two steps: (1) The proposal step gives a new model sampled from certain *proposal distributions*, which depend on the current model, and statistics of the data; (2) The decision step computes an acceptance probability α based on the *fitness* of the proposed new model using model posterior and proposal strategies, and then this proposal is *accepted* or *rejected* with probability α.

MCMC will converge to the global optimum *in probability* if certain constraints [Andrieu et al., 2003] are satisfied for the proposal distributions, yet the speed of convergence largely depends on the *goodness* of the proposals. In addition to parameters learning, model selection can also be addressed in the same framework with reverse-jump MCMC (RJ-MCMC) [Green, 1995], by constructing reversible moves between parameter spaces of different dimensions. In particular, [Andrieu et al., 2001] applied RJ-MCMC to the learning of radial basis function (RBF) neural networks by introducing birth-death and split-merge moves to the RBF kernels. This is similar to our case of learning a variable number of Gaussians in the feature space that correspond to the emission probabilities.

In this work, we deployed a MCMC scheme to learn the optimal state-space of an HHMM model. We use a mixture of the EM and MCMC algorithms, where the model parameters are updated using EM, and model structure learning uses MCMC. We choose this hybrid algorithm in place of full Monte Carlo update of the parameter set and the model, since MCMC update of parameters will take much longer than EM, and the convergence behavior does not seem to suffer in practice.

10.3.2 MCMC for HHMM

Model adaptation for HHMM involves moves similar to [Andrieu et al., 2003] since many changes in the state space involve changing the number of Gaussian kernels that associate states in the lowest level with observations. We included four general types of movement in the state-space, as can be illustrated form the tree-structured representation of the HHMM in figure 10.1(a): (1) *EM*, regular parameter update without changing the state space size. (2) *Split(d)*, to split a state at level d. This is done by randomly partitioning the direct children (when there are more than one) of a state at level d into two sets, assigning one set to its original parent, the other set to a newly generated parent state at level d; when split happens at the lowest level(i.e. $d = D$), we split the Gaussian kernel of the original observation probabilities by perturbing the mean. (3) *Merge(d)*, to merge two states at level d into one, by collapsing their children into one set and decreasing the number of nodes at level d by one. (4) *Swap(d)*, to swap the parents of two states at level d, whose parent nodes at level $d-1$ was not originally the same. This special new move is needed for HHMM, since its multi-level structure is non-homogeneous within the same size of overall state-space. Note we are not including birth/death moves for simplicity, since these moves can be reached with multiple moves of split/merge.

Model adaptation for HHMMs is choreographed as follows:

1 Initialize the model Θ_0 from data.

2 At iteration i, based on the current model Θ_i, compute a probability profile $P_{\Theta_i} = [p_{em}, p_{sp}(1:D), p_{me}(1:D), p_{sw}(1:D)]$ according to Equations (10.A.1)-(10.A.4) in the appendix, then propose a move among the types $\{$*EM, Split(d), Merge(d), Swap(d)*$|d = 1, \ldots, D\}$

3 Update the model structure and the parameter set by appropriate action on selected states and their children states, as described in the appendix;

4 Evaluate the acceptance ratio r_i for different types of moves according to Equations (10.A.7)–(10.A.11) in the appendix, this ratio takes into account model posterior, computed with BIC (Equation 10.5), and alignment terms that compensate for the fact that the spaces between which we are evaluating the ratio are of unequal size. Denote the acceptance probability $\alpha_i = min\{1, r_i\}$; we then sample $u \sim U(0,1)$, and accept the this move if $u \leq \alpha_i$, reject otherwise.

5 Stop if converged, otherwise goto step 2

BIC [Schwarz, 1978] is a measure of *a posteriori* model fitness, it is the major factor that determines whether or not a proposed move is accepted.

$$BIC = \log(P(x|\Theta)) \cdot \lambda - \frac{1}{2}|\Theta| \log(T) \tag{10.5}$$

Intuitively, BIC is a trade-off between data likelihood $P(X|\Theta)$ and model complexity $|\Theta| \cdot \log(T)$ with weighting factor λ. Larger models are penalized by the number of free parameters in the model $|\Theta|$; yet the influence of the model penalty decreases as the amount of training data T increases, since $\log(T)$ grows slower than $O(T)$. We empirically choose the weighting factor λ as $1/16$ in the simulations of this section as well as those in Section 10.4, in order for the change in data likelihood and that in model prior to be numerically comparable over one iteration.

10.4 Feature selection for unsupervised learning

Feature extraction schemes for audio-visual streams abound, and we are usually left with a large pool of diverse features without knowing which ones are actually relevant to the important events and structures in the data sequences. A few features can be selected manually if adequate domain knowledge exists. But very often such knowledge is not available in new domains, or the connection between high-level structures and low-level features is not obvious. In general, the task of feature selection is divided into two aspects — eliminating *irrelevant* features, and eliminating *redundant* ones. Irrelevant features usually disturb the classifier and degrade classification accuracy, while redundant features add to computational cost without bringing in new information. Furthermore, for unsupervised structure discovery, different subsets of features may relate to different events, and thus the events should be described with separate models rather than being modelled jointly.

