Multilevel Chinese Takeaway Process and Label-Based Processes for Rule Induction in the Context of Automated Sports Video Annotation

Aftab Khan, David Windridge, and Josef Kittler

***Abstract*—We propose four variants of a novel hierarchical hidden Markov models strategy for rule induction in the context of automated sports video annotation including a multilevel Chinese takeaway process (MLCTP) based on the Chinese restaurant process and a novel Cartesian product label-based hierarchical bottom-up clustering (CLHBC) method that employs prior information contained within label structures. Our results show significant improvement by comparison against the flat Markov model: optimal performance is obtained using a hybrid method, which combines the MLCTP generated hierarchical topological structures with CLHBC generated event labels. We also show that the methods proposed are generalizable to other rule-based environments including human driving behavior and human actions.**

***Index Terms*—Chinese restaurant process, hidden Markov models, hierarchical HMMs, stick-breaking construction, video annotation.**

I. Introduction

MULTIMEDIA data production has grown exponentially over the past decade. This data exists in various forms such as broadcast content (including television news and sports), personal content (e.g., social media videos including mobile phone footage), recorded interviews or meetings and footage from surveillance cameras, and so on. Availability of high quality digital hand-held camcorders has also facilitated the expansion of video recordings at a domestic consumer level. Most of this data is generally intended for general viewing and hence basic labeling (date, time, title, and so on) is attached to it. However, in many cases it would be useful to add additional labels to retrieve information in a more flexible and systematic fashion (e.g., a tennis sports video can potentially be labeled with match-events description). Such metadata will assist in finding material within the multimedia footage via browsing, querying, or searching.

Sports videos have a high demand for automatic annotation as there is considerable interest in browsing key events (such as goals in football). Complete annotation may also be used to extract match statistics and to construct performance analysis of teams. For easy retrieval of information from a very large quantity of archived footage, it would be very useful to have sports videos annotated automatically [8], [35], [52], i.e., to create a system that could understand the content of the video (manual annotation being too unwieldy).

Sports videos consist of rich multimedia content, as well as contextual details. Key temporal event information is critical in understanding sports videos. Sports games in general have a rule structure, built around low level visual events that are further interpreted as game events and similar high level contextual information. Events can thus be expressed in the form of a hierarchical structure. For example, in a game of tennis, low level visual events include tennis ball transitions within the court, and player movements enacting game play on the court surface. These transitions can be interpreted in a more contextualized form such as a hit taking place at a particular location on a court box. These high-level events can then be combined to describe the tennis game, incorporating all the rule salient temporal details as annotations.

Hidden Markov Models (HMMs) [37] are often used to represent stochastic processes, and can effectively model temporal sequences of data [e.g., stock market [18], audio/video signals [27], and patient's electrocardiography (ECG) [26], and so on]. However, as indicated, some domains such as sports games in general are hierarchical in nature, with a clear delineation between low-level visual representations and progressively higher levels of contextual interpretations. If a game is to be modeled stochastically, with various levels of progressive abstractions, this implies the use of the hierarchical HMMs (hHMMs) [12] to model game transitions at different levels of contextual interpretation.

However, a particular disadvantage of the classical HMM framework is that it generally requires the number of states to be fixed *a priori*, and in practical applications they are usually fixed heuristically. Teh *et al.* [43] have proposed a non-parametric Bayesian implementation of HMM in which

Manuscript received January 14, 2013; revised June 29, 2013 and October 25, 2013; accepted December 30, 2013. This work was supported by EPSRC under Grant EP/F069421/1. This paper was recommended by Associate Editor W. Hu.

A. Khan is with Culture Lab, School of Computing Science, Newcastle University, Newcastle upon Tyne, Tyne and Wear NE1 7RU, U.K. (e-mail: aftab.khan@newcastle.ac.uk).

D. Windridge and J. Kittler are with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: d.windridge@surrey.ac.uk; j.kittler@surrey.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2299955

the hierarchical Dirichlet process (HDP) provides a prior distribution over countably infinite state spaces resulting in a generalized HMM. Hierarchical Dirichlet process HMMs (HDP-HMMs) have been effectively employed in tackling different problems such as visual scene recognition [23], and the modeling of genetic recombination [49], and so on.

Our aim is to achieve automated stochastic rule induction for a rule-based sport game environment. We make use of the non-parametric Chinese restaurant process (CRP) [1] to produce hierarchical structures with states and a stick-breaking construction [42] to generate their probabilistic state transitions, i.e., we systematically parametrize hHMMs to build a game rule model. As a variant on this approach, we also propose a novel label-based hierarchical method to build hHMMs and show the significance of having prior knowledge of a labeled system in the construction of the hierarchy.

We thus compare a number of derived hHMM models against the flat Markov Model, which serves as the baseline for all our methodological variants.

First, we propose a new label-based method that takes into account the actual label structure that defines a particular game play sequence in order to define an hHMM generation method that proceeds in a bottom-up, data driven fashion. We call this methodological variant, Cartesian product label-based hierarchical bottom-up clustering (CLHBC).

A further variant is introduced via a novel implementation of the CRP called the multilevel Chinese takeaway process (MLCTP). This is a constrained version of the standard CRP that is more relevant to applications with a limited state space, i.e., where the number of rule-defining events are known and a limited rule depth is present, i.e., rule induction occurs under a certain unknown, but relatively limited number of levels.

MLCTP does not intrinsically exploit labeled states, and we speculate that the highest likelihood inferred rule structure, given a set of hyper-parameters representing the MLCTP model, can be further improved via employing the label structures. Thus, we also propose two hybrid methods that combine the unlabeled MLCTP with the labeled structure from the flat Markov model and CLHBC. The main idea is thus to combine the hierarchical topological structures of MLCTP with the event label transition probabilities.

We show comparative results of all the proposed methods on the following various datasets from different domains.

1) *Badminton*: Ground truth annotated events in badminton (mens singles, Czech versus Great Britain, Beijing Olympics, 2008).
2) *Tennis*: Ground truth annotated events extracted from a complete tennis match (S. Williams versus V. Williams, women's final, Australian Open, 2003).
3) *Tennis*: Labeled sequential events in tennis obtained via a computer vision based annotator [25], [30] (S. Williams versus V. Williams, women's final, and A. Agassi versus R. Schüttler, Men's final at Australian Open, 2003).
4) *Highway Rules*: Ground truth annotated events obtained from a camera-equipped car driven across a city [47].
5) *Website Domain*: Website visit counts at different web pages within MSNBC.com on September 28, 1999 [4].
6) *Human Activity Dataset*: Recordings of five sensor-tagged people performing different actions [19] such as sitting, walking, lying, and so on.

List of terminologies used in this paper are shown in Table I.

TABLE I
TERMINOLOGIES USED IN PAPER

Symbol	Description
Ω_i	Label components indexed by i
$L_t^{\{k\}}$	Event label at time t composed with the omitted label set $\{k\}$
Q_n^h	CLHBC-defined hidden state number n at level h
$\mathscr{E}_X$	Observations indexed by $X = 1, 2, 3, ..., G$
α, γ	MLCTP's concentration parameters
$\mathscr{G}$	MLCTP's truncation parameter
H	MLCTP-defined number of levels
o_c	Number of people at takeaway c in MLCTP model
$\mathscr{C}^h$	Number of states at level h in MLCTP
${}_x\zeta_y^h$	MLCTP generated state number y, with parent state number x and at level h
π	Stick-breaking construction weights for transition probabilities
${}_x\delta_y^h$	Self transition probability for state ${}_x\zeta_y^h$
${}_x\psi_y^h$	Remaining transition probability for state ${}_x\zeta_y^h$ i.e. $1 - {}_x\delta_y^h$
R	Total number of MLCTP-generated Topologies
s	Number of $\mathscr{G}$-defined selected Topologies
Z	Total number of MLCTP-generated transition matrices
λ_h	HMM parameters for level h
A_{ij}^h	Estimated state transition probabilities from state i to j

II. RELATED WORK

Automated video annotation is one of the classic research problems in computer vision which includes challenges at various levels. Graphical models are usually used in creation of decision-making systems in this context and one of the most substantial development has been the introduction of HMMs [37]. Event detection and action recognition are the key related problems [14], [31], [36], [45].

Application domains vary from surveillance [11] to processing entertainment movies [29] and TV shows [34]. Sign language recognition has also been explored [6].

Sports videos are also examined in [9] and [50] where a range of methodologies have been implemented. Soccer is one of the most widely explored application domains as it has a very challenging player tracking problem with severe levels of occlusion [10], [15], [16], [39], [40], [48]. Among other sports, cricket [5] and snooker [7] have also been explored for umpire's gesture recognition and video summarization.

Tennis videos have been employed for various research tasks such as, shot classification [9], [21], within-shot event detection [38], tennis ball based event recognition [2], players' stroke type classification [33], analysis of player tactics [53], and scene retrieval [44]. Additionally, problems like anomaly detection [3], anomaly rectification [20], and domain change detection [22] have also been addressed in this context.

In this paper, we mainly deal with the contextual level representations of tennis by using observations from players, and cues from other types of agents such as the court lines and the ball. Our goal is to create a generic sport rule induction system based on tennis' rich label structure that can be employed to generate hierarchical HMMs for the creation of a complete reasoning system. We also aim to create cognitive

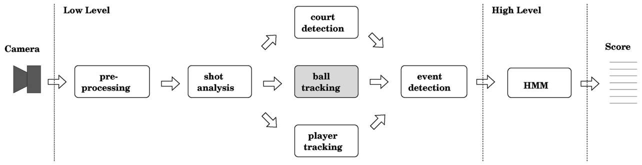


Fig. 1. Tennis video annotation system.

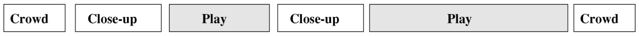


Fig. 2. Illustrative example of the composition of a tennis video. The length of each shot is proportional to the width of the corresponding block in the figure.

systems that will be able to learn to use context that is specific to the required application, i.e., via domain adaptation.

In the next section, our tennis video annotation system is introduced.

III. Tennis Video Annotation System

We aim to provide a generalizable high-level module for the tennis video annotation system of [24] and [25] in this paper. Before moving on to discussing stochastic rule induction algorithms, we first briefly introduce our tennis annotation system. Fig. 1 shows the simplified block diagram of the annotation system with individual processing sections. Each section has individual modules.

1) *Low-Level Preprocessing:* In the low-level block, image frames are initially de-interlaced into fields as some of the tennis videos employed in our experiments are captured with interlaced cameras. Fields are used to remove the effects of temporal aliasing and is important for tennis ball tracking. After de-interlacing, the geometric distortion of camera lens is corrected. It is assumed that the camera position on the court is fixed, and the global transformation between frames is assumed to be a homography [17]. The homography is found by: 1) tracking corners through the sequence; 2) applying RANSAC [13] to the corners to find a robust estimate of the homography; and 3) finally applying a Levenberg-Marquardt optimizer [28] to improve the homography.

