Hierarchical Clustering of Dynamical Systems based on Eigenvalue Constraints

Hiroaki Kawashima and Takashi Matsuyama

Graduate School of Informatics, Kyoto University,
Yoshida-Honmachi Sakyo, Kyoto 6068501, JAPAN,
{kawashima,tm}@i.kyoto-u.ac.jp

Abstract. This paper addresses the clustering problem of hidden dynamical systems behind observed multivariate sequences by assuming an interval-based temporal structure in the sequences. Hybrid dynamical systems that have transition mechanisms between multiple linear dynamical systems have become common models to generate and analyze complex time-varying event. Although the system is a flexible model for human motion and behaviors, the parameter estimation problem of the system has a paradoxical nature: temporal segmentation and system identification should be solved simultaneously. The EM algorithm is a well-known method that solves this kind of paradoxical problem; however the method strongly depends on initial values and often converges to a local optimum. To overcome the problem, we propose a hierarchical clustering method of linear dynamical systems by constraining eigenvalues of the systems. Due to the constraints, the method enables parameter estimation of dynamical systems from a small amount of training data, and provides well-behaved initial parameters for the EM algorithm. Experimental results on simulated and real data show the method can organize hidden dynamical systems successfully.

1 Introduction

Hybrid dynamical systems (hybrid systems) such as switching dynamical systems [6] and segment models [10] have become common models for speech recognition, computer vision, graphics, and machine learning researchers to generate and analyze complex time-varying event (e.g., human speech and motion [3, 12, 2, 9]). They assume that a complex event is consist of dynamic primitives, which is often referred to as phonemes, movemes [3], visemes, motion textons [9], and so on. For instance, a cyclic lip sequence in Figure 1 can be described by simple lip motions (e.g., "open", "close", and "remain closed"). Once the set of dynamic primitives is determined, an observed or generated time-varying pattern can be partitioned by temporal intervals with the labels of primitives.

A hybrid system represents each dynamic primitive by a simple dynamical system, and models transition between dynamical systems by a discrete-event model, such as an automaton and a hidden Markov model. Therefore, the system has a capability of generating and analyzing multivariate sequences that consist of temporal regimes of dynamic primitives.

In spite of the flexibility of hybrid systems, especially for modeling human motion and behaviors such as gestures and facial expressions, the real applications
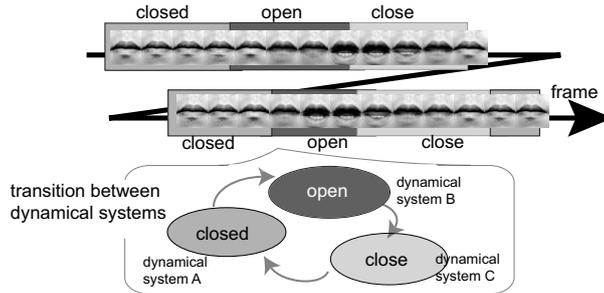
Fig. 1. Example of a lip image sequence modeled by a hybrid system.

are often beset with difficulties of parameter estimation due to the paradoxical nature of the estimation problem, as we see in the next paragraph.

This paper proposes a bottom-up approach that estimates a set of dynamical systems based on an agglomerative hierarchical clustering process, which iteratively merges dynamical systems. A constraining method for system eigenvalues (spectra) is proposed to identify stable linear dynamical systems, which are appropriate systems to model human motion, from a small number of training sequences. In this paper, we use only linear dynamical systems to model dynamic primitives and often omit the term "linear". Since the hierarchical clustering method provides approximate parameters of linear dynamical systems comprised in a hybrid system, it successfully initializes refinement process of the overall system such as a maximum likelihood estimation process.

Difficulty of the Parameter Estimation: Let us assume that a large amount of training data (multivariate sequences) is given. Then, the parameter estimation problem requires us to simultaneously estimate temporal partitioning of the training data (i.e., segmentation and labeling) and a set of dynamical systems. The reason is that identification methods of dynamical systems require partitioned and labeled training sample sequences; meanwhile segmentation and labeling methods of the sample sequences require an identified set of dynamical systems. The expectation-maximization (EM) algorithm [5] is a well-known method that solves this kind of paradoxical problems with iterative calculations; however, it strongly depends on the initial parameters and does not converge to the optimum solution, especially if the model has a large parameter space to search. Therefore, the parameter estimation of hybrid systems necessitates an initialization method that searches an appropriate set of dynamical systems (i.e., the number and parameters of dynamical systems) from given training data.

