Bayesian Order-Adaptive Clustering for Video Segmentation

Peter Orbanz, Samuel Braendle, and Joachim M. Buhmann

Institute of Computational Science, ETH Zurich
{porbanz,jbuhmann}@inf.ethz.ch

Abstract. Video segmentation requires the partitioning of a series of images into groups that are both spatially coherent and smooth along the time axis. We formulate segmentation as a Bayesian clustering problem. Context information is propagated over time by a conjugate structure. The level of segment resolution is controlled by a Dirichlet process prior. Our contributions include a conjugate nonparametric Bayesian model for clustering in multivariate time series, a MCMC inference algorithm, and a multiscale sampling approach for Dirichlet process mixture models. The multiscale algorithm is applicable to data with a spatial structure. The method is tested on synthetic data and on videos from the MPEG4 benchmark set.

1 Introduction

Clustering algorithms usually operate on a fixed set of data. When clustering is applied to perform segmentation, the input data might e.g. be a digital image (group the image into spatially coherent segments) or a time series (decompose the series into coherent segments along the time axis, such as speaker clustering). In this article, we consider a different problem arising from the formalization of video segmentation as a clustering problem: Given is a time series of fixed-size data frames, each of which has a spatial structure, i.e. the 2D structure of the frame image. The series is to be decomposed into a sequence of spatially coherent segmentations of the frames, which should reflect the temporal smoothness of the sequence.

Clustering problems of this type have been actively studied in video segmentation [1]. For example, [2] proposes a parametric mixture model for optical flow features with neighborhood constraints. The number of clusters is selected by a likelihood heuristic. Temporal context is modeled implicitly by using differential motion features. Explicit context models include designs based on HMMs [3] or frame-to-frame model adaptation [4]. A method which approaches the problem's time series structure in a manner similar to Bayesian forecasting has recently been suggested in [5]. The authors propose a Gaussian mixture model to represent image rather than motion features. Temporal context is incorporated by using the estimate obtained on a given frame in the sequence as prior information for the following frame.

We propose a Bayesian method capable of addressing both temporal context and estimation of the number of clusters by a single model. The distribution of each cluster in feature space is described by an exponential family model, which is estimated under its respective conjugate prior [6]. The components are combined in a Bayesian mixture model to represent a segmentation. For each component, the prior is defined by the component's posterior estimated during the previous time step. Due to the "chaining" properties of conjugate pairs, this results in a closed model formulation for the entire time series. The mixture proportions and number of components are controlled by a Dirichlet process (DP) prior [7]. As we will argue, the conjugate nature of the DP leads to a chaining property analogous to the exponential family case. This property is used to propagate cluster structure along the time series in a similar manner as the conjugate component distributions propagate parameter information. Inference of the model is conducted by an adaptation of the Gibbs sampler for DP mixture models [8] to the time series model. To facilitate application of our model to the large amounts of data arising in video segmentation, we (i) show how the efficiency of the Gibbs sampler can be substantially increased by exploiting temporal smoothness and (ii) introduce a multiscale sampling method to speed up processing of individual frames. Just as the model, the multiscale algorithm is based on the properties of exponential family distributions.

2 Background

Method overview. Our approach to video segmentation is based on local features extracted from each image frame (where the "image" may be the original frame image, one of its color space dimensions, or a filter response image). A local window is positioned around each point of an equidistant grid within the image, and the pixel values within the window are extracted as feature values. The data vectors described in the following may, for example, be histograms of the pixels within local windows. They will generally be denoted as $\mathbf{x}_i^t$, where t is a time index (frame index) and i indexes a window position within the frame image. Image segments are modeled as clusters. Each cluster k is modeled by a distribution $F(\mathbf{x}_i^t|\theta_k^t)$. That is, the parameter vectors θ_k^t describe the segments, and constitute the target variables of the estimation problem. Priors on the parameters will generally be denoted G. For a given time t, the individual cluster models are combined into an overall segmentation solution by joining them in a Bayesian mixture model. A DP is used adapt the model order (i.e. the number of clusters). DPs define distributions on the mixture weights of a mixture model such that the total number of clusters is a random variable. In the video or time series context, they are suitable for the definition of clustering models that allow the number of clusters to change between time steps, as segments appear in or disappear from the scene. This section will review the basic ingredients of the model: Mixture models, conjugate prior distributions and Dirichlet processes (including MCMC inference). These concepts will be used in Sec. 3 to define

a model for clustering in time series. Estimation algorithms for the model are developed in Sec. 4.

2.1 Clustering with Mixture Models

A *finite mixture model* is a probability density representable as a convex combination

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{N_C} c_k F(\mathbf{x}|\theta_k) \tag{1}$$

of component densities $F(\,.\,|\theta_k)$. In the clustering context, each component density represents the distribution of a single cluster. For a given data set $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a *clustering solution* is an assignment of each observation $\mathbf{x}_i$ to one cluster $k \in \{1, \ldots, N_C\}$. Assigning an observation expresses the assumption that it was generated by the respective density $F(\,.\,|\theta_k)$. We encode such a solution by a set of *assignment variables* $S_1, \ldots, S_n$, one for each observation, where $S_i = k$ if $\mathbf{x}_i$ is assigned to cluster k. Clustering solutions are obtained either by expectation-maximization (EM) algorithms [9], or by Markov chain Monte Carlo sampling in a Bayesian regime. Both approaches rely on a latent variable structure, by treating the assignment variables S_i as random quantities that are estimated from the data along with the parameters.