2) *Shot Analysis:* A broadcast tennis video is composed of shots, such as play, close-up, crowd, and commercials. An illustrative example of the composition of a tennis video is shown in Fig. 2. Shot boundaries are detected by using color histogram intersection between adjacent frames that are then classified into appropriate types by using a combination of color histogram mode and corner point continuity.

3) *Court Detection, Ball Tracking, and Player Tracking:* For a play shot, the tennis court is detected through a combination of edge detection and Hough transformation. The players are tracked using a particle filter, and player actions are classified (see [33], [41] for details). For ball tracking, background subtraction is employed to generate candidate ball blobs. A simple feature vector is computed for each blob, describing its size, colors, and edges. SVM classification is then performed to extract strong ball candidates [51].

The ball tracks are established in two stages. First, tracklets are built from sets of extracted strong object candidates in the form of second-order (roughly parabolic) trajectories. Then a graph-theoretic data-association technique is used to link the tracklets into complete ball tracks [51].

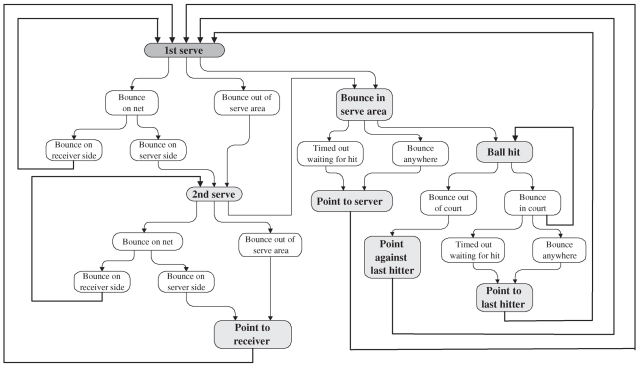


Fig. 3. True tennis rule model as defined in [24] and [25].

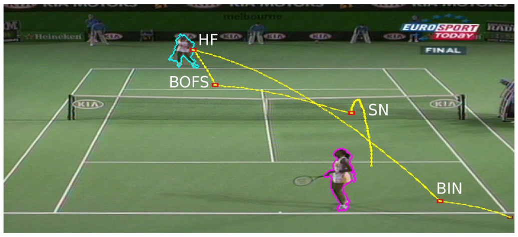


Fig. 4. Tennis playshot with detected low-level features such as player positions, ball trajectories, key events (red squares) and high-level labeled annotations (see Table II).

4) *Event Detection:* By examining the tennis ball trajectories, motion discontinuity points are detected. These points are combined with player positions, player actions and court lines in the event detection module to generate key event description such as hit, bounce, and net.

5) *High-Level Reasoning:* Finally, the generated key events are sent to a hardwired high level module, where the tennis rules are incorporated into an HMM. The HMM is used as a reasoning tool to generate the annotation, i.e., outcome of play, point awarded, and so on (see [25] for additional details). Single frame output of the annotation system is shown in Fig. 4.

It is this module that we propose to replace with a generic model able to—ultimately—learn rules of any input game. Fig. 3 shows the non-hierarchical tennis rule model used by Kolonias *et al.* [24], [25] to determine game scores. Our aim is to autonomously learn (instead of predefining) such rule models in a hierarchical fashion that is also applicable to other domains in addition to tennis.

IV. Cartesian Product Label-Based Hierarchical Bottom-up Clustering

A. Introduction

Sports games have a specific rule structure built around temporal events that are based on transitions between labeled states according to structured game rules as follows.

1) Football: kick, pass left, pass right, and so on.
2) Tennis/Badminton: Serve near left, hit far, and so on.
3) Cricket: Square drive, straight drive, yorker, and so on.

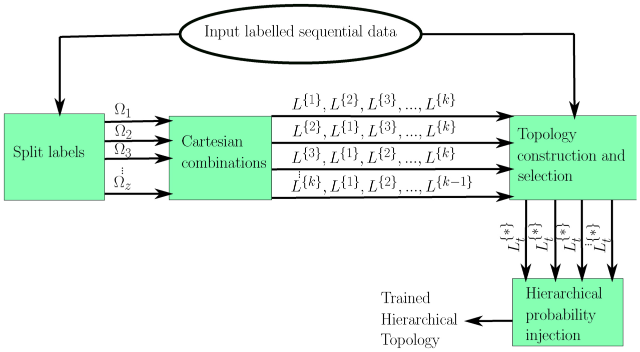

Input labelled sequential data
Split labels
Ω_1
Ω_2
Ω_3
Ω_z
Cartesian combinations
$L^{\{1\}}, L^{\{2\}}, L^{\{3\}}, ..., L^{\{k\}}$
$L^{\{2\}}, L^{\{1\}}, L^{\{3\}}, ..., L^{\{k\}}$
$L^{\{3\}}, L^{\{1\}}, L^{\{2\}}, ..., L^{\{k\}}$
$L^{\{k\}}, L^{\{1\}}, L^{\{2\}}, ..., L^{\{k-1\}}$
Topology construction and selection
$L_t^{\{*\}}$
$L_t^{\{*\}}$
$L_t^{\{*\}}$
$L_t^{\{*\}}$
Hierarchical probability injection
Trained Hierarchical Topology

Fig. 5. Cartesian product label-based hierarchical bottom-up clustering.

Generally, labeled events contain not only temporal information but also spatial details, for instance in tennis, a serve followed by a bounce taking place at the out and far side of the court can be represented by a concatenated descriptor BOFS [24]. Thus, each event label is constructed by incorporating relevant sub-labels providing detailed spatio-temporal information related to game-play, which are crucial for inferring rule structures.

By taking the whole sequence of event labels into account, we can thus represent rule-related information by using the Cartesian combinations of these sub-labels where they collectively constitute a lattice in which coarse-grained event labels are clustered bottom-up to form a hierarchical topology that can potentially represent abstract rule structures.

Thus, various hierarchical label clusters obtained using Cartesian products of sub-labels produce different, but meaningful topological structures that are potentially capable of modeling the underlying abstract structure of the game. This is autonomously achieved by taking all possible permutations of the label order that constitutes these hierarchical structures and, with a predefined selection criterion, a rule-like topological structure is chosen. Methodological details of this method are formulated in the next section and a comparative analysis against other methodologies is conducted in Section VII.

B. Methodology

Sports games have a repetitive rule structure such that a particular sequence of events often repeats during the course of the game-play. For example, a serve followed by a bounce repeats very frequently at the start of every play-shot in tennis where a play-shot is defined as a sequence of events that starts with a serve and ends with the point allocation to one of players in the case of court games [24]. Additionally, game exchanges can also be interpreted via contextual notations such as game transitions between two players. In the middle of a play-shot, they can be represented as rally, or a bounce out of the court area can be represented as a point allocation to either of the players.

Such behavior, with various levels of abstractions, can be modeled using a hierarchical state structure, i.e., hierarchical HMMs. We propose the CLHBC method to generate different hierarchical HMMs capable of producing rule structures representing sports games.

TABLE II

SUMMARY OF BADMINTON AND TENNIS EVENTS [24]

Event	Description
SF$\phi\phi$	Serve by Far player
SN$\phi\phi$	Serve by Near player
HF$\phi\phi$	Hit by Far player
HN$\phi\phi$	Hit by Near player
BIFϕ	Bounce Inside Far player's half court
BOFϕ	Bounce Outside Far player's half court
BINϕ	Bounce Inside Near player's half court
BONϕ	Bounce Outside Near player's half court
BOFS (Tennis only)	Bounce Out of Far player's Serve area
BONS (Tennis only)	Bounce Out of Near player's Serve area

Our input to the system is a set of event labels shown in Table II for badminton and tennis, extracted from [24] and translated into a Cartesian product notation. We argue, more generally, that in most situations complex labeling scenarios can be treated in this fashion (usually with the proviso that we can introduce a null value, ϕ, where there exists incomplete factorizability of the labels intrinsically). Labels are thus constituted of various sub-labels which can represent event types, Ω_E, distance from the camera, Ω_D, sides of the court area Ω_S, and position with respect to the court lines Ω_P, and so on

$$\Omega_E = \begin{Bmatrix} S \\ H \\ B \end{Bmatrix}, \Omega_D = \begin{Bmatrix} N \\ F \\ \phi \end{Bmatrix}, \Omega_S = \begin{Bmatrix} L \\ R \\ \phi \end{Bmatrix}, \Omega_P = \begin{Bmatrix} I \\ O \\ \phi \end{Bmatrix} \tag{1}$$

where ϕ is the null value in the argument description (henceforth, we shall omit this).

To train the model, we divide the stream of input event labels into groups of play-shots (that starts with a serve and ends with a point allocated to either player), for example $SN\phi\phi \rightarrow HF\phi\phi \rightarrow HN\phi\phi \rightarrow HF\phi\phi \rightarrow HN\phi\phi \rightarrow BIF\phi$.

There are repeated sequences within almost every play-shot ($HF\phi\phi \rightarrow HN\phi\phi$ is repeated twice in the example above). These can potentially form hidden states representing common meta-labels, on the next hierarchical level. The method achieves this by combining labels in a manner similar to an explicitly hierarchical Lempel-Ziv-Welch (LZW) encoding [46], i.e., common labels are combined together sequentially to form parent nodes of the hierarchy, for example, different types of serves (*SN* and *SF*) can be combined to represent a parent node labeled *S*, representing the *Serve* meta-label (see Fig. 6). Similarly, another combination (by changing label order) can also be formed combining the *N* and *F* meta-labels to form two separate nodes at the parent level that consequently shall represent game transitions between the *Near* and *Far* side of the court.

These Cartesian meta-labels form the parent level nodes, clustering sets of un-omitted labels beneath it. For example, the string above in terms of event type labels, Ω_E (achieved via the omission of Ω_D labels) looks like $S \rightarrow H \rightarrow H \rightarrow H \rightarrow H \rightarrow BI$.

Play-shots can be represented in the form of other Cartesian label type subsets by changing the label order. Fig. 5 shows a block diagram representing the CLHBC method in context.