The Assumed Parameter Estimation Scheme: To solve the problem above, we assume a multiphase learning approach (see Figure 2). The first step is a hierarchical clustering process of dynamical systems, which is applied to a comparatively small number of typical sequences selected from given training data set. For the second step, we assume a refinement process for all the system parameters based on a maximum likelihood method via EM algorithm [12]. The method not only refines parameters of dynamical systems but estimates parameters of
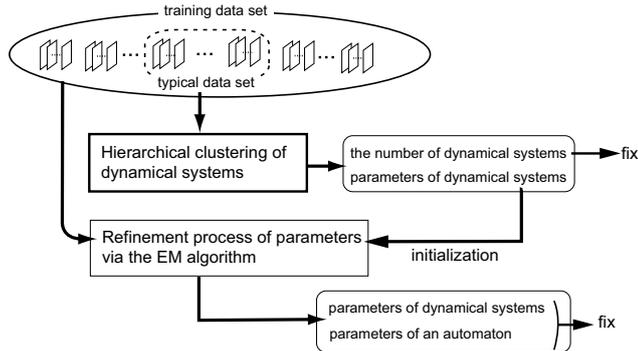
Fig. 2. The assumed parameter estimation scheme of a hybrid system. This paper concentrates on a hierarchical clustering of dynamical systems that works as an initialization process of the EM algorithm.

the automaton that models transition between the constituent linear dynamical systems. This refinement process is applied to all the given training data. Thanks to the estimated parameters in the hierarchical clustering process, the refinement process can be initialized by parameters that are relatively close to optimum compared to randomly selected parameters. As a result, the refinement process converges to the optimal solution successfully. This paper concentrates on the first step of the multiphase approach as an initialization process for the second step (i.e., EM algorithm).

The Advantage of the Hierarchical Clustering: Although several clustering approaches have been proposed to find a set of linear dynamical systems from given training sequences, such as greedy approaches [9], we propose an agglomerative hierarchical clustering method that extracts dynamical systems. The reason is that the method provides useful interfaces, such as the history of model fitting errors in each merging steps, to determine the number of clusters.

In Section 2, we describe a structure of a hybrid system. Section 3 explains the hierarchical clustering method proposed in this paper. We evaluate the method using simulated and real data to verify the expressiveness of the extracted dynamical systems in Section 4.

2   A Hybrid Dynamical System

2.1   System Architecture

A hybrid system is a generative model that can generate multivariate vector sequences by changing (or switching) the activation of constituent dynamical systems [6, 12]. In most case, the dynamical systems are linear. The system has a two-layer architecture. The first layer has a finite state automaton that models stochastic transition between dynamic primitives. The automaton has an ability of generating interval sequences, where each interval is labeled by one of the dynamic primitives. The second layer consists of a set of multiple dynamical systems $\mathcal{D} = \{D_1, ..., D_N\}$. In this paper, all the constituent dynamical systems

are assumed to share a n-dimensional continuous state space, and each activated dynamical system can generate sequences of continuous (real valued) state vector $x \in \mathbf{R}^n$. The generated state sequences are mapped to observation sequences of multivariate vector $y \in \mathbf{R}^m$ in a m-dimensional observation space by a linear function that is also shared by all the dynamical systems.

2.2 Linear Dynamical Systems

The state transition in the continuous state space by a linear dynamical system D_i, and the mapping from the continuous state space to the observation space is modeled as follows:

$$x_t = F^{(i)}x_{t-1} + g^{(i)} + \omega_t^{(i)}, \quad y_t = Hx_t + v_t, \tag{1}$$

where $F^{(i)}$ is a transition matrix and $g^{(i)}$ is a bias vector. Note that each dynamical system has $F^{(i)}$ and $g^{(i)}$ individually. H is an observation matrix that defines linear projection from the continuous state space to the observation space. $\omega^{(i)}$ and v is a process noise and an observation noise, respectively. We assume that the process noise and the observation noise has Gaussian distribution $\mathcal{N}(0, Q^{(i)})$ and $\mathcal{N}(0, R)$, respectively. The notation $\mathcal{N}(a, B)$ is a Gaussian distribution with an average vector a and a covariance matrix B. As we described in the previous subsection, we assume that all the dynamical systems share a continuous state space to simplify the model and to reduce the parameters. Using the notations above, we can consider the probability distribution functions: $p(x_t|x_{t-1}, d_t = D_i) = \mathcal{N}(F^{(i)}x_{t-1}, Q^{(i)})$ and $p(y_t|x_t, d_t = D_i) = \mathcal{N}(Hx_t, R)$, where the variable d_t represents an activated dynamical system at time t.