2.2 Conjugate Models

Several aspects of this work build on exponential family models and the concept of conjugacy [6]. We provide a brief summary of these models and those properties of importance for our approach. A distribution of a random variable $\mathbf{X}$ with domain Ω_x is called an *exponential family model* if its density can be written as

$$F(\mathbf{x}|\theta) := h(\mathbf{x}) \exp(\langle s(\mathbf{x})|\theta\rangle - \phi(\theta)) \,. \tag{2}$$

Here, $\theta \in \Omega_\theta$ is a parameter vector, $\langle\,.\,|\,.\,\rangle$ denotes the scalar product on the parameter space Ω_θ, and the function ϕ is the normalization term $\phi(\theta) := \log \int h(\mathbf{x}) \exp(\langle s(\mathbf{x})|\theta\rangle) d\mathbf{x}$. Of particular interest is the *sufficient statistic* function $s : \Omega_x \to \Omega_\theta$, which effectively defines all properties of the model relevant for parameter estimation. If a density F and a prior G are used in a Bayesian estimation framework, they are called *conjugate* if the resulting posterior is a distribution of the same type as G, i.e. prior and posterior differ only in their parameters. The concept of conjugacy is inherently tied to exponential families: It can be shown that, on the one hand, any exponential family model has a conjugate model (which is also an exponential family). On the other hand, only exponential family models have non-trivial conjugate models [6]. If F is a model of the form (2), then $G(\theta|\lambda, y) := \frac{1}{K(\lambda,y)} \exp(\langle\theta|y\rangle - \lambda\phi(\theta))$ is a conjugate prior, and the posterior under n independent observations $\mathbf{x}_i$ is

$$G(\theta|\lambda + n, \mathbf{y} + \sum_{i=1}^{n} s(\mathbf{x}_i)) \,. \tag{3}$$

The average $\sum s(x_i)$ carries all information the observation sample contains about the model parameter, i.e. it is sufficient. Conjugacy is a key property for a Bayesian time series model that uses the posterior of a previous step as prior for the present one: If a non-conjugate pair is used, the derived prior will change from time step to time step.

2.3 Dirichlet Process Mixture Models

Model order selection. For mixture model estimation, the overall number N_C of clusters is an input parameter. Techniques for estimation of this quantity from data are referred to as *model order selection*. Non-Bayesian solutions are usually based on regularity assumptions: The model's capability to explain the given data, measured by the likelihood, is traded off against a measure of model complexity to avoid overfitting. Model complexity is typically measured as a function of the model and the sample size (information criteria approaches such as AIC, BIC and MDL; cf [10] for an overview). The *stability* method [11] measures model complexity as stability of the solution under random perturbations of the data. All these methods proceed in an exhaustive search manner, i.e. scores for a range of models are computed and the best model is chosen. Bayesian DP-based methods [8] represent the number of clusters as a random variable. An estimate is obtained by sampling the model posterior under the observed data. Whereas scoring methods aim at identifying the "true" number of clusters under a chosen set of assumptions, DP methods provide a user parameter that controls the qualitative behavior of the random variable N_C. As will be discussed below, this makes them a natural choice for problems in which the number of model clusters has to change adaptively over a range of instances.

The Dirichlet process approach. In our clustering model, individual cluster components will be represented by parametric exponential family models, which are combined in a mixture model. To control the structure of the mixture, we use a Dirichlet process mixture (DPM) model [8, 12]. For the sake of brevity, we will forego the model's derivation as a stochastic process and only describe its practical properties. Loosely speaking, a DPM is a mixture model with a "complexity control" term. The extra term governs the creation of new components. In the finite parametric mixture (1), different components of the mixture are defined by different values of the parameter vector θ_k. If the mixture is estimated from data, the mixture weights are chosen in proportion to the size of classes, i.e. $c_k = \frac{n_k}{n}$ if n_k out of n total data points are assigned to component k. The model (1) can be regarded as the result of integrating the distribution function F against

$$G_{\text{MM}}(\theta) := \sum_{k=1}^{N_C} c_k \delta_{\theta_k}(\theta), \tag{4}$$

where δ_{θ_k} denotes the Dirac function on Ω_θ centered at θ_k. DPM models augment the expression (4) by an extra term,

$$G_{\text{DP}}(\theta) := \sum_{k=1}^{N_C} c_k \delta_{\theta_k}(\theta) + \alpha G_0(\theta) \; . \tag{5}$$

The function G_0 is a probability distribution on the domain Ω_θ of mixture parameters, and $\alpha \in \mathbb{R}_+$ a positive scalar parameter. The dynamics of this model can be illustrated by considering a data generation process: A new datum $\mathbf{x}$ is generated by drawing from a standard mixture or DPM, which we assume to have been estimated from data $\mathbf{x}_1, ..., \mathbf{x}_n$. When drawing from a mixture model (corresponding to (4)), $\mathbf{x}$ will be drawn from one of the N_C parametric mixture components in a two-stage manner by sampling $\theta \sim G_{\text{MM}}$, then $\mathbf{x} \sim F(\,.\,|\theta)$. In the DPM case (5), $\mathbf{x}$ is drawn either according to one of the N_C parametric mixture components, or (with a probability proportional to α) from a new mixture component $F(\,.\,|\theta^*_{N_C+1})$, where the new parameter value $\theta^*_{N_C+1}$ is sampled from G_0. The DPM model constitutes a Bayesian approach to model order selection in mixture models, due to its ability to generate as many clusters as required to explain the observed data.

