# Manipulation Pattern Discovery: A Nonparametric Bayesian Approach

Bingbing Ni
Advanced Digital Sciences Center
Singapore 138632
bingbing.ni@adsc.com.sg

Pierre Moulin
University of Illinois at Urbana-Champaign
Urbana, IL 61801
moulin@ifp.uiuc.edu

## Abstract

*We aim to unsupervisedly discover human's action (motion) patterns of manipulating various objects in scenarios such as assisted living. We are motivated by two key observations. First, large variation exists in motion patterns associated with various types of objects being manipulated, thus manually defining motion primitives is infeasible. Second, some motion patterns are shared among different objects being manipulated while others are object specific. We therefore propose a nonparametric Bayesian method that adopts a hierarchical Dirichlet process prior to learn representative manipulation (motion) patterns in an unsupervised manner. Taking easy-to-obtain object detection score maps and dense motion trajectories as inputs, the proposed probabilistic model can discover motion pattern groups associated with different types of objects being manipulated with a shared manipulation pattern dictionary. The size of the learned dictionary is automatically inferred. Comprehensive experiments on two assisted living benchmarks and a cooking motion dataset demonstrate superiority of our learned manipulation pattern dictionary in representing manipulation actions for recognition.*

## 1. Introduction

Understanding manipulation actions is attracting increasing interest from the computer vision community given its promising applications in assisted living, smart surveillance, human-robot interaction, work-flow optimization, etc. The fundamental task is to characterize and model manipulation action (motion) patterns, i.e., how human interact with different objects. We make several observations. First, given different objects being manipulated (i.e., object in use), there exist specific interaction patterns. For example, in daily living: *dial phone*, *use silverware*, etc. or in cooking: *break egg*, *mix vegetable*, etc., the action (motion) patterns associated with these object manipulations are quite distinctive, due to their different functionalities. Second, many types of motion patterns are shared among different human-object interactions. For example, there is **NOT** much perceivable difference between the motions *pick up banana* or *pick up cup*. Last but not least, manipulation actions have large variations due to the diversity of human and object and it is general hard to specify in prior how many types of manipulation patterns are of particular interest (i.e., representative). Therefore, manually defining a set of manipulation primitives are infeasible for realistic applications. There exist some previous works on manipulation action recognition. Yang et al. [20] proposed a concept of *consequences of actions* in understanding manipulation actions. The method monitors the appearance and topological structure of the manipulated object and uses a visual semantic graph (VSG) to recognize action consequences. In both [6] and [7] object classification and action understanding are performed jointly by exploring the mutual context and interaction between object and action. Similarly, Yao et al. [21] jointly detect human pose and object in still images. Moore et al. [11] introduced a hand centric action recognition framework using HMM by taking positions of the de-

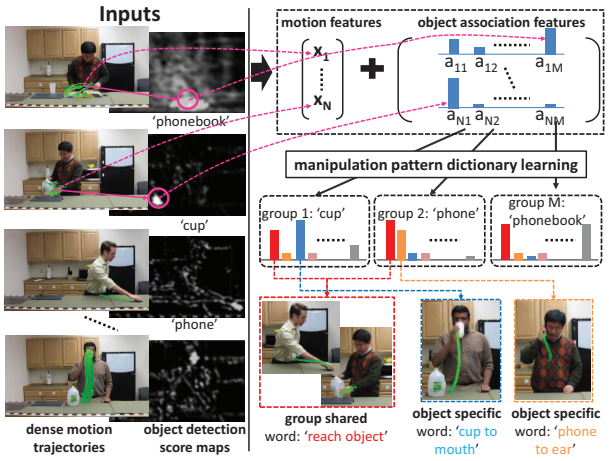


Figure 1. Overview of our method. Inputs are paired motion features $\{\mathbf{x}_i\}_{i=1,\cdots,N}$ and the corresponding (i.e., surrounding) object association features $\{\mathbf{a}_i\}_{i=1,\cdots,N}$. Our contribution is a nonparametric Bayesian approach (unsupervised) that learns grouped (according to the type of object being manipulated) and representative manipulation pattern (i.e., **manipulation words**) dictionary, including shared and object specific words. Best view in color.

tected object and hand as observations. Messing et al. [10] proposed a daily activity (which involves various manipulation actions) recognition method based on velocity history of tracked key points.