Calculation of Likelihood in Intervals: Let us assume that a continuous state has a Gaussian distribution at each time t. Then, the transition of the continuous state becomes a Gauss-Markov process, which is inferable in the same manner as the Kalman filtering [1]. Therefore, the predicted state distribution under the condition of observations from 1 to $t-1$ is formulated as $p(x_t|y_1^{t-1}, d_t = D_i) = \mathcal{N}(x_{t|t-1}^{(i)}, V_{t|t-1}^{(i)})$ and $p(y_t|y_1^{t-1}, d_t = D_i) = \mathcal{N}(Hx_{t|t-1}^{(i)}, HV_{t|t-1}^{(i)}H^{\mathrm{T}} + R)$, where the average vector $x_{t|t-1}^{(i)}$ and covariance matrix $V_{t|t-1}^{(i)}$ are updated every sampled time t. Suppose that the dynamical system D_i represents an observation sequence $y_b^e \triangleq y_b, ..., y_e$, which has a duration length $e-b+1$, then the likelihood that the system D_i generates the sequence is calculate by the following equation:

$$p(y_b^e|d_b^e = D_j) = \prod_{t=b}^{e} p(y_t|y_1^{t-1}, d_t = D_j), \tag{2}$$

where we assume a Gaussian distribution $N(x_{\mathrm{init}}^{(i)}, V_{\mathrm{init}}^{(i)})$ for the initial state distribution in each interval represented by dynamical system D_i.

In the following sections, we assume the observation matrix and the noise covariance matrix is $H = I$ (unit matrix) and $R = O$ (zero matrix), respectively, to concentrate on extracting dynamic primitives represented by transition matrices. Hence, the parameters to be estimated in a hybrid system become the following.

- the number of dynamical systems N
- the parameters of dynamical systems $\Theta = \{\theta_1, ..., \theta_N\}$, where
  $\theta_i = \{F^{(i)}, g^{(i)}, Q^{(i)}, x_{\text{init}}^{(i)}, V_{\text{init}}^{(i)}\}$ is a parameter set of dynamical system D_i
- the parameters of an automaton that models transition between dynamics

As we described in the Section 1, we concentrate on estimating N and Θ that initializing the EM algorithm. We assume that the parameters of the automaton are estimated and the parameter set Θ is refined by the EM algorithm.

2.3   Generable Time-varying Patterns by Linear Dynamical Systems

The generable class of time-varying patterns (corresponds to trajectories of points in the state space) from a linear dynamical system can be described by the eigenvalues of the transition matrix. To concentrate on the temporal evolution of the state in the dynamical system, let us assume the bias and the process noise term is zero in Equation (1). Using the eigenvalue decomposition of the transition matrix:

$$F = E\Lambda E^{-1} = [e_1, ..., e_n]\text{diag}(\lambda_1, ..., \lambda_n)[e_1, ..., e_n]^{-1},$$

we can solve the state at time t with initial condition x_0:

$$x_t = F^t x_0 = (E\Lambda E^{-1})^t x_0 = E\Lambda^t E^{-1} x_0 = \sum_{p=1}^{n} \alpha_p e_p \lambda_p^t, \tag{3}$$

where e_p and λ_p is a corresponding eigenvalue and eigenvector pair. We omit the indices i for simplification. A weight value α_p is determined from the initial state x_0 by calculating $[\alpha_1, ..., \alpha_n]^\top = E^{-1} x_0$. Hence, the generable patterns from the system can be categorized by the sign (especially in the real parts) and the norm of the eigenvalues $\lambda_1, ..., \lambda_n$. For instance, the system can generate time-varying patterns that converge to certain values if and only if $|\lambda_p| < 1$ for all $1 \leq p \leq n$ (using the term in control theory, we can say that the system is stable); meanwhile, the system can generate non-monotonic or cyclic patterns if the imaginary parts have nonzero values.