The standard Gibbs sampling algorithm for DPM models [13] encodes assignments of data points to mixture components by means of latent variables, in a manner similar to the EM algorithm. For each $\mathbf{x}_i$, the discrete index variable S_i specifies the index of the mixture component to which $\mathbf{x}_i$ is assigned. Within the algorithm, a value of $S_i = 0$ indicates the generation of a new mixture component from G_0 as a model for $\mathbf{x}_i$. The S_i are determined by computing mixture proportions $\tilde{q}_{ik}$ as

$$\begin{aligned} \tilde{q}_{i0} &:= \int_{\Omega_\theta} F(\mathbf{x}_i|\theta) G_0(\theta) d\theta \\ \tilde{q}_{ik} &:= n_k^{-i} F(\mathbf{x}_i|\theta_k) \qquad \text{for } k = 1, ..., N_C, \end{aligned} \tag{6}$$

where n_k^{-i} is the number of data points assigned to component k with $\mathbf{x}_i$ removed from the data set. The proportions are normalized to obtain mixture probabilities $q_{ik} := \frac{\tilde{q}_{ik}}{\sum_{l=0}^{N_C} \tilde{q}_{il}}$, for $k = 0, \ldots, N_C$. The sampling algorithm repeats the following steps:

Assignment step: For all $i = 1, ..., n$,

1. Compute $q_{i0}, ..., q_{iN_C}$.
2. Sample $S_i \sim (q_{i0}, ..., q_{iN_C})$.
3. If $S_i = 0$, create a new component: Sample $\theta_{N_C+1} \sim F(\mathbf{x}_i|\theta_{N_C+1}) G_0(\theta_{N_C+1})$.

Estimation step: For each $k = 1, ..., N_C$, sample

$$\theta_k \sim G_0(\theta_k) \prod_{i|S_i=k} F(\mathbf{x}_i|\theta_k) \; . \tag{7}$$

In the conjugate case, the integral in (6) has an analytic solution and the component posteriors in (7) are distributions of the same type as G_0. Thus, if G_0 can be sampled, the component posterior can be sampled as well.

3 Order-Adaptive Clustering in Time Series

The present article considers the problem of obtaining a clustering solution for each time step in a multivariate time series. The clustering solutions are required to exhibit a temporal coherence. In a video sequence, each time step corresponds to a single frame image. The overall clustering solution then consists of a segmentation for each frame. If the number of clusters can change between frames, a suitable clustering method must be *order-adaptive*, i.e. capable of adjusting the model order over time. Order-adaptive methods require (i) automatic model order selection and (ii) a meaningful way to match clusters in different frames. If clustering solutions are obtained independently on each frame, the latter must be addressed by matching heuristics. Any principled approach requires the use of context information, i.e. the clustering solution for a given frame has to be obtained in a manner conditional on the solutions for the previous frame. In this section, we discuss how cluster structure can be propagated along a time series if the clustering solutions on individual frames are controlled by a DP prior.

An Order-Adaptive Clustering Model. We consider a multivariate time series $\mathbf{x}^t := (\mathbf{x}_1^t, \ldots, \mathbf{x}_n^t)$ that, for each time step $t = 1, 2, \ldots$, generates a set of n outputs $\mathbf{x}_i^t$. For each t, a single clustering solution $\mathbf{S}^t := (S_1^t, \ldots, S_n^t)$ is obtained. For temporal coherence, we require that, if $S_i^t = k$, then also $S_i^{t+1} = k$ with high probability, unless the corresponding observations $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t+1}$ differ significantly. For the video segmentation problem, this reflects the assumption that size and location of segments change slowly on the time scale of frame renewal. The standard Bayesian approach to address temporal coherence requirements in time series models is to encode context with priors. The posterior distribution of the model parameter vector θ^t at a given time is used as prior distribution θ^{t+1}. This requires a conjugate model, i.e. a model for which prior and posterior belong to the same family of distributions (otherwise, the type of the prior distribution would change between time steps). Though we are ultimately interested in DP priors, let us first exemplify the approach for a parametric model. Let $F(\mathbf{x}|\theta)$ be a density modeling observations at a single time step (that is, a likelihood). We use the *ab initio* form $G(\,.\,|\lambda, \mathbf{y})$ of the prior, with G conjugate to F. The prior for the parameter vector θ^{t+1} is the corresponding posterior under the previous observation $\mathbf{x}^t$,

$$G(\theta^t|\lambda + 1, \mathbf{y} + s(\mathbf{x}^t)) \propto F(\mathbf{x}^t|\theta^t)G(\theta^t|\lambda, \mathbf{y}) \; . \tag{8}$$

One may variate upon this strategy to accumulate observations over time,

$$G\Big(\theta^{t+1}\Big|\lambda + \tau, \mathbf{y} + \sum_{t-\tau<r\leq t} s(\mathbf{x}_r)\Big) \propto \prod_{t-\tau<r\leq t} F(\mathbf{x}^r|\theta^r)G(\theta^{t-\tau+1}|\lambda, \mathbf{y}) \tag{9}$$

resulting in a process with a τ-step memory. A similar mechanism is applicable to DP models. The DP has a natural conjugate property, implicit in Ferguson's [7] characterization of the DP posterior: If $\theta_1, \ldots, \theta_n \sim \mathrm{DP}(\alpha G_0)$, then $\theta_{n+1} \sim \mathrm{DP}\Big(\alpha G_0 + \sum_{i=1}^n \delta_{\theta_i}\Big)$. For convenience, we adopt the following symbolic notation: Observe that the Dirac sum $\hat{G}_n := \sum_{i+1}^n \delta_{\theta_i}$ of n draws from G can be regarded as a finite draw from an infinite categorial distribution (i.e. as a partial, finite observation from an "infinite histogram"). Just as a finite histogram can be generated by a multinomial distribution parameterized by a draw from a Dirichlet (its conjugate prior), the measure $\hat{G}_n$ can be explained as a draw from a multinomial process (MP) parameterized by a draw from a DP. The DP prior and posterior can then be linked explicitly in a Bayesian formula

$$\mathrm{DP}\left(\alpha G_0 + \hat{G}_n\right) \propto \mathrm{MP}\left(\hat{G}_n | G\right) \mathrm{DP}\left(\alpha G_0\right) \ .$$