Hence the scope of our problem, is to select a relevant and compact feature subset that fits the HHMM model assumption in unsupervised learning over temporally correlated data streams.

10.4.1 Feature selection algorithm

Denote the feature pool as $F = \{f_1, \ldots, f_D\}$, the data sequence as $X_F = X_F^{1:T}$, then the feature vector at time t is X_F^t. The feature selection algorithm proceeds through the following steps, as illustrated in figure 10.2:

1 (Let $i = 1$ to start with.) At the i-th round, produce a *reference set* $\tilde{F}_i \subseteq F$ at random, learn HHMM $\tilde{\Theta}_i$ on $\tilde{F}_i$ with model adaptation,

perform Viterbi decoding of $X_{\tilde{F}_i}$, and obtain the *reference state-sequence* $\tilde{Q}_i = \tilde{Q}^{1:T}_{\tilde{F}_i}$.

2 For each feature $f_d \in F \setminus \tilde{F}_i$, learn HHMM Θ_d, get the Viterbi state sequence Q_d, then compute the information gain (Section 10.4.2) of each feature on the Q_d with respect to the reference partition $\tilde{Q}_i$. We then find the subset $\hat{F}_i \subseteq (F \setminus \tilde{F}_i)$ with significantly large information gain, and form the consistent feature group as union of the *reference set* and the *relevance set*: $\bar{F}_i \stackrel{\triangle}{=} \tilde{F}_i \cup \hat{F}_i$.

3 Use Markov blanket filtering of Section 10.4.3, eliminate redundant features within the set $\bar{F}_i$ whose Markov blanket exists. We are then left with a relevant and compact feature subset $F_i \subseteq \bar{F}_i$. Learn HHMM Θ_i again with model adaptation on X_{F_i}.

4 Eliminate the previous candidate set by setting $F = F \setminus \bar{F}_i$; go back to step 1 with $i = i + 1$ if F is non-empty.

5 For each feature-model combination $\{F_i, \Theta_i\}_i$, evaluate their *fitness* using the normalized BIC criterion in Section 10.4.4, rank the feature subsets, and interpret the meanings of the resulting clusters.

After the feature-model combinations are generated automatically, a human operator can look at the structures marked by these models, then come to a decision on whether a feature-model combination shall be kept based on the meaningfulness of the resulting structures, and the BIC criterion.

10.4.2 Evaluating information gain

Step 1 in Section 10.4.1 produces a reference labelling of the data sequence induced by the classifier learned over the reference feature set.

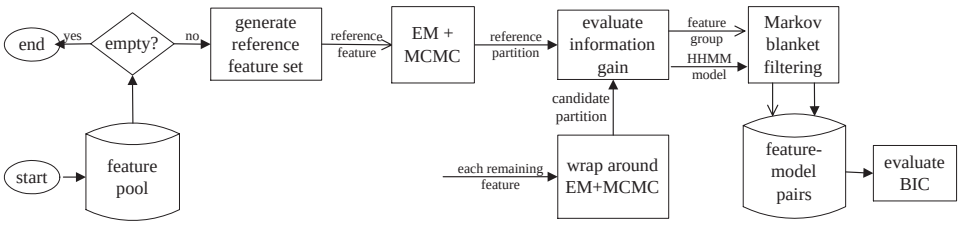

Figure 10.2. Feature selection algorithm overview

We want to find features that are *relevant* to this reference. One suitable measure to quantify the degree of *agreement* in each feature to the reference labelling, as used in [Xing and Karp, 2001], is the mutual information [Cover and Thomas, 1991], or the *information gain* achieved by the new partition induced with the candidate features over the reference partition.

A classifier Θ_F learned over a feature set F generates a partition, i.e. a label sequence Q_F, on the observations X_F, where there are at most N possible labels, we denote the label sequence as integers $Q_F^t \in \{1, \ldots, N\}$. We compute the probability of each label using the empirical portion, by counting the samples that bear label i over time $t = 1, \ldots, T$(Equation 10.6). We compute similarly the conditional probability of the reference labels $\tilde{Q}_i$ for the i-th iteration round given the new partition Q_f induced by a feature f (Equation 10.7), by counting over pairs of labels over time t. Then the information gain of feature f with respect to $\tilde{Q}_i$ is defined as the mutual information between $\tilde{Q}_i$ and Q_f(Equation 10.8).