However, limited attention has been paid to how to discover and characterize representative manipulation action (motion) patterns associated with different objects from realistic action sequences in an unsupervised manner. On the one hand, to achieve an action representation, previous works mostly manually categorize manipulation motion into a few *action primitives*. For instance, in [8] manipulation is divided into four types of individual motor primitives including *approach*, *retreat*, *push*, and *rotate*; and in [6] human object interaction is categorized into four classes, i.e., *object perception*, *reaching*, *manipulation* and *object reaction*. Given the large variation and rich content of manipulation actions caused by the diversity of human and object, it is not feasible to define manipulation pattern types manually and thus an unsupervised way to automatically discover representative manipulation patterns from realistic video sequences (without action labels or annotations) is demanded. On the other hand, previous methods [11, 6, 7] use tracked hand trajectories or object detections for representing actions. However, both hand trajectory and object detection are difficult to obtain reliably in complex scenarios. False or missing detection and tracking can severely harm action recognition accuracy. Also, knowing hand position and pose alone cannot provide very detailed (i.e., fine-grained) motion information since some important and informative motions and interactions are usually subtle. Thus these methods are generally incapable of automatically discovering representative manipulation patterns.

To address above mentioned issues, we propose a probabilistic framework to discover representative manipulation patterns as follows (illustrated in Figure 1). First, instead of explicitly tracking hand movement, we extract dense motion trajectories proposed recently in [15], given that: 1) dense motion trajectories are easily extracted; and 2) they describe local motions in more detail than hand position and pose, which not only include subtle hand motion but also the movement associated with the object being manipulated and thus richer manipulation patterns can be captured. Second, to obtain the object in use information, instead of explicitly detecting (i.e., localizing) the object of interest, we compute the object detection score maps and augment each extracted motion trajectory with its surrounding object detection scores (denoted as **object association features**). We can view this combination of motion trajectory and object detection as a probabilistic (or *soft*) association which is less sensitive to false or missing detection and tracking of either hand or object being manipulated. Taking this paired motion and object association features as input, our key contribution is a nonparametric Bayesian approach to learn a dictionary (denoted as **manipulation dictionary**) of representative object manipulation patterns (denoted as **manipulation words**) in an unsupervised manner. Adopting a hierarchical Dirichlet process (HDP) prior [14], our generative model can automatically discover and model the shared manipulation patterns among different objects being manipulated as well as object specific manipulation patterns. The size of the manipulation pattern dictionary is also inferred. We then derive a Gibbs sampling scheme for learning the proposed probabilistic model. The learned manipulation dictionary is utilized for action representation. The novelty of our HDP model is that it not only performs feature clustering (HDP) but also performs multi-modal (object map and motion feature) association/fusion. This results in local and more detailed object-use words (conveys richer object-use information) due to the designed geometric rule for linking motion and object map. Comprehensive experiments in two assisted living benchmarks and a cooking motion dataset demonstrate that our method possesses the superior capability in representing manipulation patterns for action recognition.

## 2. Related Works

Our method is partly inspired by the latent topic models for action recognition [12]. However, our model aims to discover object manipulation patterns but [12] only models motion features, e.g., STIPs [9, 3]. Also, using nonparametric Bayesian can avoid the difficulty in selecting optimal dictionary size, which is a key parameter in [12] that greatly affects action recognition accuracy.

Wang et. al [18] used hierarchical nonparametric Bayesian models for crowd analysis. Atomic activities are modeled as distributions over low-level visual features and multi-agent interactions are modeled as distributions over atomic activities, respectively. While their work only focuses on traffic (crowd behavior) analysis, we propose to use nonparametric Bayesian for discovering representative object manipulation patterns.

Packer et al. [13] presented a system that is able to recognize complex, fine-grained human actions involving the manipulation of objects in cooking action sequences. However, Kinect-type depth camera is required for providing very accurate human pose trajectories and performing explicit object segmentation and localization, which limits its application scenario. On the contrary, inputs into our method are just object detection maps and dense motion trajectories, which are very easy to obtain. Moreover, in contrast to the discriminative approach of [13], our focus is to automatically discover representative manipulation patterns given unlabeled video sequences. Finally, in [13] the object-use information is modeled in a very coarse/global way which only encodes co-occurrence between motion feature and object instance, which is not sufficient for fine-grained action.