Although our use of this notation is purely symbolic, it can be derived in a mathematically precise manner using Kolmogorov's extension theorem for stochastic processes. In perfect analogy to the conjugate parametric case, we can construct a DP prior for an estimation problem in a time series at time $(t+1)$ as the DP posterior at time t:

$$G^{t+1} \sim \mathrm{DP}\left(\alpha G_0 + \hat{G}_n^t\right) \propto \mathrm{MP}\left(\hat{G}_n^t | G^t\right) \mathrm{DP}\left(\alpha G_0\right) \ .$$

The formulation immediately extends to a τ-step memory, by means of

$$\mathrm{DP}\Big(G\Big|\alpha G_0 + \sum_{t-\tau<r\leq t} \hat{G}_n^r\Big) \propto \prod_{t-\tau<r\leq t} \mathrm{MP}\left(\hat{G}_n^r | G\right) \mathrm{DP}\left(G|\alpha G_0\right) \ . \tag{10}$$

Note that, for the cluster propagation problem, the draws from the DP are parameters θ_i rather than observations. The observed data $\mathbf{x}^{t+1}$ at time $(t+1)$ is then generated as

$$\begin{aligned} \mathbf{x}_i^{t+1} &\sim F(\,.\,|\theta_i^{t+1}) \\ \theta_i^{t+1} &\sim G^{t+1} \\ G^{t+1} &\sim \mathrm{DP}\left(\alpha G_0 + G^t\right) \ . \end{aligned} \tag{11}$$

Observations from multiple channels. In image and video processing applications, the input data usually consists of multiple channels. For standard color videos, three channels correspond to the three color space dimensions. Additional channels may include other features in the form of transformed data or filter responses. For multiple data channels indexed $c = 1, \ldots, C$, multiple observations $(\mathbf{x}_{i,c}^t)_{c=1,\ldots,C}$ are obtained for each frame t and location i. These are represented in the model as a product of likelihoods. That is, the generative model is obtained by substituting suitable product distributions $\prod_{c=1}^C F(\mathbf{x}_{i,c}^{t+1}|\theta_{i,c}^{t+1})$ and $\prod_{c=1}^C G_c^{t+1}$ for F and G_0 in (11). This does not increase the complexity of the clustering problem. Cluster assignments are still encoded by one variable S_i^t per

location. The product model defines C parallel estimation problems, which are coupled by the assignment variables. Since the latter are the actual target variables of the clustering problem, the approach effectively increases the amount of observational data per estimated solution variable. If the features are chosen well, this can significantly increase the quality of estimates. Choosing a large number of uninformative channels may have a converse effect: The assignment probabilities q_{ik}^t computed by the inference algorithm are derived from the product model as averages over channels. Multiple uninformative features may thus obfuscate information provided by informative ones.

A note on exchangeability. A well-known fact in statistics is that data generated from a DP is *exchangeable*, i.e. its probability is invariant under permutations of the order [6]. This raises some questions about the applicability of such models to time series, where the order of observations is crucial. The model described above derives a prior for step $(t+1)$ based on two inputs, the initial parameters $(\lambda, \mathbf{y})$, which apply uniformly in all steps, and the previous observation $\mathbf{x}^t$. The implicit assumption is that pairs of adjacent steps are exchangeable. This local exchangeability is, in turn, equivalent to assuming processing of the time series to be invariant under inversion of the time axis. In other words, inference results should not depend on whether the time series is processed in increasing or decreasing index order. Such an assumption is justifiable for some time series, and in particular for the video segmentation problem. Segmentation as a mid-level vision problem is oblivious to semantic and high-level content (such as whether a falling object moves downwards or upwards). However, it requires the neighborhood relations of frames to be preserved. Hence, we may invert the order, but not shuffle the frames. The argument does not apply to a process with a multi-step memory as in (10), which would require exchangeability of more than two frames. Such a model should therefore be regarded as an approximation. Note that many models assume independence of data points, which is a much stronger assumption than exchangeability. Such models often perform well even in applications where the generative process of the data will clearly result in dependent observations. We therefore believe that, in many cases, approximation by multi-step exchangeability may be beneficial, and the ability of such a model to draw on multiple observations will outweigh the loss of descriptive precision incurred by the violation of the model assumption.

4 MCMC Inference for High-Throughput Problems

In this section, we discuss inference techniques for the model described in Sec. 3. Available methods for DPM inference are extended to address two issues: Time series structure and efficiency. Existing hidden-variable methods can be modified for time series inference by initializing inference for a given time step by the model state estimated for the previous step. To increase the sampler efficiency for individual time steps, we propose a multi-scale sampling scheme exploiting the spatial structure of frame sequence data. Both approaches can be combined to obtain a efficient sampling algorithm.