Input events sequence contains individual labels, L_t, describing an event at time t, and constituted of z label components drawn from Ω_i, such that

$$\Omega_1 = \begin{Bmatrix} \omega_1^1 \\ \omega_1^2 \\ \omega_1^3 \\ \vdots \\ \phi \end{Bmatrix}, \Omega_2 = \begin{Bmatrix} \omega_2^1 \\ \omega_2^2 \\ \omega_2^3 \\ \vdots \\ \phi \end{Bmatrix}, \Omega_3 = \begin{Bmatrix} \omega_3^1 \\ \omega_3^2 \\ \omega_3^3 \\ \vdots \\ \phi \end{Bmatrix}, \ldots, \Omega_z = \begin{Bmatrix} \omega_z^1 \\ \omega_z^2 \\ \omega_z^3 \\ \vdots \\ \phi \end{Bmatrix}. \tag{2}$$

So, at any given time t, L_t represents the Cartesian products of all the Ω_z labels at each instant in the sequence, defining the base of a z-dimensional lattice (i.e., the lattice formed from differing subsets of Ω labels)

$$L_t \in \{\Omega_1 \times \Omega_2 \times \Omega_3 \times \ldots \times \Omega_z\}^t. \tag{3}$$

Thus, various Cartesian combinations can be formed within the lattice by progressively omitting Ω_z labels, such that, for example

$$L_t^{\{k\}} \in \{\Omega_1 \times \ldots \times \Omega_{k-1} \times \Omega_{k+1} \times \ldots \times \Omega_z\}^t \tag{4}$$

where the omitted label set $k \subseteq \{1, 2, 3, ..., z\}$.

Hence, $L_t^{\{2\}}$, with $z = 3$, represents Cartesian combination of all of the three labels with the exception of Ω_2 that is

$$L_t^{\{2\}} \in \{\Omega_1 \times \Omega_3\}^t. \tag{5}$$

$L_t^{\{k\}}$ is thus composed of a sequence of ordered pairs, $l_i : i = 1, \ldots, t$, derived from the remaining ω labels, such that, in this form, a particular event might look like

$$l_i^{\{2\}} = (\omega_1^2, \omega_3^1). \tag{6}$$

However, note that because label omission is carried out sequentially, not all of the hierarchies within the lattice space are sampled; in fact only a unique hierarchical subset is selected for a particular input label ordering.

Topology selection criterion: Before sampling the resultant hierarchical structure, we repeat the hierarchy generation process above under different orderings, i.e., L_t is represented via other permutations of Ω_i. In the case of the example sequence above, SN can be represented as NS and so on (omitting ϕ for simplicity). This results in various other hierarchies which may or may not approximate the domain rules. For this purpose, a selection criterion is introduced via counting the number of nodes with non-mono child nodes (excluding the leaf nodes). Resultant hierarchies are ranked according to this criterion, for example, Fig. 6 has a rank of 4 and a differently ordered near-far model has a rank of 3. The hierarchical topology with the highest rank is selected for training by sampling the space of transition probabilities in the hierarchy i.e., by explicitly modeling hierarchical transitions (explained in the next section).

Note that usually a human annotator implicitly follows a certain label order (typically general–to–specific) that results in a particular form of rule structure. In case of the tennis/badminton games, the label order followed (see Table II) contains an implicit rule structure that results in the topology shown in Fig. 6. In order to generalize the method's capability and assuming no prior knowledge about label order, a selection criterion that explores all label permutations can autonomously choose a richer rule structure.

Modeling hierarchical transitions: In the following analysis, we will model transitions within the lattice hierarchy (chosen with the criterion above) on a Markovian basis. However, this means that the model as a whole is not consistent with the Markov property (the higher level hidden hierarchical state transitions effectively constitute a memory). It is, though, still possible to represent the entire hierarchy as an implicitly Markovian model. This differs from the standard flat Markovian in which P_f represents a transition likelihood between states Q_{n-1} and Q_n, derived by histogramming over components of an observed sequence (or set of sequences), $S(j)$, $j = 1, \ldots, T$, that is

$$P_f(Q_n|Q_{n-1}) = \frac{1}{\text{F̧}} \sum_{j=1}^{T-1} f(S(j-1), S(j)) \tag{7}$$

where $f = \begin{cases} 1 \; S(j-1) = Q_{n-1}, \;\; S(j) = Q_n \\ 0 \text{ otherwise} \end{cases}$ and F̧ represents the normalization factor. In the following analysis this flat model will serve as our baseline.

We define this implicitly Markovian model as follows. In a z-dimensional CLHBC-generated lattice space, q levels can be formed (depending on the Cartesian combinations) where $q \leq z$, such that a resultant augmented likelihood $\overset{C}{\wedge}$, of event transitions can be computed by considering transitions at all the levels of the constructed hierarchy. The concept of augmented likelihood centers on the modification of observed event likelihoods in order to explicitly favor hierarchicality (i.e., by sampling events at all the levels of the hierarchy).

We introduce a bijective mapping of the constructed hierarchy's leaf states to observations which we use to compute transition likelihood between observations $\mathscr{E}_{X-1}$ to $\mathscr{E}_X$; $X = 1, 2, 3, ..., G$ where G is the total number of leaf nodes (Fig. 6 has $G = 8$). This is achieved using the normalized products of all the super-lying parent state transitions via connected nodes resulting in augmented likelihood of state transitions

$$\overset{C}{\bigwedge}(\mathscr{E}_X|\mathscr{E}_{X-1}) = \frac{1}{\text{Ç}} \prod_{h=1}^{q} \{\frac{1}{\text{Ņ}} \sum_{i=1}^{T} g(S_h(i-1), S_h(i))\} \tag{8}$$

where

$$g = \begin{cases} 1 \; S_h(i-1) = Q_{n-1}^h, \;\; S_h(i) = Q_n^h \\ 0 \text{ otherwise} \end{cases}$$

Q_n^h is the observed state at level h of the hierarchy (i.e., under progressive label omission); Ç is a Cartesian normalization factor and Ņ is the level-based normalization factor. The hierarchical probability injection step computes the augmented likelihood (Fig. 5). Probabilities in the hierarchy are computed top-down and injected per level based on (8) resulting in a single matrix representation of observation state transitions that are bijectively mapped onto the bottom level leaf nodes.

Note, the label space at the bottom level of the hHMM needs to be fully sampled by the data i.e., such that at least a single instance of each label has been observed (however, there are no such restriction higher up in the hierarchy). We have not directly distinguished label uncertainty from state uncertainty, since the latter is fully capable of modeling the former.

The Markovian model thus defined differs from the flat model in that transition likelihoods for observed states are biased by progressively higher-level hidden state transitions, for which there exist better sample-statistics (due to coarser-grained transition likelihoods). We thus influence low-level, rapidly-changing, potentially more noise-influenced transitions by higher-level, more slowly-transitioning states. Consequently, we retain all of the advantages associated with the Markov assumption (in particular, the ability to rapidly model sequence likelihoods via transition matrices), while leveraging the descriptive potential of hierarchical modeling.

1) *Worked Example:* Consider an example sequence of events, with $z = 3$ types of labels
$\mathscr{E} = l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4 \rightarrow l_5 \rightarrow l_6$.

A 3-D lattice of labels is formed. Each event label l_t may look like $(\omega_1^1, \omega_2^1, \omega_3^1)$. After analyzing the whole sequence above l_1 to l_6, different common combinations are extracted at the sub-label level. For example, in tennis or badminton, a sequence of hits can be combined to produce a hidden state semantically equivalent to a rally [these sub-labels are identified and decomposed into a series of Ω_i represented in (2) and (3)]
$(\omega_1^1, \omega_2^1, \phi) \rightarrow (\omega_1^2, \omega_2^2, \phi) \rightarrow (\omega_1^2, \omega_2^1, \phi) \rightarrow (\omega_1^2, \omega_2^2, \phi) \rightarrow (\omega_1^2, \omega_2^1, \phi) \rightarrow (\omega_1^3, \omega_2^1, \omega_3^1)$.

The above sequence can also be represented in its $\{k\} = \{2, 3\}$ sub-label form [see (4)] as
$(\omega_1^1) \rightarrow (\omega_1^2) \rightarrow (\omega_1^2) \rightarrow (\omega_1^2) \rightarrow (\omega_1^2) \rightarrow (\omega_1^3)$.

Common sequential sub-labels are thus extracted as a meta-label that constitutes a node in the next highest level. In this example, three nodes are formed for the sequence such that the augmented likelihood $\overset{c}{\wedge}$ of event transitions [see (8)] can be computed with $q = 3$ representing the number of labels and resultant levels, $G = 4$, and $T = 6$.

In applying the CLHBC model to a badminton game, we find that Cartesian labeling can split the labeled sequential data into various categories of play shot sequences demonstrating the applicability of the method with regards to the label structure of events, for example, we autonomously combine labels according to event types (serves, hits, and so on). In Fig. 6, an example of the bottom-up labeling with colors indicating hereditary of states is shown. Events are delineated in accordance with the play structure by combining starts, rallies and ends together, in turn constituted by serve, hit and bounce meta-states, respectively. The two transition matrices represent non-zero transition probabilities at each level of the hierarchy (using badminton as an example).

V. Multilevel Chinese Takeaway Process

A. Introduction and Motivation

As discussed in Section IV-A, court-games are inherently hierarchical in nature and we attempt to create stochastic approximations of the game rules using hHMM for contextual game description covering various levels of abstractions, ultimately, giving rise to meaningful annotations. As explained in Section IV-B, our input observations are a set of events that occur over a temporal sequence marked when a particular event starts happening (Table II).

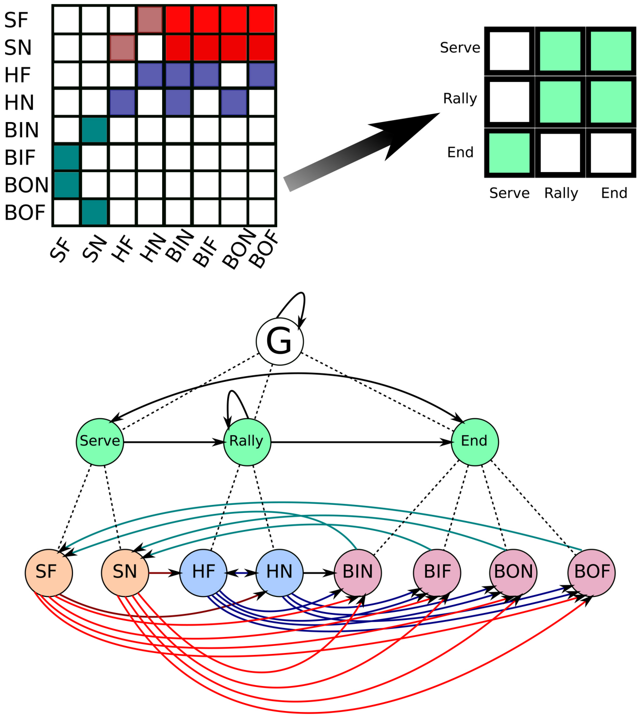


Fig. 6. Three level CLHBC with transition matrices generated at each level (colored so as to indicate common heredity).

In a rule based environment, these events contribute meaningful attributes on a contextual level. Thus, events like serves and hits relate to player's actions while near and far correspond to court locations. Events can be either described in terms of player's actions or rule-defined combinations such as rallies and game points.

To build a generic hHMM framework suitable for characterizing such environments, we propose a constrained variant of the widely used CRP first introduced in [1], which allows us to establish rule structures that are capable of describing sports games in a compact and efficient fashion. The proposed method does not intrinsically exploit labeled information (unlike CLHBC of Section IV) making it more suitable for applications with limited metadata.