## 3. Our Methodology

As introduced in the previous section, instead of explicitly tracking hand position and pose and exactly localizing the object of interest, we propose to first extract dense motion trajectories, compute object detection score maps from the image sequences and then associate both types of features in a probabilistic (i.e., generative) framework, for the purpose of discovering representative object manipulation action (motion) patterns. The advantages are as follows: 1) it is much easier to extract proposed features than explicit hand tracking and object detection; 2) dense motion trajectories capture richer and fine-grained motion information than hand trajectory; and 3) using a probabilistic framework for linking motion and object association features can achieve a better representation which is less sensitive to false or missing object detection and inaccurate hand tracking. We give detailed descriptions as follows.

### 3.1. Manipulation Feature Extraction

Histogram of oriented gradients (HOG) detector [2] has been widely used for object and human detection, which we also adopt in this work. In practice, when the object being manipulated is of too small size or deformable, HOG based detector gives degraded detection performance. Therefore, we also utilize recently proposed hough forest based object detector [5], which gives better detection accuracy for objects with small size or partial deformations. We run two detectors on input video frames and the resulting two detection score maps are normalized to the range of $[-1, 1]$ and then fused by weighted sum.

To extract motion features, we adopt the recently developed dense motion trajectories [15]. Dense motion trajectories are very easy and efficient to extract and they capture detailed local motion information than hand position and pose sequence. The study in [15] showed dense motion trajectories achieve state-of-the-art recognition accuracies on several human action benchmarks [15]. As in [15], we use a motion boundary histogram (MBH) descriptor and a trajectory-aligned (TA) descriptor (i.e., we drop the less discriminative HOF and HOG descriptors used in [15]) to represent a motion trajectory. Trajectory length is fixed to $l = 15$ frames throughout all experiments.

To **associate a motion trajectory $i$ to $j$-th type of object being manipulated**, we do as follows. For each point along an extracted motion trajectory $i$, we calculate the average object detection score of the neighborhood patch centered at this point (i.e., within a radius of 10 pixels) in the corresponding detection score map. We then average these values over all points along trajectory to value $a_{ij}$, which indicates the strength of the association of motion trajectory $i$ to the $j$-th type of object. Assume we have $M$ types of objects of interest, then for motion trajectory $i$, we can denote its **object association feature vector** as

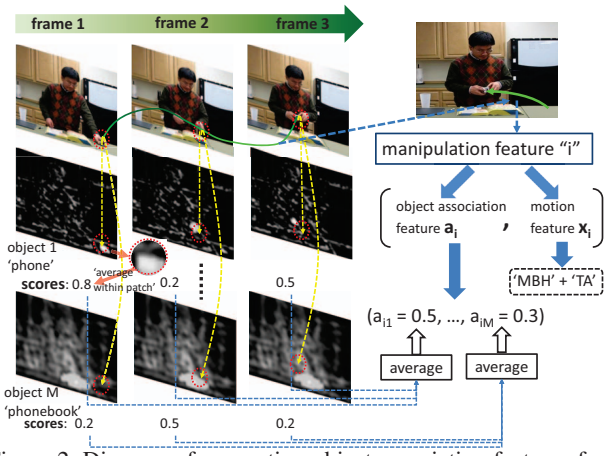


Figure 2. Diagram of computing object association features for a motion trajectory of length $l = 3$. Best view in color.

$\mathbf{a}_i = (a_{i1}, a_{i2}, \cdots, a_{iM})^T$, i.e., a concatenation of the object association values for all $M$ types of objects. Figure 2 illustrates this motion trajectory and object in use (i.e., being manipulated) association scheme. We denote the motion feature vector for trajectory $i$ as $\mathbf{x}_i$ (e.g., a $D$-dimensional vector composed of MBH and TA descriptors). We further denote the pair $(\mathbf{x}_i, \mathbf{a}_i)$ as the $i$-th observed object **manipulation feature**.