4.1 Sampling in Time Series

Parameter inference for the time series clustering problem estimates the cluster parameters θ_k^t and the states of the assignment variables S_i^t, for each $t = 1, \ldots$. Estimates are obtained by sampling the relevant posteriors with a Gibbs sampler. To derive a suitable algorithm, we note that, for a given time index t, recovering the states S_i^t and parameters θ_k^t given the current observations $\mathbf{x}_i^t$ is a DPM mixture inference problem with prior $\mathrm{DP}(\alpha G_0 + \hat{G}_n^t)$. The history of the process enters via the prior parameter, i.e. the measure $(\alpha G_0 + \hat{G}_n^t)$. Full conditionals for the Gibbs sampler are immediately obtained from the standard sampling algorithm, by substituting $(\alpha G_0 + \hat{G}_n^t)$ for αG_0. (A sampler for a series with a τ-step memory can be obtained by substituting the corresponding posterior parameter in (10)). Estimates for the whole time series can be computed by running the Gibbs sampler for the appropriate posterior at each time step. The parameter estimates (summarized by $\hat{G}_n^t$) are then substituted into the DP prior of the subsequent step. The algorithm is an online method, as it only performs a single pass over the time series. The cluster correspondence problem is solved implicitly, by propagating information from one time step to the next through the DP base measure. Initially, the same clusters as in the previous step are available for assignment, and their indices are preserved. Classes may be newly generated by drawing from the continuous component G_0 of the DPM, or deleted if no longer supported by the data. Gibbs sampling is potentially time-consuming, and performing a full run of a DPM Gibbs sampler for each time step in the series is computationally prohibitive. A substantial speed-up is achieved by exploiting temporal smoothness. If changes in the data occur slowly w. r. t. to the time scale (frame rate) of the time series, the model state estimated at time t provides an almost-perfect initialization for sampling at time $(t+1)$. The algorithm therefore obtains an initial estimate at time $t = 1$ by performing a full run of the Gibbs sampler. For $t \geq 2$, the Gibbs sampler is initialized by the previous model state, and then run only for a few steps, to allow the model to adapt to changes in the data.

4.2 Multiscale Sampling

For data exhibiting a spatial neighborhood structure, DPM inference algorithms that are more efficient then the standard Gibbs sampler can be derived using a multiscale approach. Multiscale methods attempt to increase the efficiency of iterative algorithms by replacing the original input problem with a compressed replacement problem (*coarsening*). This reduced-size problem is solved, and the solution transformed into a solution of the larger input problem (*refinement*). The compression operation exploits neighborhood structures in the data (such as spatial or sequential neighborhoods). In images, adjacent pixels are grouped into blocks, and each block B is compressed by computing a summary variable $\mathbf{x}_B$. The coarsened problem is given by the set of summary variables for all blocks. The coarsening operation therefore has to be designed to limit the loss of

relevant information under compression, and to result in a coarse-scale problem to which the processing algorithm in question is applicable.

Coarsening. Our aim is the design of MCMC sampling algorithms. The information to be preserved under coarsening is therefore the information relevant to statistical parameter estimation. A simple averaging approach is not suitable in general, as it will only preserve moment information of first order. For the models considered in our work, a suitable coarsening approach can be derived from the properties of sufficient statistics. For an exponential family density as in (2), all information relevant to parameter estimation is contained in the sufficient statistic $s(\mathbf{x})$. Furthermore, for multiple observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$, the sum $\hat{s}_n := \sum_{i=1}^n s(\mathbf{x}_i)$ is sufficient. Given a data block B, consisting of the observations $\{\mathbf{x}_{b_1}, \ldots, \mathbf{x}_{b_N}\}$, the summary variable $\mathbf{s}_B$ is computed as

$$\mathbf{s}_B := \sum_{i=1}^{N} s(\mathbf{x}_{b_i}) \; . \tag{12}$$

If $\mathbb{R}^d$ data, for example, is modeled by a Gaussian distribution, the summary variable will be the pair $\mathbf{s}_B = \left(\sum_{i=1}^N \mathbf{x}_{b_i} \, , \, \sum_{i=1}^N \mathbf{x}_{b_i} \mathbf{x}_{b_i}^T \right)$. Coarsening is therefore performed by averaging in parameter space, in contrast to the standard multi-scale schemes used by many computer vision algorithms, which average in the data domain. A spatial partition of the input data into blocks $B_1, \ldots, B_m$ will result in a set of summary variables $\mathbf{x}_{B_1}, \ldots, \mathbf{x}_{B_m}$. The DPM sampling algorithm described in Sec. 2.3 is directly applicable to this replacement data, by substituting summary variables $\mathbf{x}_B$ for sufficient statistic values $s(\mathbf{x}_i)$.

The coarsening operation is *perfect* in the sense that it does, by the properties of sufficient statistics, preserve all information relevant for estimation purposes. More precisely, assume that $\mathbf{x}_{b_1}, \ldots, \mathbf{x}_{b_N} \sim F(\,.\,|\theta)$, with a conjugate prior $G(\theta|\lambda, \mathbf{y})$ on the parameter. Then the posterior $\mathit{\Pi}$ satisfies the invariance

$$\mathit{\Pi}(\theta|\mathbf{s}_B; \lambda, \mathbf{y}) = \mathit{\Pi}(\theta|\mathbf{x}_{b_1}, \ldots, \mathbf{x}_{b_N}; \lambda, \mathbf{y}) \; , \tag{13}$$

since

$$\mathit{\Pi}(\theta|\mathbf{x}_{b_1}, \ldots, \mathbf{x}_{b_N}; \lambda, \mathbf{y}) = G\Big(\theta \Big| \lambda \mathbf{y} + \sum_{i=1}^{N} s(\mathbf{x}_{b_i})\Big) = G(\theta|\lambda \mathbf{y} + \mathbf{s}_B) = \mathit{\Pi}(\theta|\mathbf{s}_B; \lambda, \mathbf{y}) \; .$$

Intuitively, a parameter estimated from data is a valid description of the data on the fine scale or *any* coarsened scale.