$$P_{Q_f}(i) = \frac{|\{t|Q_f^t = i, t = 1, \ldots, T\}|}{T}; \tag{10.6}$$

$$P_{\tilde{Q}_i|Q_f}(i \mid j) = \frac{|\{t|(\tilde{Q}_i^t, Q_f^t) = (i, j), t = 1, \ldots, T\}|}{|\{t|Q_f^t = j, t = 1, \ldots, T\}|}; \tag{10.7}$$

$$I(Q_f; \tilde{Q}_i) = H(P_{\tilde{Q}_i}) - \sum_j P_{Q_f} \cdot H(P_{\tilde{Q}_i|Q_f=j}) \tag{10.8}$$

$$\text{where} \quad i, j = 1, \ldots, N$$

Here $H(\cdot)$ is the entropy function. Intuitively, a larger information gain for candidate feature f suggests that the f-induced partition Q_f is more consistent with the reference partition $\tilde{Q}_i$. After computing the information gain $I(Q_f; \tilde{Q}_i)$ for each remaining feature $f_d \in F \setminus \tilde{F}_i$, we perform hierarchical agglomerative clustering on the information gain vector using a dendrogram [Jain et al., 1999], look at the top-most link that partitions all the features into two clusters, and pick features that lies in the upper cluster as the set with satisfactory consistency with the reference feature set.

10.4.3 Finding a Markov blanket

After wrapping information gain criterion around classifiers built over all feature candidates (step 2 in Section 10.4.1), we are left with a subset of features with consistency yet possible redundancy. The approach

for identifying redundant features naturally relates to the conditional dependencies among the features. For this purpose, we need the notion of a Markov blanket [Koller and Sahami, 1996].

DEFINITION 10.1 *Let f be a feature subset, M_f be a set of random variables that does not contain f, we say M_f is the Markov blanket of f, if f is conditionally independent of all variables in $\{F \cup C\} \setminus \{M_f \cup f\}$ given M_f. [Koller and Sahami, 1996]*

Computationally, a feature f is redundant if the partition C of the data set is independent of f given its *Markov blanket* F_M. In prior work [Koller and Sahami, 1996; Xing and Karp, 2001], the Markov blanket is identified with the equivalent condition that the posterior probability distribution of the class given the feature set $\{M_f \cup f\}$ should be the same as that conditioned on the Markov blanket M_f only. i.e.

$$\Delta_f = D(\ P(C|M_f \cup f)\ ||\ P(C|M_f)\ ) = 0 \tag{10.9}$$

where $D(P||Q) = \Sigma_x P(x) \log(P(x)/Q(x))$ is the Kullback-Leibler distance [Cover and Thomas, 1991] between two probability mass functions $P(x)$ and $Q(x)$.

For unsupervised learning over a temporal stream however, this criterion cannot be readily employed. This is because (1) the posterior distribution of a class depends not only on the current data sample, but also on adjacent samples; (2) we would have to condition the class label posterior over all dependent feature samples, and such conditioning quickly makes the estimation of the posterior intractable as the number of conditioned samples grows; (3) we will not have enough data to estimate these high-dimensional distributions by counting over feature-class tuples since the dimensionality is high. We therefore use an alternative necessary condition that the optimum state-sequence $C_{1:T}$ should not change conditioned on observing $M_f \cup f$ or M_f only.

Koller and Sahami have also proved that sequentially removing features one at a time with its Markov blanket identified will not cause divergence of the resulting set, since if we eliminate feature f and keep its Markov blanket M_f, f remains unnecessary in later stages when more features are eliminated. Additionally, as few if any features will have a Markov blanket of limited size in practice, we sequentially remove features that induce the least change in the state sequence given the change is small enough ($< 5\%$). Note this step is a filtering step in our HHMM learning setting, since we do not need to retrain the HHMMs for each candidate feature f and its Markov blanket M_f. Given the HHMM trained over the set $f \cup M_f$, the state sequence Q_{M_f}, decoded with the observation sequences in M_f only, is compared with the state sequence

$Q_{f \cup M_f}$ decoded using the whole observation sequence in $f \cup M_f$. If the difference between Q_{M_f} and $Q_{f \cup M_f}$ is *small enough*, then f is removed since M_f is found to be a Markov blanket of f.

10.4.4 Normalized BIC

Iterating over Section 10.4.2 and Section 10.4.3 results in disjoint small subsets of features $\{F_i\}$ that are compact and consistent with each other. The HHMM models $\{\Theta_i\}$ learned over these subsets are best-effort fits on the features, yet the $\{\Theta_i\}$s may not fit the multi-level Markov assumptions in Section 10.0.0.0.0.