### 3.2. Unsupervised Manipulation Pattern Discovery

Assume that from training video set, we obtain $N$ object manipulation features (pairs) $\mathcal{X} = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=1,\cdots,N}$, i.e., observation data. Our task is to learn a dictionary of representative manipulation patterns (manipulation words) which are capable of describing various manipulation actions. We have the following observations. Some manipulation motions are general (i.e., object independent) such as *pick up/put down object* and thus the learned manipulation words associated with these motions should be shared among different types of object being manipulated. Other manipulation motions are object specific such as *cutting on the chopping board*, *phone to ear* etc. Also, it is in general unknown how many manipulation words are sufficient for well describing various actions. These observations motivate us to utilize hierarchical Dirichlet process (HDP) mixture models [14]. HDP mixture models consider input of groups of data and learn a dictionary of words (mixture components) that are shared among groups. In our case, a group can be naturally considered as manipulation patterns associated with a type of object being manipulated. HDP specifies different distributions over the mixture proportions for different groups, and this well matches our problem: some manipulation words (i.e., mixture components) are shared by different object (being manipulated) groups while others are only possessed by a specific group. Also, the merit of HDP (and other nonparametric Bayesian approaches) is that the dictionary size can be automatically

inferred.

We use HDP mixture models as our prior distribution for motion features $\{\mathbf{x}_i\}_{i=1,\cdots,N}$. Object association features $\{\mathbf{a}_i\}_{i=1,\cdots,N}$ are considered as another evidence for grouping motion features. We introduce a set of variables $\mathcal{S} = [s_{ij}]_{i=1,\cdots,N;j=1,\cdots,M}$ to indicate the group assignment, namely, $s_{ij} = 1$ means that motion feature $\mathbf{x}_i$ is associated with object group $j$. Note that one motion feature can be simultaneously assigned to more than one object group. This is natural as a manipulation motion sometimes involves several objects. We propose a probabilistic model to utilize motion features and the corresponding object association features in a collaborative way for manipulation dictionary learning. The graphical representation of our probabilistic model is illustrated in 3(a). The generative process of this model is described as follows.

1. Sample the global random probability measure $G_0$ from the Dirichlet process: $G_0 \sim DP(\gamma, H)$. The base distribution $H$ is a $D$-dimensional (i.e., dimensionality of motion feature $\mathbf{x}$) Gaussian with zero mean and covariance $\Sigma_0$, i.e., $h(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \Sigma_0)$. In this paper, we use upper case alphabet to denote probability distribution, e.g., $H(.)$ and the corresponding lower case to denote its density function, e.g., $h(.)$. The scalar $\gamma$ is the concentration parameter.

2. For each (object) group $j = 1, \cdots, M$, sample a random probability measure $G_j$ from the Dirichlet process: $G_j \sim DP(\alpha, G_0)$, by taking $G_0$ as the base distribution. The scalar $\alpha$ is the concentration parameter.

3. For each input motion feature $\mathbf{x}_i, i = 1, \cdots, N$ and for each object association score $a_{ij}, j = 1, \cdots, M$, sample the corresponding group assignment indicator $s_{ij}$ from the binomial distribution: $s_{ij} \sim (P(s_{ij} = 1|a_{ij}), P(s_{ij} = 0|a_{ij}))$. Here $P(s_{ij} = 1|a_{ij})$ is defined as a sigmoid function:

$$P(s_{ij} = 1|a_{ij}) = \frac{1}{1 + \exp(-a_{ij})}. \quad (1)$$

   (a) If $s_{ij} = 0$ (i.e., $\mathbf{x}_i$ belongs to background motion), sample $\mathbf{x}_i$ according to a universal background motion distribution (UBM) [19]: $\mathbf{x}_i \sim p(\mathbf{x}_i|s_{ij} = 0)$. Similar as in [19], we use Gaussian mixture models (GMM) to define the UBM. The GMM parameters (mixture weights, means, covariances) are denoted as $\boldsymbol{\xi} = \{\{w_d\}_{d=1,\cdots,U}; \{\boldsymbol{\mu}_d\}_{d=1,\cdots,U}; \{\sigma_d\}_{d=1,\cdots,U}\}$, i.e., the number of mixtures is $U = 1000$. We assume independent and isotropic variances for all feature dimensions. We off-line train the UBM parameter set $\boldsymbol{\xi}$ on all training motion features $\{\mathbf{x}_i\}_{i=1,\cdots,N}$ using EM algorithm.

   (b) If $s_{ij} = 1$ (i.e., $\mathbf{x}_i$ associates with (object) group $j$), sample a $D$-dimensional factor (i.e., manipulation word) $\boldsymbol{\theta}_{ij}$ according to distribution $G_j$. Then sample the motion feature observation $\mathbf{x}_i$ according to the likelihood $\mathbf{x}_i \sim p(\mathbf{x}_i|\boldsymbol{\theta}_{ij})$, which is a Gaussian $p(\mathbf{x}_i|\boldsymbol{\theta}_{ij}) = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}_{ij}, \Sigma)$ centered at $\boldsymbol{\theta}_{ij}$ with covariance $\Sigma$.