Refinement. The coarsening strategy described above and subsequent sampling on the coarse scale will result in a DPM clustering solution defined by cluster parameters $\theta_1^*, \ldots, \theta_k^*$. Because these parameters also define an admissible clustering solution on the fine (original) scale of the problem, it is not necessary to explicitly propagate coarse-scale assignments to the fine scale. Instead, a solution of the fine-scale problem is obtained by substituting the estimates θ_k^* into the fine-scale model and performing a single assignment step. We note that, as another consequence of the parametric description applying simultaneously

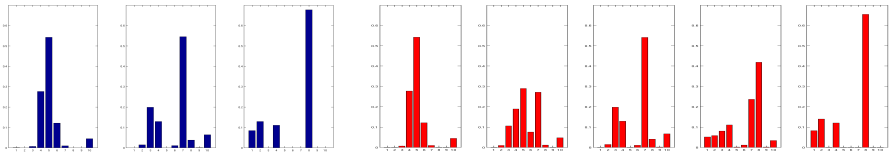

Fig. 1. Erroneous creation of classes during coarse-scale sampling: Average histograms of the three input classes (blue/left) and parameter vectors of the five classes estimated by coarse-scale sampling (red/right).

over different coarsening scales, the method is capable of incorporating locally adaptive coarsening/refinement strategies.

Coarse-scale artifacts. When applied to clustering, the multiscale sampler may erroneously produce additional classes at a coarse scale, if a block summarized by a single variable during coarsening overlaps the boundary between two segments. The resulting mixed distribution may distinctively differ from the average distribution of both segments, and thus produce an additional cluster. Such errors can be corrected by the fine-scale assignment step. To illustrate the behavior of the sampler, Fig. 1 shows estimation results obtained on a simple artificial image consisting of three block segments arranged in sequence (i.e. there are two boundaries between adjacent segments). Local windowed grayscale histograms are extracted as features. As clustering model, we apply a DPM model, with a multinomial likelihood F to account for the histograms. The cluster parameters θ_k^* can be interpreted as average histograms of the respective cluster. These average histograms are plotted for the true (generative) model on the left in Fig. 1. The multiscale algorithm is run on the data with a coarsening coefficient of 2, and the coarse-scale solution is compared to the artificial ground-truth. When sampling on a coarse scale, the algorithm models five classes, for which the class parameter vectors θ_k^* are plotted on the right. Clusters 2 and 4 are due to mixing of histograms at the segment boundaries. When assignments are performed for the fine-scale histograms to the coarse-scale class parameters, all histograms are correctly assigned to their original three classes, and clusters 2 and 4 remain empty. That is, coarse-scale artifacts vanish during the fine-scale assignment. The algorithm benefits from the ability of the DP to create new clusters for the boundary points, without distorting the remaining cluster structure.

Multiscale approaches to Markov Chain sampling have been considered e.g. in [14]. These methods are based on the idea that suitable coarse-scale formulations of a Markov chain may mix faster than the original chain, and use a coupled formulation integrating both chains. The aim is to reduce the number of iterations required for the algorithm to converge, while retaining accuracy. In contrast, our approach aims at reducing the execution time of individual iterations. Keeping in mind the large amounts of data arising in visual processing and video, we trade in accuracy and statistical guarantees for speed. Though the coarsening operation is perfect for individual distributions, it will lose in accuracy when the coarsening blocks overlap segment boundaries, as shown in

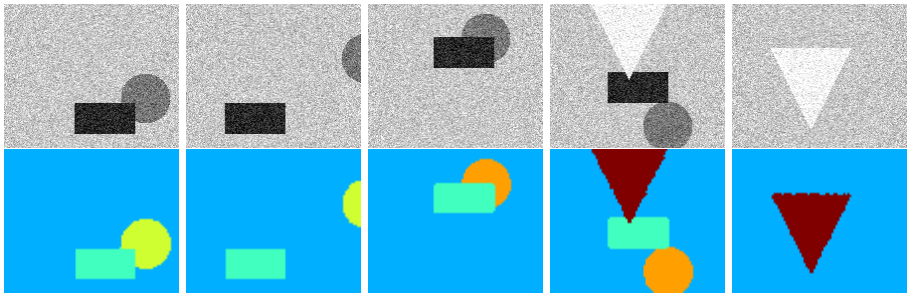

Fig. 2. Synthetic noisy data (top) and segmentation results (bottom). Clusters are correctly created or deleted as objects enter or leave the scene.

the example. In practice, we should not expect the fine-scale assignment step to correct all errors. The rationale for risking a loss of accuracy is that, for vision applications, we put more emphasis on speed and plausible results than on statistical guarantees.

5 Experiments

This section provides experimental results for the application of our model to video segmentation. Experiments were performed on both synthetic data and real-world data (sequences from the MPEG4 benchmark set).

Processing pipeline. Features are extracted from each frame image by placing an equidistant grid within the image. A local window is placed around each grid node i, pixel values are extracted from within the window, and collected in a histogram (denoted $\mathbf{x}_i^t$ in the previous sections). For color images, the method is applied individually to each color channel. The resulting set of features for each frame is a list of multiple histograms, indexed by their position i within the image. On this data, the inference algorithm described in Sec. 4 is applied. Inference is conducted single-pass, and is hence capable of online processing. For each time step t, the assignment variables S_i^t describe the estimated segmentation (i.e. S_i^t is interpreted as the segment index of site i). In the examples shown in Figs. 2-5, segment assignments have been color-coded.