We refer to it as the MLCTP, and in the next section we explore the methodological details of this particular variant of the classical CRP and its application to rule-based environments (i.e., sports games).

B. Methodology

The CRP is a non-parametric stochastic process that is naturally capable of representing grouped sequential data. In a rule-based environment, data can be grouped together in a hierarchy and thus we require a hierarchical CRP for stochastic approximation of rules induced via input observations. CRP's hierarchical version is referred to as the Chinese restaurant franchise (CRF) first coined by Teh *et al.* [43]. Due to the limited state-space hierarchy of sport rule structures evident from the types and number of events, it is desirable to implement a hierarchical, but also constrained, version of the classical CRP which we call the MLCTP. To understand this particular variant of the CRP, we step-wise explore the methodological details of MLCTP. To intuitively understand the process, we make use of an analogy similar to the CRP [1].

There are three main methodological steps in generating hierarchical topologies (i.e., rule structures): 1) the state generation phase; 2) transition probabilities generation phase; and 3) the hierarchical state transition matrices injection phase.

1) *State Generation Phase:* This phase is similar to CRP where the number of states is defined by the process via the number of tables. The notion of tables in MLCTP is replaced with takeaways to leverage revisits and further recommendations to other takeaways (further explained in Section V-B2). For the sake of consistency, we replace the notion of tables with takeaways in the first phase.

To start the process, people (tokens) enter a city with infinite number of takeaways and choose a particular takeaway to visit. First person visits the first takeaway in the city with the initial probability equal to 1. The takeaway visit probability, v_i for the ith person is thus defined as

$$P(v_i = c|v_{1:i-1})= \begin{cases} \frac{o_c}{i-1+\alpha} & \text{if } c \leq \mathscr{C} \\ \frac{\alpha}{i-1+\alpha} & \text{otherwise } c \text{ is the new takeaway} \end{cases} \tag{9}$$

where o_c is the number of people who have visited the takeaway c. $\mathscr{C}$ is the number of takeaways for which $o_c > 0$, i.e., visited. α is the concentration parameter. Intuitively, high α implies more visited takeaways with fewer customers.

We initialize the process of state generation assuming one top level state. We henceforth call the top level the first level. For the second level, we follow the takeaway visit process expressed in (9), and generate this level with $\mathscr{C}^2$ states defined by α. For each state at this level, (9) is followed recursively to generate the third level, where a total number $\mathscr{C}^3$ states are created and so on. The process continues until the maximum truncation point is reached, which is defined by the number of event types in the training dataset. Note that $\mathscr{C}^H > \mathscr{C}^{H-1} > \ldots\mathscr{C}^2 > \mathscr{C}^1$, where H represents the total number of levels, i.e., a hierarchy is formed.

At the end of this phase, we thus establish a hierarchical topology with states generated top-down with vertical edges (i.e., representing connections not transitions). Note that this phase is precisely controlled, based on the number of events. As such, as soon as the number of states generated by CRP in the next level to be generated exceeds the termination criterion, the process halts and a new topology is generated. Otherwise, the process continues and, if matched, the process proceeds to Phase 2, the transition probability generation phase. An example topology is shown in Fig. 7.

2) *Topological State Transition Matrix Generation Phase:* This second step for generating the state transition matrix involves two major sub-steps; firstly we extract state transition probabilities, defined by (9) for all the levels. We define each takeaway visit—self transition probability—as ${}_{h'}\delta^h_{i_h}$ for state number i_h at hth level with h' its mother state

$${}_{h'}\delta^h_{i_h} = \frac{\text{Total number of visits to takeaway } i_h}{\text{Total number of visits via } h'}. \tag{10}$$

The remaining probability of transition, ${}_{h'}\psi^h_{i_h} = (1 - {}_{h'}\delta^l_{i_h})$ from the i_hth state to all the other states at level h is further redistributed by executing a stick-breaking construction as follows. We use hyper-parameter γ for all the states controlling the redistribution of the state transitions. This can be intuitively represented by replacing tables in CRP with takeaways where people are recommended $\mathscr{C}^h - 1$ other takeaways to try additionally in city h.

We start with the stick of length 1. The stick is broken $(\mathscr{C}^h - 2)$ times to create $(\mathscr{C}^h - 1)$ partitions representing all other transitions where $\mathscr{C}^h$ represents the total number of takeaways/states at level h. Equation (11) represents the stick-breaking construction weights

$${}^{i_h}\pi^h_k = {}^{i_h}\beta^h_k \prod_{j=1}^{\mathscr{C}^h-2} (1 - {}^{i_h}\beta^h_j) \tag{11}$$

and, the final weight is defined (due to finite states) as

$${}^{i_h}\pi^h_{\mathscr{C}^h-1} = \sum_{c=1}^{\infty} {}^{i_h}\pi^h_c - (\sum_{1}^{\mathscr{C}^h-2} {}^{i_h}\pi^h_k) \text{ where } \sum_{c=1}^{\infty} {}^{i_h}\pi^h_c = 1 \tag{12}$$

${}^{i_h}\pi^h_k$ represents kth weight at level h for state i_h

$${}^{i_h}\beta^h_k \sim \mathit{Beta}(1, \gamma). \tag{13}$$

Within the levels so generated, ${}_{h'}\psi^h_{i_h}$ is partitioned in the manner indicated and transitions to all the other states (left to right indexed) are represented with weights ${}^*\pi^*_1\psi^*_*, {}^*\pi^*_2\psi^*_*, {}^*\pi^*_3\psi^*_*, \ldots,$ and so on.

For each level, h, state transition matrix is built using self-transitions [i.e., the takeaway visit probability of (9)] and the remaining probability is distributed across all the other states at level h, using the stick-breaking construction.

While, in principal, this phase can be defined using another CRP implementation, the use of the SB construction within a CRP generated state structure is most natural in this context, given that the classical CRP defines the probability of table occupancy rather than the probability of transiting from one table to another. Moreover, in such representation, dynamic revisits can also be straightforwardly enabled by expanding the standard SB construction steps from $(\mathscr{C}^h - 2)$ (for level h) to $((\sum \mathscr{C}^h) - 2)$ (for all levels) replacing intralevel connectors with directed edges (with probabilities of transition).

Hyper-parameters: MLCTP is governed by two hyper-parameters α and γ. α controls the number of new states at each level as employed in the classical CRP model. Additionally, α also defines the self-transition probability for each state. The second hyper-parameter γ, is employed in the Beta distribution of (13) and controls the size of the stick-break defined in (11), which furthermore defines the contribution of the remaining probability.

Various combinations of α and γ hyper-parameters are used to generate different types of topologies with varying transition probabilities. Topologies are generated via a greedy uniform sampling distribution across the whole range of the α and γ hyper-parameters between 0 and 1 with incremental steps of 0.1 in each hyperparameter. 1000 topologies are then generated from each of these samples, with a further filtration applied

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

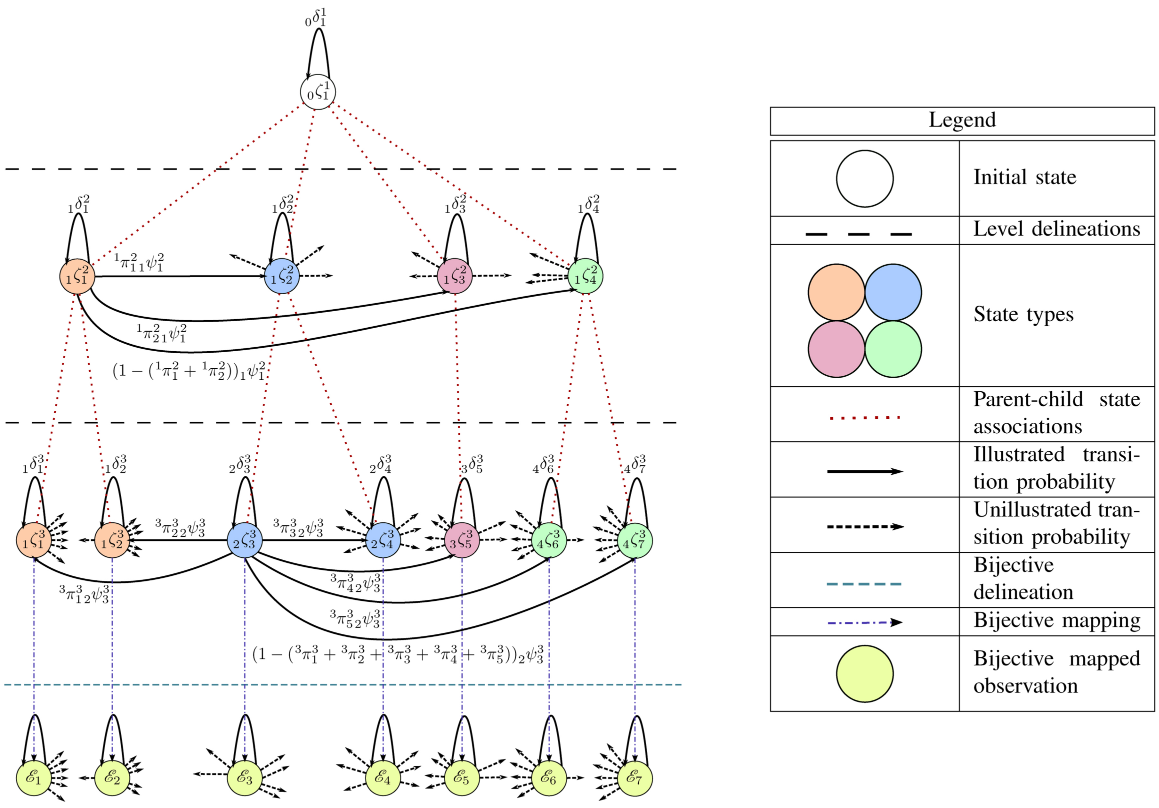

${}_0\delta_1^1$
${}_0\zeta_1^1$
${}_1\pi_{1\,1}^2\psi_1^2$
${}^1\pi_{2\,1}^2\psi_1^2$
$(1-({}^1\pi_1^2+{}^1\pi_2^2))_1\psi_1^2$
${}^3\pi_{2\,2}^3\psi_3^3$
${}^3\pi_{3\,2}^3\psi_3^3$
${}^3\pi_{1\,2}^3\psi_3^3$
${}^3\pi_{4\,2}^3\psi_3^3$
${}^3\pi_{5\,2}^3\psi_3^3$
$(1-({}^3\pi_1^3+{}^3\pi_2^3+{}^3\pi_3^3+{}^3\pi_4^3+{}^3\pi_5^3))_2\psi_3^3$
$\mathscr{E}_1$ $\mathscr{E}_2$ $\mathscr{E}_3$ $\mathscr{E}_4$ $\mathscr{E}_5$ $\mathscr{E}_6$ $\mathscr{E}_7$
Legend
Initial state
Level delineations
State types
Parent-child state associations
Illustrated transition probability
Unillustrated transition probability
Bijective delineation
Bijective mapping
Bijective mapped observation

Fig. 7. MLCTP example topology with $H = 3$ and $\mathscr{G} = 7$ (i.e., $\mathscr{O} = 1, 2, 3, \ldots, 7$).

based on equivalence of the leaf node count to the number of symbols within the training data.