There are two criteria proposed in prior work [Dy and Brodley, 2000], scatter separability and maximum likelihood (ML). Note the former is not suitable to temporal data since multi-dimensional Euclidean distance does not take into account temporal dependency, and it is non-trivial to define another proper distance measure for temporal data; while the latter is also known [Dy and Brodley, 2000] to be biased against higher-dimensional feature sets. We use a normalized BIC criterion (Equation 10.10) as the alternative to ML, which trades off normalized data likelihood $\tilde{L}$ with model complexity $|\Theta|$. Note the former has weighting factor λ in practice; the latter is modulated by the total number of samples $\log(T)$; and $\tilde{L}$ for HHMM is computed in the same forward-backward iterations, except all the emission probabilities $P(X|Q)$ are replaced with $P'_{X,Q} = P(X|Q)^{1/D}$, i.e. normalized with respect to data dimension D, under the *naive-Bayes* assumption that features are independent given the hidden states.

$$\widetilde{BIC} = \tilde{L} \cdot \lambda - \frac{1}{2}|\Theta| \log(T) \tag{10.10}$$

Initialization and convergence issues exist in the iterative partitioning of the feature pool. The strategy for producing the random *reference set* $\tilde{F}_i$ in step (1) affects the result of feature partition, as even producing the same $\tilde{F}_i$ in a different sequence may result in different final partitions. Moreover, the *expressiveness* of the resulting structures is also affected by the reference set. If the dimension of $\tilde{F}_i$ is too low for example, the algorithm tends to produce many small feature groups where features in the same group mostly *agree* with each other, and the learned model would not be able to identify potential complex structures that must be identified with features carrying complementary information, such as features from different modalities (audio and video). On the other hand, if $\tilde{F}_i$ is of very high dimension, then the information gain criterion will give a large feature group around $\tilde{F}_i$, thus mixing different event

streams that would better be modelled separately, such as the activity of pedestrians and vehicles in a street surveillance video.

10.5 Experiments and Results

In this section, we report the tests of the proposed methods in automatically finding salient events, learning model structures, and identifying informative feature set in soccer and baseball videos. We have also experimented with variations in HHMM transition topology and found that the additional hierarchical structure imposed by HHMM over an ordinary HMM introduces more modelling power on our test domain.

Sports videos represent an interesting domain for testing the proposed techniques in automatic structure discovery. Two main factors contribute to this match between the video domain and the statistical technique: the distinct set of semantics in the sports domain exhibit strong correlations with audio-visual features; the well-established rules of games and production syntax in sports video programs impose strong temporal transition constraints. For example, in soccer videos, *plays* and *breaks* are recurrent events covering the entire time axis of the video data. In baseball videos, transitions among different perceptually distinctive mid-level events, such as pitching, batting, running, are semantically significant for the game.

Clip Name	Sport	Length	Resolution	Frame rate	Source
Korea	Soccer	25'00"	320×240	29.97	MPEG-7
Spain	Soccer	15'00"	352×288	25	MPEG-7
NY-AZ	Baseball	32'15"	320×240	29.97	TV program

Table 10.1. Sports video clips used in the experiment.

All our test videos are in MPEG-1 format, their profiles are listed in Table 10.1. For soccer videos, we have compared with our previous work using supervised methods on the same video streams [Xie et al., 2002b]. The evaluation basis for the structure discovery algorithms is for two semantic events, *play* and *break*, defined according to the rules of soccer. These two events are dense since they cover the whole time scale of the video, and distinguishing *break* from *play* will be useful for efficient browsing and summarization, since *break* takes up about 40% of the screen time, and viewers may browse through the game play by play, skipping all the breaks in between, or randomly access the break segments to find player responses or game announcements. For baseball videos, we conducted the learning without having labelled ground truth or manually identified features *a priori*, and an human observer (the

first author) reported observations on the selected feature sets and the resulting structures afterwards. This is analogous to the actual application of structure discovery to an unknown domain, where evaluation and interpretation of the result is done after automatic discovery algorithms are applied.

It is difficult to define general evaluation criteria for automatic structure discovery results that are applicable across different domains, this is especially the case when domain-specific semantic labels are of interest. This difficulty lies in the gap between computational optimization and semantic meaning: the results of unsupervised learning are optimized with measures of statistical fitness, yet the link from statistical fitness to semantics needs a match between general domain characteristics and the computational assumptions imposed in the model. Despite this difficulty, our results have shown support for constrained domains such as sports. Effective statistic models built over statistically optimized feature sets have good correspondence with semantic events in the selected domain.

10.5.1 Parameter and structure learning

We first test the automatic model learning algorithms with a fixed feature set manually selected based on heuristics. The selected features, *dominant color ratio* and *motion intensity*, have been found effective in detecting soccer events in our prior work [Xu et al., 2001; Xie et al., 2002b]. Such features are uniformly sampled from the video stream every 0.1 second. Here we compare the learning accuracy of four different learning **schemes** against the ground truth.

1 Supervised HMM: This is developed in our prior work [Xie et al., 2002b]. One HMM per semantic event (i.e., play and break) is trained on manually defined chunks. For test video data with unknown event boundaries, the videos are first chopped into 3-second segments, where the data likelihood of each segment is evaluated with each of the trained HMMs. The final event boundaries are refined with a dynamic programming step taking into account the model likelihoods, the transition likelihoods between events, and the probability distribution of event durations.