Results. Synthetic data experiments were conducted to verify the method's capability to adjust the number of clusters. The artificial data consists of simple geometric objects with additive Gaussian noise moving at random within a scene. Objects may newly appear or disappear, but only by entering or leaving the scene from the border (i.e. temporal changes are smooth). A sample experiment is shown in Fig. 2. Features used are local gray-scale histograms. In all experiments conducted, the algorithm consistently assigns each object to the same cluster over the whole running time of the sequence. The cluster number only changes if an object vanishes temporarily, either by occlusion or because it leaves the scene and reappears (as is the case for the disc in Fig. 2). As a

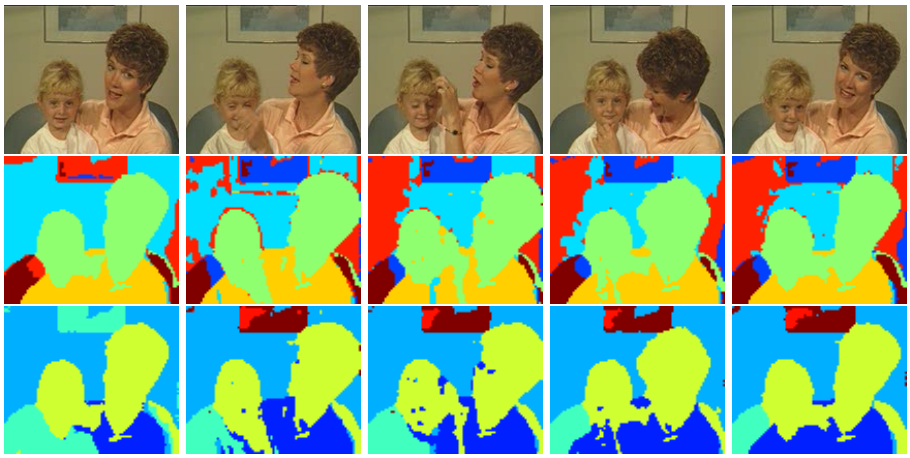

Fig. 3. "Mother and child" test sequence (top). Results are shown for color and saturation histogram features (middle), and with additional location features modeled by Gaussian distributions (bottom).

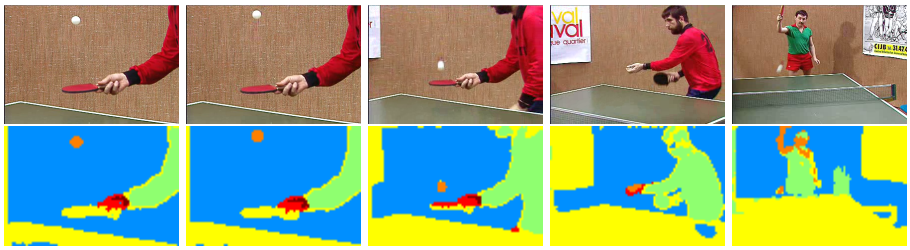

Fig. 4. "Table tennis" test sequence, with histogram and location features.

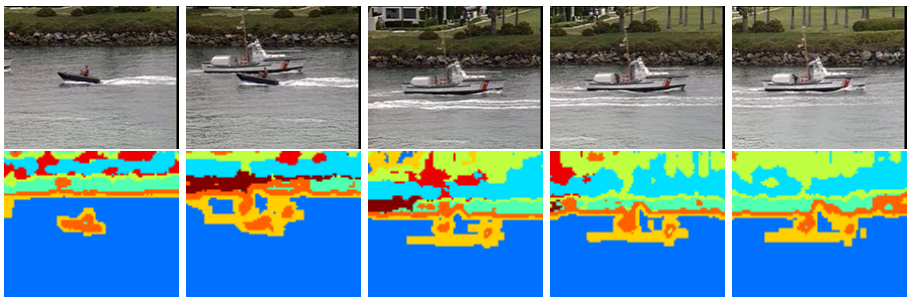

Fig. 5. More difficult data:"Coastguard" test sequence (top) and segmentation results (bottom), with histogram and location features.

mid-level vision algorithm, the method cannot (and should not) distinguish between initial appearance and reappearance of an object. Results on real video data were obtained on sequences from the MPEG4 benchmark set. Five feature channels where used: Four are histograms, representing the three RGB color channels plus saturation, and described by a multinomial likelihood F. In addition, we used a location feature, i.e. the center position of the local window in the image, represented by a two-dimensional Gaussian likelihood. The overall model likelihood is a product likelihood as described in Sec. 3. The scatter parameters of the parametric priors and the scatter parameter α of the DP, which control the level of cluster resolution, can be adjusted on the first few frames of the sequence. The key parameter of the feature extraction is the size of the local windows, which has to trade off sample size against precision: Large windows, to their advantage, contain many pixel examples, which results in stable histogram estimates and reduces scatter in feature space. Their drawback is a lack of precision: Large windows overlapping a cluster boundary generate histograms that represent a mixture of the two cluster distributions. Such mixtures tend to differ significantly from the individual distributions of the clusters, and hence cause additional clusters to appear at the segment boundaries. All results shown here were obtained using window sizes of 5×5 or 9×9 pixels. Sample results are shown in Figs. 3-5. In Fig. 3, results shown in the middle row where obtained using only the four histogram features (color and saturation). The background is split up into incoherent segments. Results are improved by the additional use of location features. Modeling these with a Gaussian in the spatial domain favors spatially coherent solutions, improving the segmentation of the background (bottom row). Likewise, all five features where used in the computation of results shown in Fig. 4. A more difficult sequence is shown in Fig. 5. In this case, local segmentation features provide poor information. Color differences within some segments (e.g. the large boat) are more significant than those between segments. With the size of the windows chosen sufficiently large during feature extraction to obtain stable input histograms, a boundary cluster effect is observable (note the two boats being split into an internal and a boundary segment). Results may possibly be improved by including additional motion features (such as histograms of frame differences). In general, the choice of features proves crucial for the performance of the segmentation algorithm. The parametric components of the clustering model (e.g. Gaussian and multinomial) are location-scatter type models, which represent "clouds" in their respective feature spaces. Like most mixture models, the method relies on the feature extraction step to map the segments to groups in feature space that are sufficiently well-separated to be resolved by the probabilistic model.