3) *Hierarchical State Transition Matrix Injection Phase:* The next step is to form a state transition matrix for the whole topological structure. We do that by first forming state transition matrices for each level using all the state transitions extracted in Section V-B2 and then use the notion of *probability injection* introduced in Section IV-B. Equation (8) is employed again to represent the augmented likelihood of transitions between all the leaf states.

Note, the bottom-level states are associated with the input number of labels, i.e., we introduce bijective mapping (similar to Section IV-B for CLHBC model) of leaf states to observations that we use to compute the transition likelihood between observations $\mathscr{E}_{\mathscr{O}-1}$ to $\mathscr{E}_{\mathscr{O}}$; $\mathscr{O} = 1, 2, 3, \ldots, \mathscr{G}$ (see Fig. 7 where $\mathscr{G} = 7$). This is achieved using the normalized products of all the super-lying parent state transitions via connected nodes such that

$$\bigwedge^{U}(\mathscr{E}_{\mathscr{O}}|\mathscr{E}_{\mathscr{O}-1}) = \frac{1}{\text{Ḑ}}\prod_{V=H}^{1} P({}_x\zeta_y^V(\mathscr{O})|_{x'}\zeta_{y'}^V(\mathscr{O}-1)) \qquad (14)$$

where $\overset{U}{\wedge}(\mathscr{E}_{\mathscr{O}}|\mathscr{E}_{\mathscr{O}-1})$ is the augmented likelihood for MLCTP generated state transition between events, $\mathscr{E}_{\mathscr{O}-1}$ and $\mathscr{E}_{\mathscr{O}}$. Ḑ is the normalization constant and H is the total number of levels. ${}_x\zeta_y^V$ represents state ζ, indexed by y, with its parent state x and is at level V, where $V = H, H-1, \ldots, 1$, for an input observation index $\mathscr{O}$.

Generated topologies have some generic properties such as: i) each child state has a unique parent but each parent can have one or more than one child state such that a state, represented by ${}_x\zeta_y^h$ has only one x for each y at level h, i.e., x is unique for all y at h and ii) transitions between two child states at level h of a single parent state represents self-transition at level $h+1$ of the corresponding parent state, that is

$$P({}_x\zeta_{y+1}^h|{}_x\zeta_y^h) \Rightarrow P({}_{x'}\zeta_x^{h+1}|{}_{x'}\zeta_x^{h+1}). \qquad (15)$$

Note that the main difference between CLHBC and MLCTP lies in the construction of the rule structure. As such CLHBC is label based with probabilities computed using observations directly, whereas in MLCTP this is achieved using recursive CRPs per state per level until truncation is reached, and SB-construction is then used for calculating topological transition probabilities. The hierarchical probability injection step for calculating augmented likelihoods of state transitions for both of these methods is similar.

4) *Worked Example:* In this section, we initially present the topology construction process for a three-level ($L = 3$), topological structure, i.e., where people visit takeaways in three cities. We also present the topological states' transition matrices generation phase where people are recommended to visit takeaways in the same city including revisiting the same takeaway (representing transition to other states and a self-transition, respectively).

Following is the step-wise instantiation of the process.

Step 1: We begin the process by assuming that the top most level ($h = 1$) has a single state and that the self-transition probability for this state is unity

$$P({}_0\zeta_1^1|{}_0\zeta_1^1) = {}_0\delta_1^1 = 1 \qquad (16)$$

where ${}_x\zeta_y^h$ represents state ζ number y, under the mother state number x. For the top level, represented by (16), $x = 0$ representing no mother node, $y = 1$ the one and only state number and $h = 1$ (the top level index). Intuitively, this level represents city number, $h = 1$, with one take away, $y = 1$ and has no prior recommendations, $x = 0$.

${}_0\delta_1^1$ represents the self-transition probability for the top most state shown in Fig. 7. The state transition matrix for this level is a single number representing the self-transition.

Step 2: In this step, the first instantiation of the MLCTP takes place as formulated in (9) and (SB_Construct). The number of resultant generations (takeaways) represents the number of states under the mother node from Step 1. Analogically, people who have visited the takeaway ${}_x\zeta_y^h$ in city, $h = 1$, are recommended takeaways in city, $h = 2$ (note that as we shall see, it is not always the case that $h = x + 1$).

$\mathscr{C}^2$, representing the number of takeaways in the second city ($h = 2$), in our example is 4 (i.e., $y = [1, 2, 3, 4]$) with the same mother node (i.e., $x = 1$). Their self-transition probabilities, ${}_x\delta_y^2$ i.e., the probability of visiting the same takeaway the next day, are defined in (9).

The remaining probability is broken $\mathscr{C}^2 - 2$ times (i.e., 2 times in this example), so as to generate transitions to all the other states, i.e., the probability of visiting another takeaway, the next day, having visited the current takeaway. This is achieved via the stick-breaking construction of (11) and (13), and is repeated for all the 4 takeaways. Fig. 7 shows all the possible transitions for the first state at the second level (${}_1\zeta_1^2$)

$$P({}_1\zeta_1^2|{}_1\zeta_1^2) = {}_1\delta_1^2 \tag{17}$$
$$P({}_1\zeta_2^2|{}_1\zeta_1^2) = {}^1\pi_1^2(1 - {}_1\delta_1^2) = {}^1\pi_1^2{}_1\psi_1^2 \tag{18}$$
$$P({}_1\zeta_3^2|{}_1\zeta_1^2) = {}^1\pi_2^2{}_1\psi_1^2 \tag{19}$$
$$P({}_1\zeta_4^2|{}_1\zeta_1^2) = (1 - ({}^1\pi_1^2 + {}^1\pi_2^2)){}_1\psi_1^2. \tag{20}$$

Similarly, these transition probabilities are calculated for ${}_1\zeta_2^2$, ${}_1\zeta_3^2$ and ${}_1\zeta_4^2$. The resultant state transition matrix for this example is a 4 × 4 matrix with 16 possible transitions.

Step 3: In this step, we generate new states via another instantiation of the MLCTP for each state at level 2. We do this for all $\mathscr{C}^2$-states generated in Step 2. The number of generations represents the number of states under each mother node. Analogically, people who have visited takeaways in city 2, are recommended to visit related takeaways in city 3.

x at this level is the total number of states indexed by y in the previous level in Step 2 representing the now-mother states while the length of y is determined by each instantiation of MLCTP for every x. The number of MLCTP instantiations is equal to the length of x.

Thus, at this level, $h = 3$, $x = [1, 2, 3, ..., \mathscr{C}^2]$, and $y = [1, 2, 3, ..., \mathscr{C}^3]$, which is constituted via the tuple y'

$$y' = \begin{pmatrix} \{1, 2, 3, ..., \mathscr{C}'_1\}, \\ \{1, 2, 3, ..., \mathscr{C}'_2\}, \\ \{1, 2, 3, ..., \mathscr{C}'_3\}, \\ \vdots \\ \{1, 2, 3, ..., \mathscr{C}'_{\mathscr{C}^2}\} \end{pmatrix} \text{ where } \sum_{r=1}^{\mathscr{C}^2} \mathscr{C}'_r = \mathscr{C}^3.$$

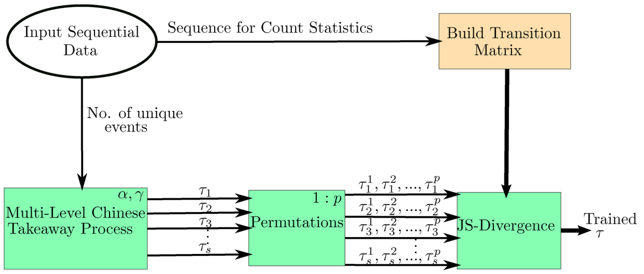

Input Sequential Data
Sequence for Count Statistics
Build Transition Matrix
No. of unique events
α, γ
Multi-Level Chinese Takeaway Process
τ_1
τ_2
τ_3
τ_s
$1 : p$
Permutations
$\tau_1^1, \tau_1^2, ..., \tau_1^p$
$\tau_2^1, \tau_2^2, ..., \tau_2^p$
$\tau_3^1, \tau_3^2, ..., \tau_3^p$
$\tau_s^1, \tau_s^2, ..., \tau_s^p$
JS-Divergence
Trained τ

Fig. 8. MLCTP.

In our example, at the lowest level, $x = [1, 2, 3, 4]$, $y = [1, 2, 3, 4, 5, 6, 7]$, constituted via the tuple $y' = (\{1, 2\}, \{1, 2\}, \{1\}, \{1, 2\})$.

Similar to Step 2, the remaining probability is broken $\mathscr{C}^3 - 2$ times (i.e., five times in this example), to generate transitions to all the other states, i.e., the probability of visiting another takeaway, the next day, having visited the current takeaway. This is achieved via stick-breaking construction of (11), and is repeated for all the seven takeaways.

Fig. 7 shows all the possible transitions for the third state at the level 3 under the second state of level 2, i.e., ${}_2\zeta_3^3$

$$P({}_2\zeta_3^3|{}_2\zeta_3^3) = {}_2\delta_3^3 \tag{21}$$
$$P({}_1\zeta_1^3|{}_2\zeta_3^3) = {}^3\pi_1^3.(1 - {}_2\delta_3^3) = {}^3\pi_1^3.{}_2\psi_3^3 \tag{22}$$
$$P({}_1\zeta_2^3|{}_2\zeta_3^3) = {}^3\pi_2^3.{}_2\psi_3^3 \tag{23}$$
$$P({}_2\zeta_4^3|{}_2\zeta_3^3) = {}^3\pi_3^3.{}_2\psi_3^3 \tag{24}$$
$$P({}_3\zeta_5^3|{}_2\zeta_3^3) = {}^3\pi_4^3.{}_2\psi_3^3 \tag{25}$$
$$P({}_4\zeta_6^3|{}_2\zeta_3^3) = {}^3\pi_5^3.{}_2\psi_3^3 \tag{26}$$
$$P({}_4\zeta_7^3|{}_2\zeta_3^3) = (1 - ({}^3\pi_1^3 + {}^3\pi_2^3 + {}^3\pi_3^3 + {}^3\pi_4^3 + {}^3\pi_5^3)).{}_2\psi_3^3. \tag{27}$$

C. Induction Protocol

Fig. 8 shows the block diagram of the experimental protocol for the MLCTP showing the training process. MLCTP is a stochastic process, and to counter the issue of stochastic variations we first generate R topologies, i.e., we execute the process R times given the hyper-parameters α and γ. The total number of selected topologies according to the truncation parameter $\mathscr{G}$ (applied such that when the exact number of leaf states is achieved the process stops and emits a topological structure), is s, where $s \leq R$. These s topologies are represented as transition matrices computed via (14), and each transition matrix goes through a selection process for the best fit as the rule defining topology. This is achieved via measuring the distance between the training matrix (using the count statistics of the training data) and the MLCTP-generated topological transition matrix.