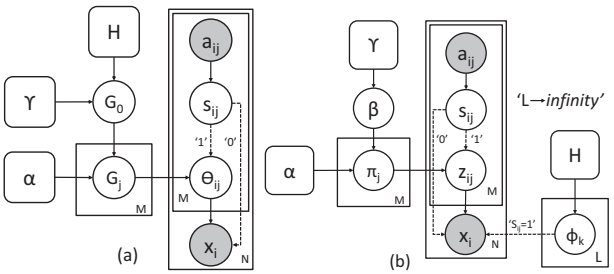

Figure 3. Graphical models of our manipulation dictionary learning method. (a) our model based on HDP mixture models; and (b) the equivalent model based on infinite mixture models. Observed data in gray.

To facilitate model learning, we derive an equivalent infinite mixture based model (i.e., $L$ mixture components and $L \to \infty$) as shown in Figure 3(b). A mixture component corresponds to a manipulation word. We use $K$ to denote the effective dictionary size, i.e., the number of mixture components with associated observations from training data $\{\mathbf{x}_i\}_{1,\cdots,N}$. The remaining $L - K$ components have no observation associated to them. We introduce a set of variables $\mathcal{Z} = [z_{ij}]_{i=1,\cdots,N;j=1,\cdots,M}$ to indicate the mixture component index associated with each observed feature $\mathbf{x}_i$, i.e., $z_{ij} = k, k = 1, \cdots, L$ means $\mathbf{x}_i$ is assigned to $k$-th mixture component. We explain various terms as follows (probability terms with the same definitions explained in the generative process are omitted). We denote $\boldsymbol{\beta}$ as the global vector of mixing proportions shared by $M$ groups (i.e., object types) which follows the Dirichlet distribution:

$$P(\boldsymbol{\beta}|\gamma) = Dir(\boldsymbol{\beta}|\gamma, L) = Dir(\frac{\gamma}{L}, \cdots, \frac{\gamma}{L}). \quad (2)$$

We can also write $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_K, \beta_u)$, namely, the first $K$ values correspond to the mixture components with observations, and $\beta_u$ denotes the sum of mixture proportions without any observed data, i.e., $\beta_u = \sum_{k=K+1}^{L} \beta_k$. We define $\boldsymbol{\pi}_j$ as the $j$-th ($j = 1, 2, \cdots, M$) object group specific vector of mixing proportions that is sampled from the global Dirichlet distribution parameterized by $\boldsymbol{\beta}$:

$$P(\boldsymbol{\pi}_j|\alpha, \boldsymbol{\beta}) = Dir(\boldsymbol{\pi}_j|\alpha, \boldsymbol{\beta}) = Dir(\alpha\beta_1, \cdots, \alpha\beta_L). \quad (3)$$

$\boldsymbol{\Pi} = \{\boldsymbol{\pi}_j\}_{j=1,\cdots,M}$. If data $\mathbf{x}_i$ is assigned to $j$-th group, i.e., $s_{ij} = 1$, we sample a mixture component index $z_{ij}$ according to the multinomial distribution parameterized by $\boldsymbol{\pi}_j$ (i.e., for $j$-th object group):

$$P(z_{ij}|\boldsymbol{\pi}_j, s_{ij} = 1) = Multi(z_{ij}|\boldsymbol{\pi}_j). \quad (4)$$

The set of mixture components (i.e., dictionary of manipulation words) are denoted as $\mathbf{\Phi} = \{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K\}$, which are sampled from base distribution $H$, i.e., the Gaussian with density function $p(\boldsymbol{\phi}) = h(\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\phi}|\mathbf{0}, \Sigma_0)$. The likelihood of $\mathbf{x}_i$ given $z_{ij}$ and the dictionary $\mathbf{\Phi}$ is:

$$p(\mathbf{x}_i|z_{ij}, \mathbf{\Phi}) = f(\mathbf{x}_i|\boldsymbol{\phi}_{z_{ij}}) = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\phi}_{z_{ij}}, \Sigma). \quad (5)$$