2 Supervised HHMM: Individual HMMs at the bottom level of the hierarchy are learned separately, essentially using the models trained in scheme 1; across-level and top level transition statistics are also obtained from segmented data; and then segmentation is obtained by decoding the Viterbi path from the hierarchical model on the entire video stream.

3 Unsupervised HHMM without model adaptation: An HHMM is initialized with known size of state-space and random parameters; the EM algorithm is used to learn the model parameters; and segmentation is obtained from the Viterbi path of the final model.

4 Unsupervised HHMM with model adaptation: An HHMM is initialized with arbitrary size of state-space and random parameters; the EM and RJ-MCMC algorithms are used to learn the size and parameters of the model; the state sequence is obtained from the converged model with optimal size. Here we will report results separately for (a) model adaptation in the lowest level of HHMM only, and (b) full model adaptation across different levels as described in Section 10.3.

For supervised schemes 1 and 2, K-means clustering and Gaussian mixture fitting is used to randomly initialize the HMMs. For unsupervised schemes 3 and 4, as well as all full HHMM learning schemes in the sections that follow, the initial emission probabilities of the initial bottom-level HMMs are obtained with K-means and Gaussian fitting; and then the multi-level Markov chain parameters are estimated using a dynamic programming technique that groups the states into different levels by maximizing the number of within-level transitions, while minimizing inter-level transitions among the Gaussians. For schemes 1-3, the model size is set to six bottom-level states per event, corresponding to the optimal model size that schemes 4a converges to, i.e. six to eight bottom-level states per event. We run each algorithm for 15 times with random start, and compute the per-sample accuracy against manual labels. The median and semi-interquartile range [2] across multiple rounds are listed in Table 10.2.

Learning Scheme	Supervised?	Model type	Adaptation?		Accuracy	
			Bottom-level	High-levels	Median	SIQ
(1)	Y	HMM	N	N	75.5%	1.8%
(2)	Y	HHMM	N	N	75.0%	2.0%
(3)	N	HHMM	N	N	75.0%	1.2%
(4a)	N	HHMM	N	Y	75.7%	1.1%
(4b)	N	HHMM	Y	Y	75.2%	1.3%

Table 10.2. Evaluation of learning schemes (1)-(4) against ground truth using on clip *Korea*

[2]Semi-interquartile as a measure of the spread of the data, is defined as half of the distance between the 75th and 25th percentile, it is more robust to outliers than standard deviation.

Results show that the performance of the unsupervised learning scheme is comparable to the supervised learning, and sometimes it achieves even slightly better accuracy than the supervised learning counterpart. This is quite surprising since the unsupervised learning of HHMMs is not tuned to the particular ground-truth. The results maintain a consistent accuracy, as indicated by the low semi-interquartile range. Also note that the comparison basis using supervised learning is actually conservative since (1) unlike [Xie et al., 2002b], the HMMs are learning and evaluated on the same video clip and results reported for schemes 1 and 2 are actually training accuracies; (2) the models without structure adaptation are assigned the *a posteriori* optimal model size.

For the HHMM with full model adaptation (scheme 4b), the algorithm converges to two to four high-level states, and the evaluation is done by assigning each resulting cluster to the majority ground-truth label it corresponds to. We have observed that the resulting accuracy is still in the same range without knowing how many interesting structures there is to start with. And the reason for this performance match lies in the fact that the *additional* high level structures are actually a sub-cluster of *play* or *break*, they are generally of three to five states each, and two sub-clusters correspond to one *larger, true* cluster of play or break (refer to a three-cluster example in Section 10.5.2).

10.5.2 Feature selection

Based on the good performance of the model parameter and structure learning algorithm, we test the performance of the automatic feature selection method that iteratively *wraps* around, and *filters* (Section 10.4). We use the two test clips, Korea and Spain as profiled in Table 10.1. A nine-dimensional feature vector sampled at every 0.1 seconds is taken as the initial feature pool, including:

> Dominant Color Ratio (DCR), Motion Intensity (MI), the least-square estimates of camera translation (MX, MY), and five audio features - Volume, Spectral roll-off (SR), Low-band energy (LE), High-band energy (HE), and Zero-crossing rate (ZCR).

We run the feature selection method plus model learning algorithm on each video stream for five times, with a one or two-dimensional feature set as initial reference set in each iteration. After eliminating degenerate cases that only consist of one feature in the resulting set, we evaluate the feature-model pair that has the largest *Normalized BIC* value as described in Section 10.4.4.