MLCTP is a stochastic and unlabeled process, and thus a topology generated given a set of hyper-parameters does not necessarily correlate with the original training transition matrix. To better sample the topological state space, we generate random permutations at observations level indicated in Fig. 8 (we employ random permutations to reduce the computation time). If b is the state-space defined by the number of leaf states and p is the number of random permutations then $p \leq b!$. Each topology, τ_I ($I = 1, 2, 3, ..., s$), is expanded to p random permutations within the state-space, where τ_1 represents the first selected topology matrix and τ_s represents the last selected topology matrix. We thus have a set $\{\tau_1^1, ..., \tau_1^p, \tau_2^1, ..., \tau_2^p, \tau_3^1, ..., \tau_3^p, ..., \tau_s^1, ..., \tau_s^p\}$.

Each of these topological transition matrices (total $s \times p$) are then compared against the flat transition matrix built from the training sequence of events. This comparison is performed via the Jensen-Shannon Divergence.

Jensen-Shannon Divergence: We use Jensen-Shannon Divergence [32] to measure the divergence between two probability distributions i.e., the output of permutations block and the flat Markov model block in Fig. 8. JS-Divergence is based on the Kullback-Leibler divergence, and is defined as the average relative entropy of the source distributions to the entropy of the average distribution. Equation 28 represents the metric Y employed in Fig. 8 for MLCTP's topological transition matrices J_Z (where $Z = 1, 2, 3, ..., s \times p$) and the training transition matrix J_{tr}

$$Y(J_{tr}, J_Z) = \frac{1}{2}\big(KL(J_{tr} \mid\mid K) + KL(J_Z \mid\mid K)\big) \tag{28}$$

where K is the average distribution of the two sources, that is

$$K = \frac{1}{2}(J_{tr} + J_Z) \tag{29}$$

and the KL-divergence can be defined between two vectors, M_1 and M_2 as

$$KL(M_1 \mid\mid M_2) = \sum_i M_1(i) ln \frac{M_1(i)}{M_2(i)}. \tag{30}$$

Each topological transition matrix (J_Z) from the permutations block is compared against the training transition matrix J_{tr} and the closest topological structure (i.e., one with the smallest Y metric against the training matrix) is taken as the learned hierarchical topology with respect to the input training sequence of events. This trained hierarchical topology is used in the following experimental investigation (of Section VII) for predicting future events based on the input sequence.

VI. Hybrid Models

A. Multilevel Chinese Takeaway Process with Recursive Baum-Welch Estimated State Transitions (MLCTP-BW)

In addition to the above label-based and generative methods, we also propose a pair of hybridized methods suitable for stochastic inference of rule structures in sport videos. The first hybrid model extracts hierarchical structures using MLCTP's topological state generation process shown in Section V-B1. The topological state transition matrix generation phase is ignored in this model, so the hierarchical structure output from MLCTP is effectively just the arrangement of nodes, connected in a hierarchy. These topologies are built top-down and we similarly select topologies based on MLCTP's truncation parameter $\mathscr{G}$ (defined as the number of differentiated states in the input sequential data).

In order to recalculate transition probabilities on the topological edges for the hybrid model, we first compute the count statistics of the sequence of event transitions, i.e., the flat Markov model of (7) that calculates the observed transition probabilities. Leaf states are thus mapped bijectively to observations, effectively leaving the number of states in the higher (i.e., non-observation or hidden) levels to be defined by the MLCTP-generated topological rule structures. This structure is then used to estimate a set of state transition probabilities at each level via recursive Baum-Welch estimation (Fig. 9 has the block diagram representing the training process using this method).

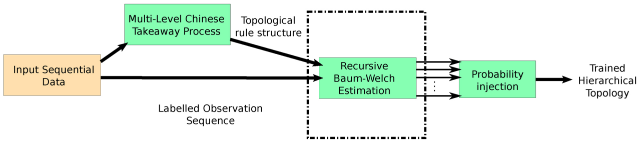

Multi-Level Chinese Takeaway Process
Topological rule structure
Input Sequential Data
Recursive Baum-Welch Estimation
Probability injection
Trained Hierarchical Topology
Labelled Observation Sequence

Fig. 9. MLCTP with Baum-Welch hidden state transition estimation.

We can characterize the model via the following notation. MLCTP emitted topological structure has $h = 1, ..., H$ levels and for each pair of levels, we can specify a set of HMM parameters $\lambda_h = \{a_{ij}^h, e_i^h(.), \eta^h(i)\}$, where $\eta^h(i)$ is the initial distribution of states per level defined by MLCTP, $e_i^h(.)$ is the emission probability for level h, i.e., the probability of state i at level h emitting a symbol at level $h+1$, while the transition probability of a state transiting from i to j for level h is a_{ij}^h

$$a_{ij}^h = P(Q_t^h = j | Q_{t-1}^h = i) \text{ and } \sum_{i=1}^{\mathscr{C}^h} a_{ij} = 1 \quad \forall j \tag{31}$$

where Q_t^h is the current [hidden] state of a temporal sequence as represented at the hierarchical level h.

The input sequence of labeled data is thus the observed sequence from which we obtain the parameters of the model via maximum likelihood estimation. Utilizing the MLCTP-generated hierarchical topology, Baum-Welch algorithm is hence employed recursively to obtain the model parameters when the state path per level is unknown. Thus given a level-based sequence of observations $\{Q_t^{h+1}\}$ for a given number of states defined by MLCTP, $\mathscr{C}^h$, we compute $\lambda_h = \{a_{ij}^h, e_i^h(.), \eta^h(i)\}$. The parameters that maximize the likelihood of the input data are thus chosen at every level

$$A_{ij}^h = BaumWelch(\mathscr{C}^h, \{Q_t^{h+1}\}, \lambda_h) \tag{32}$$

where

$$A_{ij}^h = \frac{1}{P(Q_t^{h+1}|\lambda_h)} \sum_t F(t, i) a_{ij}^h e_j^h(Q_{t+1}^{h+1}) B(t+1, j) \tag{33}$$

where $i, j = 1, 2, ..., \mathscr{C}^h$.

Here, A_{ij}^h is the estimated state-transition probability of state i to j at level h; $F(t, i)$ represents the probability of the model emitting symbols, $Q_1^{h+1} ... Q_T^{h+1}$, when in state i at time t, obtained using the forward algorithm. a_{ij}^h and $e_j^h(Q_{t+1}^{h+1})$ respectively represent the probability of transition from state i to j and emitting the $t+1$st emission symbol at level $h+1$ (both arbitrarily instantiated and recursively updated). The backward algorithm computes $B(t+1, j)$ which is the probability of the

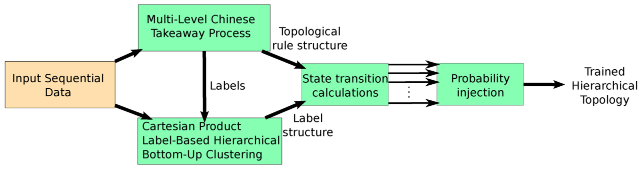

Multi-Level Chinese Takeaway Process
Topological rule structure
Input Sequential Data
Labels
State transition calculations
Probability injection
Trained Hierarchical Topology
Cartesian Product Label-Based Hierarchical Bottom-Up Clustering
Label structure

Fig. 10. MLCTP—CLHBC.

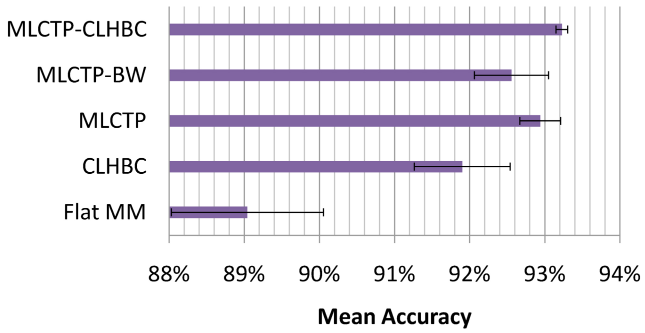


Fig. 11. Badminton Dataset. Mean prediction accuracies.

model emitting the remaining sequence if the model is in state j at time $t + 1$.

Thus, in this model, estimated hidden state transitions act as the observation level for the estimation of the next highest level hidden state transitions in the hierarchy and so on. After estimating state transition probabilities, a state sequence is generated, i.e., $\{Q_t^h\}$, which is used as input observations for the next level of MLCTP's hierarchy.

Finally, after computing state transition probabilities for each level, we perform the top-down hierarchical probability injection step [(14)] to obtain the learned augmented likelihood of events for MLCTP-generated topological structure with recursive BW estimated state transition probabilities.

The MLCTP-BW hybrid could be considered as the methodology that is conceptually closest to the standard hHMM of [12], where the hierarchical topology is fixed based on the hierarchy established using MLCTP.

B. Multilevel Chinese Takeaway Process with Cartesian Product Label-Based Hierarchical Bottom-up Clustering Computed State Transitions (MLCTP-CLHBC)

The second hybrid model variant similarly extracts the hierarchical topologies from MLCTP's topological state generation process. However, transition probabilities at the edges are then computed using the CLHBC method.

In this model, MLCTP determines the number of levels (which in CLHBC is determined by the number of sub-labels defined in Section IV). Each labeled event is replaced by an arbitrary label, comprised of z types of labels, Ω_z. In this hybrid, z for CLHBC is determined by MLCTP. Intuitively, bigger z implies a deeper hierarchy.

Thus the input training sequential data provides the event labels to be bijectively mapped into the observation level of MLCTP generated hierarchy [e.g., $Q_t^1 \rightarrow SF$, i.e., observation Q_t^1 is associated with event label serve far (see Fig. 7)].

Given the number of levels in the MLCTP-generated topological structure ($z = 3$ in Fig. 7), event labels are then replaced with the Cartesian products of z arbitrary labels, i.e., Ω_z where $z = 1, 2, 3$ such that the sub-label factors give rise to a hierarchy equivalent to that defined by MLCTP-generated topological structure. We thus reverse engineer Fig. 7, where each event is rerepresented with three (as $z = 3$) labels, e.g., $\omega_1^1, \omega_3^2, \omega_5^3$, with labeling associated with the observed states such that the common sequential factors result in the hierarchy generated using MLCTP. Following this reassociation phase, the training event sequences are regenerated using the new label-structure, and the CLHBC process executed resulting in state-transition probabilities at each level of the hierarchy generated by MLCTP.