3   Hierarchical Clustering of Dynamical Systems

The goal of the hierarchical clustering process is to estimate the parameters N and Θ by assuming only a small amount of typical training data is given.

Let us assume that a multivariate sequence $y_1^T \triangleq y_1, ..., y_T$ is given as a typical training data (we consider a single training data without loss of generality), then we simultaneously estimate a set of dynamical systems $\mathcal{D}$ (i.e., the number of dynamical system N and the parameter set Θ) with an interval set $\mathcal{I}$ (i.e., segmentation and labeling of the sequence), from the training sample y_1^T. Note that, the number of intervals K is also unknown. We formulate the problem as the search of the linear dynamical system set $\mathcal{D}$ and the interval set $\mathcal{I}$ that maximizes the total likelihood of the training data: $\mathcal{L} = P(y_1^T | \mathcal{I}, \mathcal{D})$. Because the likelihood monotonically increases with an increase in the number of dynamical systems, we need to determine the right balance between the likelihood and the

number N. A hierarchical clustering approach provides us an interface, such as the history of model fitting errors in each merging step, to decide the number of dynamical systems.

To identify the system parameters from only a small amount of training data, we need constraints to estimate an appropriate dynamics. In this paper, we concentrate on extracting human motion primitives observed in such as facial motion, gaits, and gestures; therefore, constraints based on stability of dynamics are suitable to find motion that converges to a certain state from an initial pose. The key idea to estimate stable dynamics is the method that constrains on eigenvalues. If all the eigenvalues are lower than 1, the dynamical system changes the state in a stable manner, as we described in Subsection 2.3.

In the following subsections, we first propose a constrained system identification method that constrains an upper bound of eigenvalues in the transition matrices of linear dynamical systems. The method enables us to find a set of dynamical systems that represents only stable dynamics. Second, we describe an agglomerative hierarchical clustering of dynamical systems based on the pseudo distance between two dynamical systems. The algorithm also merges two interval sets labeled by the same dynamical system in each iteration step. Thus, the clustering method solves two problems simultaneously: temporal segmentation and parameter estimation.

3.1 Constrained System Identification

Given a continuous state sequence mapped from an observation space, the parameter estimation of a transition matrix $F^{(i)}$ from the sequence of continuous state vectors $x_b^{(i)}, .., x_e^{(i)}$ becomes a minimization problem of prediction errors. Let us use the notations $X_0^{(i)} = [x_b^{(i)}, ..., x_{e-1}^{(i)}]$ and $X_1^{(i)} = [x_{b+1}^{(i)}, ..., x_e^{(i)}]$, if the temporal interval $[b, e]$ is represented by a linear dynamical system D_i. Then, we can estimate the transition matrix $F^{(i)}$ by the following equation:

$$F^{(i)*} = \arg\min_{F^{(i)}} ||F^{(i)} X_0^{(i)} - X_1^{(i)}||^2 = \lim_{\delta^2 \to 0} X_1^{(i)} X_0^{(i)\top} (X_0^{(i)} X_0^{(i)\top} + \delta^2 I)^{-1}, \quad (4)$$

where I is the unit matrix and δ is a positive real value.

To set a constraint on the eigenvalues, we stop the limit in the Equation (4) before $X_0^{(i)\top}(X_0^{(i)} X_0^{(i)\top} + \delta^2 I)^{-1}$ converges to the pseudo-inverse matrix of $X_0^{(i)}$. Using Gershgorin's theorem in linear algebra, we can determine the upper bound of eigenvalues in the matrix from its elements. Suppose $f_{uv}^{(i)}$ is an element in row u and column v of the transition matrix $F^{(i)}$. Then, the upper bound of the eigenvalues is determined by $\mathcal{B} = \max_u \sum_{v=1}^n |f_{uv}^{(i)}|$. Therefore, we search a nonzero value for δ, which controls the scale of elements in the matrix, that satisfies the equation $\mathcal{B} = 1$ via iterative numerical methods.