For clip *Spain*, the selected feature set is {DCR, Volume} The model converges to two high-level states in the HHMM, each with five lower-

level children states. Evaluation against the *play/break* labels showed a 74.8% accuracy. For clip *Korea*, the final selected feature set is {DCR, MX}, with three high-level states and {7, 3, 4} children states respectively. If we assign each of the three clusters to the semantic event that it agrees with for the most amount of times (which would be {*play, break, break*} respectively), per-sample accuracy would be 74.5%. The automatic selection of DCR and MX as the most relevant features is actually consistent with the manual selection of the two features DCR and MI in our prior work [Xie et al., 2002b; Xu et al., 2001]. MX is a feature that approximates the horizontal camera panning motion, which is the most dominant factor contributing to the overall motion intensity (MI) in soccer video, as the camera needs to track the ball movement in wide angle shorts, and wide angle shots are one major type of shot that is used to reveal overall game status [Xu et al., 2001].

The accuracies are comparable to their counterpart (scheme 4) in Section 10.5.1 without varying the feature set (75%). Yet the small discrepancy may due to (1) variability in RJ-MCMC (Section 10.3), for which convergence diagnostic is still an active area of research [Andrieu et al., 2003], and (2) possible inherent bias that may exist in the normalized BIC criterion (Equation 10.10), as we will need ways to further calibrate the criterion.

10.5.3 Testing on a different domain

We have also conducted a preliminary study on the baseball video clip described in Table 10.1. The same 9-dimensional feature pool as in Section 10.5.2 is extracted from the stream, also at 0.1 second per sample. The learning of models is carried out without having labelled ground truth or manually identified features *a priori*. Observations are reported based on the selected feature sets and the resulting structures of the test results. This is a standard process of applying structure discovery to an unknown domain, where automatic algorithms serve as a pre-filtering step, and evaluation and interpretation of the result can only be done afterwards.

HHMM learning with full model adaptation and feature selection is conducted, resulting in three consistent compact feature groups: (a) HE, LE, ZCR; (b) DCR, MX; (c) Volume, SR. It is interesting to see audio features fall into two separate groups, with visual features also in a individual group.

The BIC score for the second group, dominant color ratio and horizontal camera pan, is significantly higher than that of the other two. The HHMM model in (b) has two higher-level states, each has six and seven

children states at the bottom level, respectively. Moreover, the resulting segments from the model learned with this feature set has consistent perceptual properties, with one cluster of segments mostly corresponding to pitching shots and other field shots when the game is in play, while the other cluster contains most of the cutaways shots, score boards and game breaks, respectively. It is not surprising that this result agrees with the intuition that the status of a game can mainly be inferred from visual information.

10.5.4 Comparing to HHMM with simplifying constraints

In order to investigate the *expressiveness* of the multi-level model structure, we compare unsupervised structure discovery performances of the HHMM with a similar model with constrains in the transitions each node can make.

The two model topologies being simulated are visualized in figure 10.3:

(a) The simplified HHMM where each bottom-level sub-HMM is a left-to-right model with skips, and cross level entering/exiting can only happen at the first/last node, respectively. Note that the right-most states serving as the single exit point from the bottom level eliminate the need for a special *exiting* state.

(b) The fully connected general 2-level HHMM model used in scheme 3, Section 10.5.1, a special case of the HHMM in figure 10.1. Note the dummy *exiting* cannot be omitted in this case.

Topology (a) is of interest because the left-to-right and single entry/exit point constraints enables the learning the model with the algorithms designed for ordinary HMMs by *collapsing* this model to an ordinary HMM. The *collapsing* can be done because unlike the general HHMM case (Section 10.1), there is no ambiguity in whether or not a cross-level has happened in the original model given the last state and the current state in the collapsed model, or equivalently, the *flattened* HMM transition matrix can be uniquely factored back to recover the multi-level transition structure. Note that here the trade-off for model generality is that parameter estimation of the flattened HMMs is of complexity $O(T|Q|^{2D})$, while HHMMs will need $O(DT|Q|^{2D})$, as analyzed in Section 10.1.2. With the total number of levels D typically a fixed small constant, this difference does not influence the scalability of the model to long sequences.

Topology (a) also contains models proposed in two prior publications as special cases: [Clarkson and Pentland, 1999] uses a left-to-right model

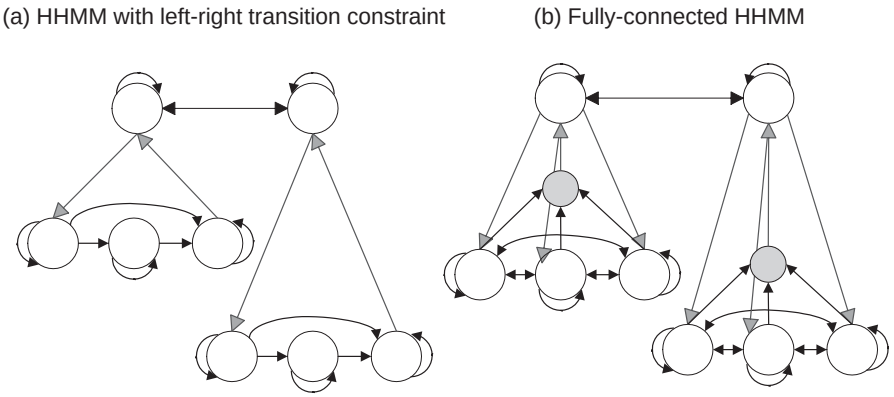

Figure 10.3. Comparison with HHMM with left-to-right transition constraints. Only 3 bottom-level states are drawn for the readability of this graph, models with 6-state sub-HMMs are simulated in the experiments.