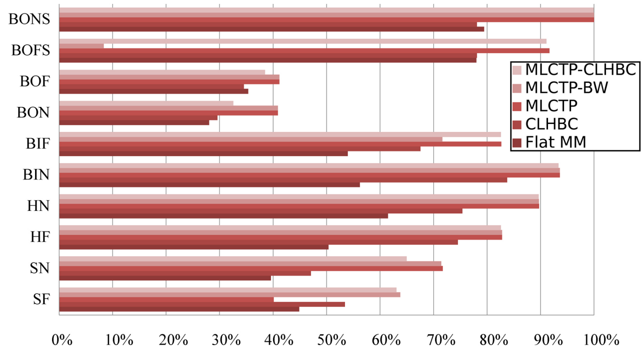


Fig. 12. Tennis dataset (video annotation system)—Individual event prediction accuracies for all of the five methods.

Transition between states is thus governed by the input data at every level; however, interlevel associations are determined by MLCTP. The method thus populates the transition likelihoods bottom up according to the MLCTP template. Fig. 10 shows the block diagram for this hybrid model in which the input sequential data's original labels are replaced with MLCTP-defined arbitrary labels. The hierarchical structure output from MLCTP is combined with this new label hierarchy and used to compute state-transition matrices for each level.

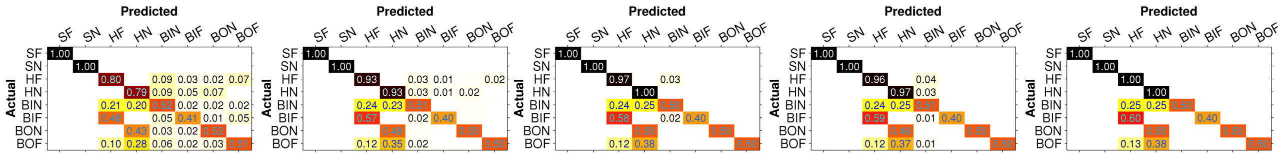
The trained augmented likelihood of events for the MLCTP-generated topological hierarchy and CLHBC formulated label structure is computed in a similar fashion to previous methods via probability injection block of Fig. 10. Experimental results for MLCTP-CLHBC are shown in Section VII.

VII. Experiment Results and Discussions

In this section, we evaluate the performance of all the four proposed variants of the novel hierarchical HMM strategy using six different datasets shown in Tables III and IV.[1] In the case of the Badminton dataset [i.e., badminton mens singles from Beijing Olympics, 2008 (BMSB08) detailed in Table III], we train the models using 77 play-shots (i.e., collections of sequences starting with the event serve and ending with a point-awarding event), and test using the remaining 20 play-shots. The number of unique events for badminton is eight (see Table II). Similarly, datasets from other domains (details shown in Table IV) are also employed.

For experimental evaluation, we measure the prediction accuracy of the next event given all the previous events as

[1] Source code available at http://www.cvssp.org/acasva/Downloads



(a) (b) (c) (d) (e)

Fig. 13. Badminton dataset—Confusion matrices for all of the five methods. (a) Flat MM, (b) CLHBC, (c) MLCTP, (d) MLCTP-BW, and (e) MLCTP-CLHBC.

TABLE III

SPORTS DATASETS WITH SOURCE INFORMATION AND NUMBER OF SAMPLES PER EVENT LABEL

	Source Information					Event Statistics										
Label	Sport	Gender	Type	Competition	Year	Total	SF	SN	HF	HN	BIN	BIF	BON	BOF	BOFS	BONS
BMSB08	Badminton	Mens	Singles	Beijing Olympics	2008	**644**	59	38	222	228	41	17	22	17	-	-
TWSA03	Tennis	Womens	Singles	Australian Open	2003	**532**	32	42	98	92	104	120	11	11	13	9
TWSA03 [25]	Tennis	Womens	Singles	Australian Open	2003	**293**	30	38	35	42	56	50	9	12	12	9
TMSA03	Tennis	Mens	Singles	Australian Open	2003	**1122**	69	53	221	224	234	265	12	22	12	10

TABLE IV

DATASETS DESCRIPTION

	Train Source	Test Source	Area	Train Size	Test Size	No. of Unique Events
Badminton Dataset	BMSO08	BMSO08	Sports	77 Play-shots	20 Play-shots	8
Tennis Dataset	TWSA03	TMSA03	Sports	74 Play-shots	122 Play-shots	10
Tennis (Annotation System [25])	TWSA03	TMSA03	Sports	68 Play-shots	122 Play-shots	10
Human Activity Dataset	UCI Repository [19]	UCI Repository [19]	Life	500 Events	275 Events	11
Human Driving Dataset	EU DIPLECS [47]	EU DIPLECS [47]	Highway Rules	135 Events	80 Events	10
MSNBC.com (Website Dataset)	UCI Repository [4]	UCI Repository [4]	Web	1 → 8000 clicks	8001 → 10000 Clicks	11

TABLE V

MEAN ACCURACY

	Flat	CLHBC	MLCTP	MLCTP-BW	MLCTP-CLHBC
Badminton Dataset	89.04% ± 1.01	91.90% ± 0.64	92.94% ± 0.27	92.56% ± 0.49	93.23% ± 0.08
Tennis Dataset	63.80% ± 1.31	69.27% ± 1.16	73.28% ± 1.16	70.07% ± 0.62	73.43% ± 0.97
Tennis (Annotation System [25])	52.73% ± 2.18	62.20% ± 2.24	73.44% ± 0.54	66.33% ± 0.21	73.80% ± 1.09
Human Activity Dataset	81.34% ± 1.67	82.34% ± 1.57	84.27% ± 1.14	83.26% ± 0.77	84.56% ± 0.78
Human Driving Dataset	55.77% ± 5.25	58.30% ± 3.34	66.75% ± 1.23	65.71% ± 0.93	67.21% ± 3.07
MSNBC.com (Website Dataset)	44.21% ± 1.55	53.53% ± 1.48	60.00% ± 0.06	57.56% ± 1.61	64.17% ± 0.64
Average Performance Gain	-	5.11%	10.63%	8.10%	11.59%

shown in Fig. 12 for the tennis dataset (extracted using the automated video annotation system of [25]). For example, the model CLHBC correctly predicts the next event 74.52% of the time, if the current event is *HF* (Table II), and so on.

Fig. 11 shows the comparative mean accuracies for the badminton dataset using all of the methods employed namely, the flat Markov model, the CLHBC, the MLCTP, Hybrid I (MLCTP-BW), and Hybrid II (MLCTP-CLHBC). Mean prediction accuracies for all of the methods applied to all of the datasets with individual standard deviations are shown in Table V. Associated mean performance gains with respect to the baseline approach are also shown.

We show that all of the proposed hHMM generating methodologies demonstrate improvement relative to the flat Markov model.

Additionally, confusion matrices for the predicted events are also presented for all the methods applied to the badminton dataset in Fig. 13. For example, in Fig. 13(c), *BON* (see Table II) is 50% of the time correctly predicted, while 50% of the time incorrectly predicted as *HN* (hit near).

As may be seen in Fig. 11 and Table V optimal performance for all the datasets is achieved using the hybrid model MLCTP-CLHBC (of Section VI-B). MLCTP-CLHBC hybrid leverages MLCTP's topological rule structure and consequently the label-based CLHBC to construct a rule model that is more accurate. The average performance gain achieved using MLCTP-CLHBC compared with the flat Markov model is 11.59% resulting in a significant improvement.

Relative performance gains in the context of a particular dataset depends upon the complexity of the data. It can be observed that in the case of more complex datasets—such as the website and tennis [25] datasets—the average performance gains achieved by MLCTP-CLHBC are around 20%.

In a different setting (introduced in Section III and explained in [24] and [25]) high-level reasoning, in terms of correctly awarded points, is performed using the hard-wired HMM based on Fig. 3. Correct point recognition rates reported using the two tennis datasets, TWSA03, and TMSA03 (detailed in Table III) were 87.5% and 73.75%, respectively.

It is crucial to highlight again that HMM used in the work of [24] and [25] requires the number of states to be fixed heuristically based on the exact rule of the game. A different domain cannot be directly introduced without manually altering the HMM topology as there is no capability in this framework to learn the rule model. Our generalized rule induction mechanism on the other hand is intrinsically adaptive, as evidenced by our demonstration of the approach in domains other than tennis.

VIII. Conclusion

In this paper, we proposed four variants of the novel hierarchical HMM strategy for rule induction and applied them to the problem of automated sports video annotation. We firstly introduced a CLHBC method that employs the latent structure in the labels used to annotate videos. Labels are thus employed to build hierarchical structures based on various Cartesian Product based combinations of sub-labels such that a hierarchical HMM of common repeated event structures is established (and which is used to evaluate the predictive capability of the method). The second proposed variant, the MLCTP, is based on the CRP with tables replaced by takeaways which may be revisited within different cities representing levels in the hierarchy. This is a stochastic process, with many hierarchies generated for a given set of hyper-parameters, such that a distance measure (JS-Divergence) is employed to infer the highest likelihood stochastic rule structure.

We also introduced two hybrid variants namely MLCTP-BW and MLCTP-CLHBC that leverage the stochasticity of MLCTP (whereby various latent hierarchical structures are produced), in conjunction with the label sequence to give a composite top-down MLCTP-driven (topologically) and bottom-up data-driven approach to hierarchical HMM inference. All of these methods finally generate finite intermediate-depth hierarchical HMMs that are well-suited to calculating the likelihood of event transitions taking place within sport video sequences typically governed by analogous hierarchical rule structures involving, e.g., matches, sets, points, and so on.

We conclude that leveraging the label information contained within sequential data (especially sports sequences) in conjunction with our novel MLCTP provides a previously unexploited opportunity for rule-induction. Comparative prediction results for all the proposed methods are shown relative to the flat Markov model, with all of the hierarchical methods shown to perform better (with the most optimal method being the MLCTP-CLHBC hybrid).

In the context of an automated video annotation system, the rule induction framework thus provides a robust context analysis module in which rules are inferred from observations and predictions are made that can serve as logical priors on detections. Such a framework can be employed for tackling various problems beside prediction generation. In particular, it can address the issue of anomaly detection; when a new domain is introduced to the system architecture, anomalous events (as opposed to outliers and errors) can be detected using the rule hierarchy triggering the domain change. In the context of the automated video annotation system, this may require switching the knowledge base by abandoning continuous adaptive learning and replacing it with a new learning process.

Rule induction framework can also be employed to address the problem of transferring knowledge from one domain to another; this can be achieved via analyzing various levels of the established rule hierarchies representing different levels of abstractions such that in a new (and related) domain, contextual inferences are transferred, i.e., minimizing the need for retraining.