3.2 Hierarchical Clustering of Dynamical Systems

The hierarchical clustering algorithm is initialized by partitioning the training sequence into motion and stationary pose intervals, which are simply divided using the scale of the first-order temporal difference of training data. In the

---

Hierarchical clustering of dynamical systems

```
for i ← 1 to N do
    D_i ← Identify (I_i)
end for
for all pair(D_i, D_j) where D_i, D_j ∈ 𝒟
do
    Dist (i, j) ← CalcDistance (D_i, D_j)
end for
while N ≥ 2 do
    (i*, j*) ← arg min_(i, j) Dist (i, j)
    𝓘_i* ← MergeIntervals (𝓘_i*, 𝓘_j*)
    D_i* ← Identify (𝓘_i*)
    erase D_j* from 𝒟;    N ← N − 1
    for all pair(D_i*, D_j) where D_j ∈ 𝒟
    do
        Dist(i*, j) ← CalcDistance (D_i*, D_j)
    end for
end while
```
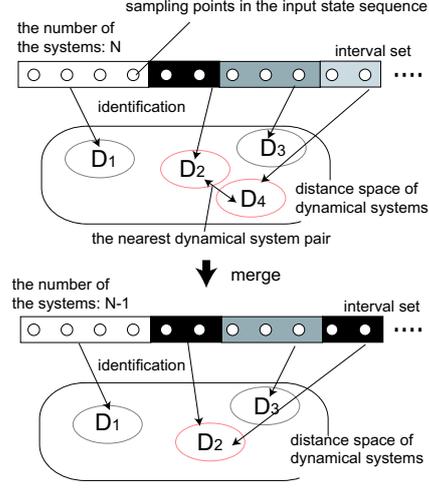
Fig. 3. Identify is a constrained system identification that we described in Subsection 3.1 $\mathcal{I}_i$ is an interval set that comprises intervals labeled by D_i. CalcDistance calculates the distance between the two modes defined in Subsection 3.2. MergeIntervals merges two interval set belongs to the nearest dynamical system pair.

first step of the algorithm, a single dynamical system is identified from each interval in the initial interval set. Then, we calculate a pseudo distances for all the dynamical system pairs based on the distance definition in the next paragraph. In the second step, the nearest dynamical systems are merged iteratively based on an agglomerative hierarchical clustering (see Figure 3.2). As a result, all the dynamical systems are merged to one dynamical system. We discuss the determination of the number of dynamical systems in the remaining of this subsection.

Distance Definition between Dynamical Systems: We define a pseudo distance between dynamical systems D_i and D_j as an average of two asymmetric divergences: $Dist(D_i, D_j) = \{KL(D_i||D_j) + KL(D_j||D_i)\}/2$, where each of the divergences is calculated as an approximation of Kullback-Leibler divergence [7]:

$$KL(D_i||D_j) \sim \frac{1}{|\mathcal{I}_i|} \sum_{I_k \in \mathcal{I}_i} \left\{ \log p(y_{b_k}^{e_k}|d_{b_k}^{e_k} = D_i) - \log p(y_{b_k}^{e_k}|d_{b_k}^{e_k} = D_j) \right\},$$

where $y_{b_k}, ..., y_{e_k}$ is a partitioned sequence by interval I_k. $|\mathcal{I}_i|$ is the summation of interval length in the interval set $\mathcal{I}_i$ that is labeled by a linear dynamical system D_i. Note that we can calculate the likelihoods based on Equation (2).

Cluster Validation Problem: The determination of the appropriate number of dynamical systems is an important problem in real applications. The problem is often referred to as the cluster validation problem, which remains essentially unsolved. There are, however, several well-known criteria, which can be categorized into two types, to decide the number of clusters. One is defined based on

the change of model fitting scores, such as log-likelihood scores and prediction errors (approximation of the log-likelihood scores), during the merging steps. If the score is decreased rapidly, then the merging process is stopped [11]. In other words, it finds *knee* of the log-likelihood curve. The other is defined based on information theories, such as minimum description length and Akaike's information criterion. The information-theoretical criteria define the evaluation functions that consist of two terms: log-likelihood scores and the number of free parameters.