without skip, and single entry/exit states; [Naphade and Huang, 2002] uses a left-to-right model without skip, single entry/exit states with one single high-level state, i.e. the probability of going to each sub-HMM is independent of which sub-HMM the model just came from, thus eliminating one more parameter from the model than [Clarkson and Pentland, 1999]. Both of the prior cases are learned with HMM learning algorithms.

This learning algorithm is tested on the soccer video clip *Korea*; it performs parameter estimation with fixed model structure of six states at the bottom level and two states at the top level, over the pre-defined features set of DCR and MI (Section 10.5.1). Results show that over 5 runs of both algorithms, the average accuracy of the constrained model is 2.3% lower than that of the fully connected model. This shows that adopting a fully connected model with multi-level control structures indeed brings in extra modelling power for the chosen domain of soccer videos.

10.6 Conclusion

In this chapter we proposed algorithms for unsupervised discovery of structure from video sequences. We model the class of dense, stochastic structures in video using hierarchical hidden Markov models. The model parameters and model structure are learned using EM and Monte Carlo sampling techniques, and informative feature subsets are automatically

selected from a large feature pool using an iterative filter-wrapper algorithm. When evaluated on TV soccer clips against manually labelled ground truth, we achieved results comparable to its supervised learning counterpart; when evaluated on baseball clips, the algorithm automatically selects two visual features, which agrees with our intuition that the status of a baseball game can be inferred from visual information only.

It is encouraging that in constrained domains such as sports, effective statistic models built over statistically optimized feature sets without human supervision have good correspondence with semantic events. We believe this success lends major credit to the correct choice in general model assumptions, and the selected test domain that matches this assumption. This unsupervised structure discovery framework leaves much room for generalizations and applications to many diverse domains. It also raises further theoretical issues that will enrich this framework if successfully addressed: modelling sparse events in domains such as surveillance videos; online model update using new data; novelty detection; automatic pattern association across multiple streams; a hierarchical modeling that automatically adapts to different temporal granularity; etc.

Appendix

Proposal probabilities for model adaptation.

$$p_{sp}(d) = c^* \cdot \min\{1, \rho/(k+1)\}; \quad (10.A.1)$$

$$p_{me}(d) = c^* \cdot \min\{1, (k-1)/\rho\}; \quad (10.A.2)$$

$$p_{sw}(d) = c^*; \quad (10.A.3)$$

$$p_{em} = 1 - \Sigma_{d=1}^{D}[p_{sp}(d) + p_{me}(d) + p_{sw}(d)]. \quad (10.A.4)$$

Here c^* is a simulation parameter, k is the current number of states; and ρ is the hyper-parameter for the truncated Poisson prior of the number of states [Andrieu et al., 2003], i.e. ρ would be the expected mean of the number of states if the maximum state size is allowed to be $+\infty$, and the scaling factor that multiplies c^* modulates the proposal probability using the resulting state-space size $k \pm 1$ and ρ.

Computing different moves in RJ-MCMC. EM is one regular hill-climbing iteration as described in Section 10.2; and once a move type other than EM is selected, one (or two) states at a certain level are selected at random for swap/split/merge, and the parameters are modified accordingly:

- Swap the association of two states:
  Choose two states from the same level, each belongs to a different higher-level state, swap their higher-level association.
- Split a state:
  Choose a state at random, the split strategy differs when this state is at different position in the hierarchy.

– When this is a state at the lowest level ($d = D$), perturb the mean of its associated Gaussian observation distribution as follows

$$\begin{aligned} \mu_1 &= \mu_0 + u_s\eta \\ \mu_2 &= \mu_0 - u_s\eta \end{aligned} \tag{10.A.5}$$

where $u_s \sim U[0,1]$, and η is a simulation parameter that ensures reversibility between split moves and merge moves.

– When this is a state at $d = 1, \ldots, D-1$ with more than one children states, split its children into two disjoint sets at random, generate a new *sibling* state at level d associated with the same parent as the selected state. Update the corresponding multi-level Markov chain parameters accordingly.

- Merge two states:
  Select two *sibling* states at level d, merge the observation probabilities or the

  – When $d = D$, merge the Gaussian observation probabilities by making the new mean as the average of the two.

$$\mu_0 = \frac{\mu_1 + \mu_2}{2}, \quad \text{if } |\mu_1 - \mu_2| \leq 2\eta \tag{10.A.6}$$

here η is the same simulation parameter as in Eq. 10.A.5.