The full joint probability of this infinite mixture based model can be expressed as:

$$\begin{aligned} &p(\mathcal{X}, \mathbf{\Phi}, \boldsymbol{\xi}, \mathbf{\Pi}, \mathcal{Z}, \mathcal{S}, \boldsymbol{\beta}, \gamma, \alpha) \\ &= P(\boldsymbol{\beta}|\gamma) \times \prod_j^M P(\boldsymbol{\pi}_j|\alpha, \boldsymbol{\beta}) \times \prod_k^L p(\boldsymbol{\phi}_k) \\ &\times \prod_{i,j}^{N,M} \{P(s_{ij} = 1|a_{ij}) P(z_{ij}|\boldsymbol{\pi}_j, s_{ij} = 1) p(\mathbf{x}_i|z_{ij}, \mathbf{\Phi})\}^{s_{ij}} \\ &\times \prod_{i,j}^{N,M} \{P(s_{ij} = 0|a_{ij}) p(\mathbf{x}_i|s_{ij} = 0)\}^{(1-s_{ij})}. \end{aligned} \quad (6)$$

Similar as in [14], it can be easily shown that the limit (i.e., $L \to \infty$) of the above model is equivalent to the aforementioned HDP mixture based model.

We derive a Gibbs sampling procedure to learn the above model (i.e., posterior sampling $\mathcal{S}$, $\mathcal{Z}$, $\mathbf{\Phi}$ and $\boldsymbol{\beta}$ given observations $\mathcal{X} = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=1,\cdots,N}$) by marginalizing out $\mathbf{\Pi}$ and setting $L \to \infty$. Each time, we sample one variable in $\{s_{ij}\}$, $\{z_{ij}\}$, $\{\boldsymbol{\phi}_k\}$ and $\boldsymbol{\beta}$ alternatively, with the reminder of the variables fixed to their old values. We first give the notations of counts maintained throughout the sampling procedure. We denote $N_j$ as the number of observed data $\mathbf{x}_i$ assigned to $j$-th object group, i,e., $N_j = \sum_i s_{ij}$. We denote $n_{j,k}$ as the number of observed data assigned to $j$-th object group while it corresponds to $k$-th mixture component (manipulation word), i.e. $n_{j,k} = \sum_{i:z_{ij}=k} s_{ij}$. We denote $m_k$ as the number of observed data associated to $k$-th mixture component for all groups, i.e., $m_k = \sum_j n_{j,k}$. When a subscript is attached to a notation, it means that the variable to the superscripted index is removed from the set or from the calculation of the count, e.g., $\mathcal{S}^{(-ij)}$ denotes the set $\mathcal{S}$ with $s_{ij}$ removed and $N_j^{(-ij)}$ means when counting $N_j$, the observed data $\mathbf{x}_i$ is removed. For the following derived probabilities, constant normalization factors are omitted where applicable, for notational simplicity.

**Sampling** $s_{ij}$. The posterior distribution of $s_{ij}$ given the reminder of the variables is as follows:

$$\begin{aligned} &P(s_{ij}|a_{ij}, \mathbf{x}_i, \mathbf{\Phi}, \boldsymbol{\xi}, \mathcal{Z}^{(-ij)}, \mathcal{S}^{(-ij)}, \boldsymbol{\beta}, \alpha) \\ &\sim \begin{cases} P(s_{ij} = 0|a_{ij}) p(\mathbf{x}_i|s_{ij} = 0), & s_{ij} = 0; \\ P(s_{ij} = 1|a_{ij}) \sum_k r_{j,k}^{(-ij)} q_k(\mathbf{x}_i), & s_{ij} = 1. \end{cases} \end{aligned} \quad (7)$$

Here $r_{j,k}^{(-ij)}$ is defined as:

$$r_{j,k}^{(-ij)} = \begin{cases} \frac{\alpha\beta_k + n_{j,k}^{(-ij)}}{\alpha + N_j^{(-ij)}}, & k \le K; \\ \frac{\alpha\beta_u}{\alpha + N_j^{(-ij)}}, & k = k^{new}. \end{cases} \quad (8)$$

and $q_k(\mathbf{x}_i)$ is defined as:

$$q_k(\mathbf{x}_i) = \begin{cases} f(\mathbf{x}_i|\boldsymbol{\phi}_k), & k \le K; \\ \int f(\mathbf{x}_i|\boldsymbol{\phi}) h(\boldsymbol{\phi}) d\boldsymbol{\phi}, & k = k^{new}. \end{cases} \quad (9)$$