Although information-theoretical criteria work well in simple models, they tend to fail in evaluating right balance between the two terms, especially if the model becomes complex and has a large number of free parameters [8]. Because the problem also arises in our case, we use model fitting scores directly. First, we extract candidates for the numbers of the dynamical systems by finding peaks in difference of model fitting errors between adjacent two steps. If the value exceeds a predefined threshold, then the number of dynamical systems in that step is added to the candidates. We consider that user should finally decide the appropriate number of dynamical systems from the extracted candidates.

4 Experimental Results

For the first evaluation, we used simulated sequences for training data to verify the proposed clustering method, because it provides the ground truth of the estimated parameters. Three linear dynamical systems and their parameters were set manually. The dimension was $n = 2$; therefore each of the system had 2×2 transition matrix. A two-dimensional vector sequence $Y = [y_1, ..., y_L]$ (Figure 4 (b)) was generated as an observation sequence from simulated transition between the dynamical systems based on the activation pattern in Figure 4 (a). The length of the sequence was $L = 100$. We then applied the clustering method proposed in Section 3. Figure 4 (c) shows the overall model fitting error between the original sequence Y and generated sequences $Y^{\mathrm{gen}}(N)$ from the extracted N dynamical systems. The error was calculated by the Euclid norm: $Err(N) = ||Y - Y^{\mathrm{gen}}(N)|| = \sqrt{\sum_{t=1}^{L} ||y_t - y^{\mathrm{gen}}(N)_t||^2}$. Figure 4 (d) shows the results of temporal segmentation partitioned by the extracted dynamical systems in each iteration step. We see that the error increases monotonically with the decrease in the number of dynamical systems. Note that there are several steep slopes in the chart. The steep slopes correspond to the iteration steps in which dynamical system pairs with a long distance were merged. The candidates of the number were determined as $N = 3$ (which corresponds to the ground truth) and $N = 8$ by extracting the steps in which the difference $Err(N - 1) - Err(N)$ exceeds the given threshold. Consequently, the history of model fitting errors helps us to decide the appropriate number of dynamical systems.

For the second evaluation, we applied the clustering method to real video data. A frontal facial image sequence was captured by 60fps camera. Facial feature points were tracked by the active appearance model [4, 13], and eight feature points around the right eye were extracted. The length of the sequence was $L = 1000$. We then applied the clustering method to the obtained 16-

dimensional vector sequence that comprised x- and y-coordinates of the feature points (both coordinate coefficients were plotted together in Figure 5 (a)). The candidates of the number of dynamical systems were determined as $N = 3$ and $N = 6$. Figure 5 (b) and (c) shows the error $||Y - Y^{\mathrm{gen}}(N)||$ in each step and the generated sequences in the steps of $N = 12$ and $N = 4$. We see that the generated sequence $Y^{\mathrm{gen}}(12)$ remains the spikes, which represent eye blinks, appeared in the original sequence; meanwhile, $Y^{\mathrm{gen}}(4)$ smoothes out them. For instance, the dominant dynamical systems D_2 and D_3 represent the intervals in which the eye remains closed and open, respectively. Hence, we can control the coarseness of the model by changing the number of dynamical systems, which work as the bases of original motion.

5 Conclusion

This paper proposed a hierarchical clustering method that finds a set of dynamical systems, which can be exploited to multiphase parameter estimation for hybrid systems that comprises a finite state automaton and multiple linear dynamical systems. The experimental results on simulated and real data show that the proposed hierarchical clustering method successfully finds a set of dynamical systems that is embedded in the training data.

References

1. B. D. O. Anderson and J. B. Moor. *Optimal Filtering.* Prentice-Hall, 1979.
2. B. N. A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.
3. C. Bregler. Learning and recognizing human dynamics in video sequences. *Proc. of Intl. Conference on CVPR*, pages 568–574, 1997.
4. T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance model. *Proc. European Conference on Computer Vision*, 2:484–498, 1998.
5. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.
6. Z. Ghahramani and G. E. Hinton. Switching state-space models. *Technical Report CRG-TR-96-3, Dept. of Computer Science, University of Toronto*, 1996.
7. B. H. Juang and L. R. Rabiner. A probabilistic distance measure for hidden markov models. *AT & T Technical Journal*, 64(2):391–408, 1985.
8. D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Trans. on Image Processing*, 7(2):180–195, 1998.
9. Y. Li, T. Wang, and H. Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. *SIGGRAPH*, pages 465–472, 2002.
10. M. Ostendorf, V. Digalakis, and O. A. Kimball. From HMMs to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. Speech and Audio Process*, 4(5):360–378, 1996.