– When $d = 1, \ldots, D-1$, merge the two states by making all the children of these two states the children of the merged state, and modify the multi-level transition probabilities accordingly.

The acceptance ratio for different moves in RJ-MCMC. The acceptance ratio for *Swap* simplifies into the posterior ratio since the dimension of the space does not change. Denote Θ as the old model and $\hat{\Theta}$ as the new model :

$$r \stackrel{\triangle}{=} \text{(posterior ratio)} = \frac{P(x|\hat{\Theta})}{P(x|\Theta)} = \frac{exp(\widehat{BIC})}{exp(BIC)} \tag{10.A.7}$$

When moves are proposed to a parameter space with different dimension, such as split or merge, we also need a proposal ratio term and a Jacobian term to align the spaces in order to ensure detailed balance [Green, 1995], as shown in Equations (10.A.8)–(10.A.11).

$$r_k \stackrel{\triangle}{=} \text{(posterior ratio)} \cdot \text{(proposal ratio)} \cdot \text{(Jacobian)} \tag{10.A.8}$$

$$r_{split} = \frac{P(k+1, \Theta_{k+1}|x)}{P(k, \Theta_k|x)} \cdot \frac{m_{k+1}/(k+1)}{p(u_s)s_k/k} \cdot J \tag{10.A.9}$$

$$r_{merge} = \frac{P(k, \Theta_k|x)}{P(k+1, \Theta_{k+1}|x)} \cdot \frac{p(u_s)s_{k-1}/(k-1)}{m_k/k} \cdot J^{-1} \tag{10.A.10}$$

$$J = \left|\frac{\partial(\mu_1, \mu_2)}{\partial(\mu_0, u_s)}\right| = \begin{vmatrix} 1 & \eta \\ 1 & -\eta \end{vmatrix} = 2\eta \tag{10.A.11}$$

References

Andrieu, C., de Freitas, N., and Doucet, A. (2001). Robust full bayesian learning for radial basis networks. *Neural Computation*, 13:2359–2407.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, special issue on MCMC for Machine Learning.

Brand, M. (1999). Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182.

Clarkson, B. and Pentland, A. (1999). Unsupervised clustering of ambulatory audio and video. In *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley, New York.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum liklihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.

Doucet, A. and Andrieu, C. (2001). Iterative algorithms for optimal state estimation of jump Markov linear systems. *IEEE Transactions of Signal Processing*, 49:1216–1227.

Dy, J. G. and Brodley, C. E. (2000). Feature subset selection and order identification for unsupervised learning. In *Proc. 17th International Conf. on Machine Learning*, pages 247–254. Morgan Kaufmann, San Francisco, CA.

Fine, S., Singer, Y., and Tishby, N. (1998). The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.

Hu, M., Ingram, C., Sirski, M., Pal, C., Swamy, S., and Patten, C. (2000). A hierarchical HMM implementation for vertebrate gene splice site prediction. Technical report, Dept. of Computer Science, University of Waterloo.

Ivanov, Y. A. and Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Transaction of Pattern Recognition and Machines Intelligence*, 22(8):852–872.

Iyengar, A., Squillante, M. S., and Zhang, L. (1999). Analysis and characterization of large-scale web server access patterns and performance. *World Wide Web*, 2(1-2):85–100.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292.

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (October 1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 8(262):208–14.

Murphy, K. (2001). Representing and learning hierarchical structure in sequential data.

Murphy, K. and Paskin, M. (2001). Linear time inference in hierarchical HMMs. In *Proceedings of Neural Information Processing Systems*, Vancouver, Canada.

Naphade, M. and Huang, T. (2002). Discovering recurrent events in video using unsupervised methods. In *Proc. Intl. Conf. Image Processing*, Rochester, NY.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Sahouria, E. and Zakhor, A. (1999). Content anlaysis of video using principal components. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(9):1290–1298.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 7:461–464.

The HTK Team (2000). Hidden Markov model toolkit (HTK3). http://htk.eng.cam.ac.uk/.

Wang, Y., Liu, Z., and Huang, J. (2000). Multimedia content analysis using both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6):12–36.

Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. (2002a). Learning hierarchical hidden Markov models for video structure discovery. Technical Report ADVENT-2002-006, Dept. Electrical Engineering, Columbia Univ., http://www.ee.columbia.edu/~xlx/research/.

Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. (2002b). Structure analysis of soccer video with hidden Markov models. In *Proc. Interational Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Orlando, FL.

Xing, E. P. and Karp, R. M. (2001). Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. In *Proceedings of the Ninth International Conference on Intelligence Systems for Molecular Biology (ISMB)*, pages 1–9.

Xu, P., Xie, L., Chang, S.-F., Divakaran, A., Vetro, A., and Sun, H. (2001). Algorithms and systems for segmentation and structure anal-

ysis in soccer video. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan.

Yeung, M. and Yeo, B.-L. (1996). Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition (ICPR)*, Vienna, Austria.