We can regard $\tilde{q}(\mathbf{x}_i|s_{ij} = 1) = \sum_k r_{j,k}^{(-ij)} q_k(\mathbf{x}_i)$ as a foreground probability density since it's a weighted sum (expectation) of the likelihood value that $\mathbf{x}_i$ is either assigned to mixture component $k < K$, or not assigned to any existing mixture component, i.e., $k = k^{new}$. Thus the posterior is composed by two parts. The first part is $P(s_{ij}|a_{ij})$ which can be considered as the evidence for motion feature grouping provided by $a_{ij}$; the second part corresponds to either the UBM $p(\mathbf{x}_i|s_{ij} = 0)$ or the foreground probability density $\tilde{q}(\mathbf{x}_i|s_{ij} = 1)$, which can be considered as the evidence for grouping provided by $\mathbf{x}_i$. This is an elegant point of our model since a motion feature $\mathbf{x}_i$ is linked with an object group in a probabilistic way, by taking consideration of information coming from both $\mathbf{x}_i$ and $\mathbf{a}_i$.

**Sampling** $z_{ij}$. If $s_{ij} = 1$, we then sample an associated mixture component index $z_{ij}$ according to the posterior distribution as follows:

$$\begin{aligned} &P(z_{ij} = k|\mathbf{x}_i, s_{ij} = 1, \mathbf{\Phi}, \mathcal{Z}^{(-ij)}, \mathcal{S}^{(-ij)}, \boldsymbol{\beta}, \alpha) \\ &\sim \begin{cases} \frac{\alpha\beta_k + n_{j,k}^{(-ij)}}{\alpha + N_j^{(-ij)}} f(\mathbf{x}_i|\boldsymbol{\phi}_k), & k \le K; \\ \frac{\alpha\beta_u}{\alpha_0 + N_j^{(-ij)}} \int f(\mathbf{x}_i|\boldsymbol{\phi}) h(\boldsymbol{\phi}) d\boldsymbol{\phi}, & k = k^{new}. \end{cases} \end{aligned} \quad (10)$$

If the generated $z_{ij}$ is associated to a newly issued component $k_{new}$, we then draw a new mixture component $\boldsymbol{\phi}_k$ and add it to the set $\mathbf{\Phi}$ according to:

$$p(\boldsymbol{\phi}_k|z_{ij} = k_{new}, \mathbf{x}_i) \sim f(\mathbf{x}_i|\boldsymbol{\phi}_k) h(\boldsymbol{\phi}_k), \quad (11)$$

and we let $K = K + 1$.

**Sampling** $\boldsymbol{\phi}_k$. The posterior probability density for $\boldsymbol{\phi}_k, k = 1, \cdots, K$ is given as:

$$p(\boldsymbol{\phi}_k|\mathcal{X}, \mathcal{Z}, \mathcal{S}) \sim h(\boldsymbol{\phi}_k) \prod_{ij:z_{ij}=k \bigwedge s_{ij}=1} f(\mathbf{x}_i|\boldsymbol{\phi}_k). \quad (12)$$

**Sampling** $\boldsymbol{\beta}$. The posterior distribution of $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_K, \beta_u)^T$ is given as:

$$p(\beta_1, \cdots, \beta_K, \beta_u|\mathcal{Z}, \mathcal{S}, \gamma) \sim Dir(m_1, m_2, \cdots, m_K, \gamma). \quad (13)$$

Note that although the hyper parameters $\alpha$, $\gamma$ can also be learned by sampling, for the trade-off between effectiveness and efficiency, we fix their values as 100 in all experiments.