11. D. K. Panjwani and G. Healey. Markove random field models for unsupervised segmentation of textured color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(10):939–954, 1995.
12. V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. *Proc. of Neural Information Processing Systems*, 2000.
13. M. B. Stegmann, B. K. Ersboll, and R. Larsen. FAME - a flexible appearance modelling environment. *Informatics and Mathematical Modelling, Technical University of Denmark*, 2003.

(a) simulated transition pattern of three dynamical systems

(b) two-dimensional observation sequence generated from (a) (solid: the first element, dashed: the second element)

(c) errors between the original and generated sequences

(d) interval patterns partitioned by different dynamical systems during the clustering process
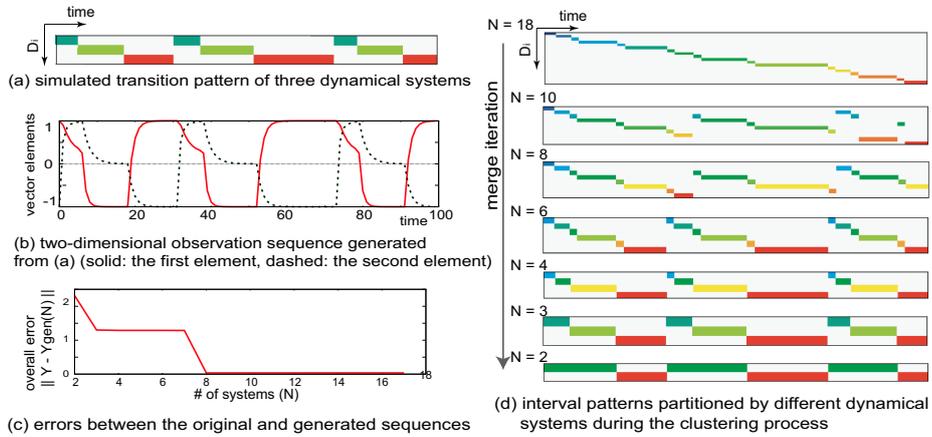
Fig. 4. Clustering results on the simulated sequence generated from three dynamical systems.

(a) the tracked feature points around the right eye

(b) error between original and generated sequences

(c) interval patterns partitioned by extracted dynamical systems (above) and generated sequences (below)
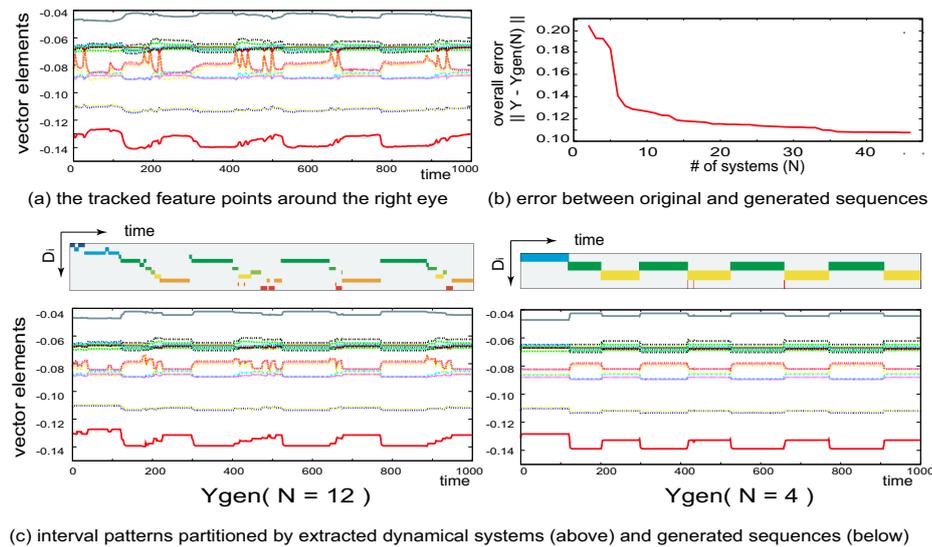
Fig. 5. Clustering results on the feature point sequence around the right eye during the subject smiled four times.