Novel methodologies introduced in this paper have been extensively evaluated in terms of event predictions. They can also be used to predictively and retrospectively identify unobserved rule-based events. For this purpose, current Markovian structure must be replaced via analyzing associated rule-grammars.

In order to expand these models for practical implementation in future generalized video annotation systems, the proof-of-concept evaluations will need to be expanded to other domains comprising other sports such as cricket, football, table tennis and non-sporting domains such as, characterizing surveillance footage, recorded meetings, lectures, and so on.

References

[1] D. J. Aldous, "Exchangeability and related topics," in *Lecture Notes in Mathematics*, vol. 1117. Berlin, Germany: Springer, 1985, pp. 1–198.

[2] I. Almajai, J. Kittler, T. de Campos, W. Christmas, F. Yan, D. Windridge, *et al.*, "Ball event recognition using HMM for automatic tennis annotation," in *Proc. IEEE ICIP*, 2010, pp. 1509–1512.

[3] I. Almajai, F. Yan, T. de Campos, A. Khan, W. Christmas, D. Windridge, *et al.*, "Anomaly detection and knowledge transfer in automatic sports video annotation," in *Detection and Identification of Rare Audiovisual Cues*. Berlin, Germany: Springer, 2012, pp. 109–117.

[4] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of navigation patterns on a web site using model-based clustering," in *Proc. ACM SIGKDD*, 2000, pp. 280–284.

[5] G. S. Chambers, S. Venkatesh, G. West, and H. Bui, "Segmentation of intentional human gestures for sports video annotation," in *Proc. IEEE MMM*, vol. 1, 2004, pp. 124–129.

[6] H. Cooper, B. Holt, and R. Bowden, "Sign language recognition," in *Visual Analysis of Humans*. London, U.K.: Springer, 2011, pp. 539–562.

[7] H. Denman, N. Rea, and A. Kokaram, "Content-based analysis for video from snooker broadcasts," *Elsevier Comput. Vision Image Understand.*, vol. 92, nos. 2–3 pp. 176–195, 2003.

[8] H. Denman, N. Rea, and A. C. Kokaram, "Content based analysis for video from snooker broadcasts," in *Proc. CIVR*, 2002, pp. 198–205.

[9] L. Duan, M. Xu, T. Chua, Q. Tian, and C. Xu, "A mid-level representation framework for semantic sports video analysis," in *Proc. ACM MM*, 2003, pp. 33–44.

[10] A. Ekin, A. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 796–807, Jul. 2003.

[11] J. Ferryman and A. Ellis, "Pets2010: Dataset and challenge," in *Proc. IEEE AVSS*, 2010, pp. 143–150.

[12] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, 1998.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *ACM Commun.*, vol. 24, no. 6, pp. 381–395, 1981.

[14] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan, "Computational studies of human motion: Part 1, tracking and motion synthesis," *Found. Trends Comput. Graph*, vol. 1, nos. 2–3, pp. 77–254, 2006.

[15] Y. Gong, L. Sin, C. Chuan, H. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," in *Proc. ICMCS*, vol. 2, 1995, pp. 167–174.

[16] J. Guillemaut and A. Hilton, "Joint multi-layer segmentation and reconstruction for free-viewpoint video applications," *Comput. Vision*, vol. 93, no. 1, pp. 73–100, 2011.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[18] M. Hassan and B. Nath, "Stockmarket forecasting using hidden Markov model: A new approach," in *Proc. ISDA*, 2005, pp. 192–196.

[19] B. Kaluza, V. Mirchevska, E. Dovgan, M. Lustrek, and M. Gams, "An agent-based approach to care in independent living," in *Ambient Intelligence*, vol. 6439. Heidelberg, Germany: Springer, 2010, pp. 177–186.

[20] A. Khan, D. Windridge, T. de Campos, J. Kittler, and W. Christmas, "Lattice-based anomaly rectification for sport video annotation," in *Proc. IEEE ICPR*, 2010, pp. 4372–4375.

[21] E. Kijak, G. Gravier, L. Oisel, and P. Gros, "Audiovisual integration for tennis broadcast structuring," *Multimedia Tools Appl.*, vol. 30, no. 3, pp. 289–311, 2006.

[22] J. Kittler, W. Christmas, T. de Campos, D. Windridge, F. Yan, J. Illingworth, *et al.*, "Domain anomaly detection in machine perception: A system architecture and taxonomy," *IEEE Trans Pattern Anal. Mach. Intell.*, no. 99, pp. 1-1, 2013.

[23] J. Kivinen, E. Sudderth, and M. Jordan, "Learning multiscale representations of natural scenes using dirichlet processes," in *Proc. ICCV*, 2007, pp. 1–8.

[24] I. Kolonias, "Cognitive vision systems for video understanding and retrieval," Ph.D. thesis, Dept. of CVSSP, Univ. Surrey, Surrey, U.K., 2007.

[25] I. Kolonias, W. Christmas, and J. Kittler, "A layered active memory architecture for cognitive vision systems," in *Proc. ICVS*, 2007.

[26] A. Koski, "Modelling ECG signals with hidden Markov models," *Artif. Intell. Med.*, vol. 8, no. 5, pp. 453–471, 1996.

[27] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Sys. Tech. J.*, vol. 62, no. 4, pp. 1035–1073, 1983.

[28] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. App. Math.*, vol. 11, no. 2, pp. 431–441, 1963.

[29] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. CVPR*, 2009, pp. 2929–2936.

[30] K. Messer, W. J. Christmas, E. Jaser, J. Kittler, B. Levienaise-Obadia, and D. Koubaroulis, "A unified approach to the generation of semantic cues for sports video annotation," *Signal Process.*, vol. 85, no. 2, pp. 357–383, 2005.

[31] T. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vision Image Understand.*, vol. 104, nos. 2–3, pp. 90–127, 2006.

[32] F. Nielsen, "A family of statistical symmetric divergences based on Jensen's inequality," *CoRR*, vol. abs/1009.4004, Sep. 2010.

[33] T. Ogata, W. Christmas, J. Kittler, and S. Ishikawa, "Tennis stroke detection and classification based on boosted activity detectors and particle filtering," in *Proc. SCIS & ISIS*, 2006, pp. 2035–2040.

[34] A. Patron-Perez, M. Marszałek, A. Zisserman, and I. Reid, "High five: Recognising human interactions in TV shows," in *Proc. BMVC*, 2010, pp. 50.1–11.

[35] M. Petkovic, V. Mihajlovic, W. Jonker, and S. Djordjevic-Kajan, "Multimodal extraction of highlights from tv formula 1 programs," in *Proc. IEEE ICME*, vol. 1, 2002, pp. 817–820.

[36] R. Poppe, "A survey on vision-based human action recognition," *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[37] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[38] N. Rea, R. Dahyot, and A. Kokaram, "Classification and representation of semantic content in broadcast tennis videos," in *Proc. IEEE ICIP*, 2005, pp. 1204–1207.

[39] I. Reid and K. Connor, "Multiview segmentation and tracking of dynamic occluding layers," *Image Vision Comput.*, vol. 28, no. 6, pp. 1022–1030, 2009.

[40] I. Reid and A. Zisserman, "Goal-directed video metrology," in *Proc. ECCV*, vol. 2, 1996, pp. 647–658.

[41] M. Roh, W. Christmas, J. Kittler, and S. Lee, "Robust player gesture spotting and recognition in low-resolution sports video," in *Proc. ECCV*, 2006, pp. 347–358.

[42] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statist. Sinica*, vol. 4, pp. 639–650, 1994.

[43] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1566–1581, 2006.

[44] M. Tien, Y. Wang, and C. Chou, "Event detection in tennis matches based on video data mining," in *Proc. IEEE ICME*, 2008, pp. 1477–1480.

[45] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[46] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, 1984.

[47] D. Windridge, A. Shaukat, and E. Hollnagel, "Characterizing driver intention via hierarchical perception action modeling," *Human-Mach. Syst.*, vol. 43, no. 1, pp. 17–31, 2013.

[48] L. Xie, P. Xu, S. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden Markov models," *Pattern Recognit. Lett.*, vol. 25, no. 7, pp. 767–775, 2004.

[49] E. P. Xing and K. Sohn, "Hidden Markov Dirichlet process: Modeling genetic recombination in open ancestral space," *Bayesian Anal.*, vol. 2, no. 3, pp. 501–528, 2007.

[50] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. Huang, *A Unified Framework for Video Summarization, Browsing & Retrieval: With Applications to Consumer and Surveillance Video*. Waltham, MA, USA: Academic Press, 2005.

[51] F. Yan, A. Kostin, W. Christmas, and J. Kittler, "A novel data association algorithm for object tracking in clutter with application to tennis video analysis," in *Proc. CVPR*, vol. 1, 2006, pp. 634–641.

[52] X. Yu, H. W. Leong, J. Lim, Q. Tian, and Z. Jiang, "Team possession analysis for broadcast soccer video based on ball trajectory," in *Proc. PCM*, vol. 3, 2003, pp. 1811–1815.

[53] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing, "Player action recognition in broadcast tennis video with applications to semantic analysis of sports game," in *Proc. ACM MM*, 2006, pp. 431–440.

Aftab Khan received the B.Eng. degree (First-class Hons.) in electronic engineering, in 2008, and the Ph.D. degree in electronic engineering from the Centre for Vision, Speech and Signal Processing, in 2013, from the University of Surrey, Surrey, U.K. He was actively involved in the Engineering and Physical Sciences Research Council (EPSRC) project "Adaptive Cognition for Automated Sports Video Annotation" over the course of the research degree.

He is currently a Research Associate at Newcastle University, Tyne and Wear, U.K., involved in multiple projects including the EPSRC project "Transforming Energy Demand in Buildings through Digital Innovation." He has authored and co-authored scientific articles in various conference and workshop venues. His current research interests include topics in machine learning, artificial intelligence, decision systems, and pattern recognition.

David Windridge is a Senior Research Fellow at the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Surrey, U.K. His research interests include pattern-recognition and cognitive systems, having authored and played a leading role on a range of grants including, most recently, EPSRC ACASVA and EU FP7 DIPLECS. He received the B.Sc. (Hons.) degree in physics in 1993, the M.Sc. degree in radio-astronomy in 1995, and the Ph.D. degree in statistical cosmology from the University of Bristol, Bristol, U.K., in 1999. He has authored more than 70 peer-reviewed publications.

Josef Kittler is currently a Professor of Machine Intelligence at the Centre for Vision, Speech and Signal Processing at the University of Surrey, Surrey, U.K. He conducts research with a focus on biometrics, video and image database retrieval, automatic inspection, medical data analysis, and cognitive vision. He has authored a textbook titled *Pattern Recognition: A Statistical Approach*, published by Prentice-Hall, and more than 170 journal papers.

He serves as a Series Editor for *Springer Verlag Lecture Notes in Computer Science*.