Average running times for our experiments on different video test sequences (300 frames at resolution 144×176 each) were: $\sim$ 190 seconds at full resolution, $\sim$ 110 seconds using a multi-scale sampler with coarsening coefficient 2, and $\sim$ 35 seconds with a coarsening coefficient of 4. This does not include the feature extraction, i.e. the extraction of the fine-scale input data from the image sequence. The running time of the algorithm scales, in addition to the obvious

dependence on the amount of input data, with the number of segments. The averages above were measured for relatively small numbers of classes ($N_C \leq 10$). If a large number of clusters is required (i.e. an over-segmentation), longer running times will have to be expected.

6 Conclusions

We have presented a clustering model for multivariate time series that represents temporally coherent sequences of clustering solutions. At each time index, the cluster structure is represented by a Bayesian mixture model, with a DP prior controlling the number of clusters. A conjugate model structure is used to propagate information along the time axis in a Bayesian framework. Two such structures are present in the model, one between parametric components (carrying parameter information of the mixture components), and one between the nonparametric DP components (carrying cluster structure information). To estimate the model from data, we have introduced an efficient Gibbs sampling approach which draws on both temporal context between frames and a multiscale approach for individual frames to increase efficiency. Applied to video segmentation, the model dynamically adjusts the number of clusters (segments) when new objects appear in or vanish from the scene. The achieved processing times of the sampler are just one order of magnitude below real time for videos in half-PAL format.

We have not provided convergence results for the sampling algorithm, since our analysis of the coarsening operation implies that, for individual component distributions, results for standard Gibbs samplers carry over to the multiscale approach. Any inaccuracies are due to coarsening blocks overlapping segment boundaries, a problem hard to capture by mathematical analysis. Furthermore, our estimation results are necessarily approximate, since the sampler is only run for a few steps on each frame image. Such an algorithm, for which the observed data changes (smoothly) in regular intervals raises some interesting questions. On the one hand, frame changes may pose a problem, if the model has not yet been sufficiently adapted to the current data. On the other hand, small data perturbations may help to avoid local minima. In-depth analysis of the algorithm is beyond the scope of this article.

The results presented here were computed on low-level features, such as color and saturation histograms, which necessarily results in limited precision. Since the model applies to both Gaussian and multinomial feature distributions, it is directly applicable to a wide range of features. Tracking applications, for example, require robustness but no coherent partition of the image. Hence, interest point features could be extracted on each frame and grouped with our model using a Gaussian likelihood supported on the frame image. Since DP models can be constrained by Markov random fields [15], the model itself can be extended by spatial smoothing, as advocated for video segmentation in [2]. Such an extension probably comes at the price of an increase in computation time, as more iterations per frame would be required for the smoothing to take effect.

The DP approach models the number of clusters as a random variable. The model order as an input parameter is replaced by a control parameter that allows the user to adjust the approximate level of cluster resolution. For fixed, static data sets, the approach may not constitute a practical advantage, as it arguable replaces one input parameter by another. We face a different situation for dynamic data, such as videos, where the number of clusters may change over time and the model has to adapt. Adaptation *requires* either a random description of the model order, or a transition heuristic (such as BIC scoring, or reversible jump in a Bayesian framework). Bayesian methods in general, and DP approaches in particular, are often regarded as inapplicable to data-intensive problems due to their computational costs. In our view, the reported results convincingly demonstrate that algorithmic efficiency need not pose an obstacle to the practical application of DP models, if temporal and spatial structure in the data can be exploited.

References

1. Tekalp, A.M.: Video segmentation. In Bovik, A.C., ed.: Handbook of Image and Video Processing. (2000) 383–399
2. Weiss, Y., Adelson, E.H.: A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In: Proc. of IEEE CVPR 1996. (1996)
3. Bregler, C.: Learning and recognizing human dynamics in video sequences. In: Proc. of IEEE CVPR 1997. (1997)
4. Khan, S., Shah, M.: Object based segmentation of video, using color, motion and spatial information. In: Proc. of IEEE CVPR 2001. (2001)
5. Goldberger, J., Greenspan, H.: Context-based segmentation of image sequences. IEEE Trans. on Pattern Analysis and Machine Intelligence **28**(3) (2006) 463–468
6. Bernardo, J.M., Smith, A.F.M.: Bayesian Theory. Wiley (1994)
7. Ferguson, T.S.: A bayesian analysis of some nonparametric problems. Annals of Statistics **1**(2) (1973)
8. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics **9** (2000) 249–265
9. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. Wiley (1997)
10. Stoica, P., Selen, Y.: Model order selection: A review of information criterion rules. IEEE Signal Processing (July 2004) 36–47
11. Lange, T., Roth, V., Braun, M., Buhmann, J.M.: Stability-based validation of clustering solutions. Neural Computation **16**(6) (2004) 1299–1323
12. Walker, S.G., Damien, P., Laud, P.W., Smith, A.F.M.: Bayesian nonparametric inference for random distributions and related functions. Journal of the Royal Statistical Society B **61**(3) (1999) 485–527
13. MacEachern, S.N.: Estimating normal means with a conjugate style Dirichlet process prior. Communications in Statistics: Simulation and Computation **23** (1994) 727–741
14. Higdon, D., Lee, H., Holloman, C.: Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems. In Bernardo et al., J.M., ed.: Bayesian Statistics 7. (2003) 181–198
15. Orbanz, P., Buhmann, J.M.: Nonparametric Bayesian image segmentation. Int. J. of Computer Vision. To appear.