### 3.3. Manipulation Action Sequence Representation

Our learned dictionary can be represented as the estimated parameter sets $\widehat{\mathbf{\Phi}} = \{\widehat{\phi_k}\}_{k=1,\cdots,\widehat{K}}$, $\widehat{\beta}$, $\widehat{\boldsymbol{\xi}}$ and the estimated values for hidden variable sets $\widehat{\mathcal{S}}$ and $\widehat{\mathcal{Z}}$. The inferred effective number of mixture components (manipulation words) is $\widehat{K}$. For simplicity, we denote $\widehat{\mathbf{\Omega}} = \{\widehat{\mathbf{\Phi}}, \widehat{\beta}, \widehat{\boldsymbol{\xi}}, \widehat{\mathcal{S}}, \widehat{\mathcal{Z}}\}$. Given a testing video sequence, i.e., with a set of extracted object manipulation features (i.e., motion and object association feature pairs) $\mathcal{X}_t = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=1,\cdots,t}$, the task of action sequence representation is to map the feature set $\mathcal{X}_t$ onto a video sequence level representation vector, which can be input into the subsequent action classifier. Towards this end, we consider the joint conditional probability $p(s_{ij} = 1, z_{ij} = k|\mathbf{x}_i, \mathbf{a}_i)$, which represents the probability of assigning the observed manipulation feature $(\mathbf{x}_i, \mathbf{a}_i)$ to object group $j$ and manipulation word $k$. Given the observations $\mathcal{X}_t$ and the trained dictionary $\widehat{\mathbf{\Omega}}$, we therefore estimate joint posterior probabilities for $\psi(j,k) = p(s_{ij} = 1, z_{ij} = k|\mathcal{X}_t, \widehat{\mathbf{\Omega}}), j = 1,\cdots,M, k = 1,\cdots,\widehat{K}$ and concatenate them to form a $M \times \widehat{K}$-dimensional representation vector $\boldsymbol{\psi}$. An estimation of $\psi(j,k)$ can be derived as:

$$\widehat{\psi(j,k)} = \frac{1}{t}\sum_{i=1}^{t} p(s_{ij} = 1|\mathbf{x}_i, a_{ij}, \widehat{\mathbf{\Omega}}) p(z_{ij} = k|\mathbf{x}_i, s_{ij} = 1, \widehat{\mathbf{\Omega}}). \tag{14}$$

where the calculation of $p(s_{ij} = 1|\mathbf{x}_i, a_{ij}, \widehat{\mathbf{\Omega}})$ and $p(z_{ij} = k|\mathbf{x}_i, s_{ij} = 1, \widehat{\mathbf{\Omega}})$ follow directly from Eqn. 7 and Eqn. 10, respectively. In the meantime, background motions (i.e., $s_{ij} = 0$) also provide useful information of the whole body movement, we thus represent them into another representation vector $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_U)^T$ to complement the manipulation action representation $\boldsymbol{\psi}$, where each $\eta_d, d = 1,\cdots,U$ can be estimated as:

$$\widehat{\eta_d} = \frac{1}{tM}\sum_{i=1}^{t}\sum_{j=1}^{M} p(s_{ij} = 0|a_{ij}) w_d \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_d, \sigma_d I). \tag{15}$$

The final representation of an action video sequence is denoted as $(\boldsymbol{\psi}^T, \boldsymbol{\eta}^T)^T$. We use linear SVM, i.e., liblinear [4] for action classification. The penalty parameter $\mathcal{C}$ for SVM is set as 1000 in all experiments.

## 4. Experiments

### 4.1. Manipulation Action Recognition in Assisted Living

We apply our method for manipulation action recognition on two assisted daily living benchmarks including: the University of Rochester Assisted Daily Living dataset (URADL) [10] and the Microsoft Research Daily Activity 3D dataset (MSRDA3D) [17]. Both datasets contains very rich object manipulation actions. For URADL the objects of interest (for which we compute the detection maps) are *phone*, *cup*, *phonebook*, *snack*, *plate and silverware*, *banana*, *chopping board*, i.e., $M = 7$; for MSRDA3D are *cup*, *phone*, *laptop*, *book*, *game controller*, *vacuum cleaner*, *guitar*, *snack*, i.e., $M = 8$, respectively. The learned effective number of mixture components $K$ (i.e., dictionary size) for both datasets are 398 and 466, respectively. Figure 4 illustrates the learned posterior distributions of $z_{ij}$ for different object groups. Also, we show some example frames with manipulation patterns according to different manipulation words. From Figure 4, we can observe that some manipulation words are shared among different object groups and others are object specific, which demonstrates our basic idea.

To demonstrate our method's capability in representing manipulation actions, we compare our method with the following methods in terms of action classification accuracy.

1. **BOW + DT**: the bag-of-words method based on dense trajectory features proposed in [15]. We use the implementation provided by the author [15] and follow the settings in [15]. The size of the visual word dictionary is 1000, which achieves the best accuracy in experiment.

2. **pLSA + DT**: the probabilistic latent semantic analysis (pLSA) based method proposed in [12]. Instead of using STIPs [3], we use dense trajectory features (MBH + TA) to train the dictionary, because our off-line result has shown that dense trajectories significantly outperform STIPs. The superiority of dense trajectory based features was also well demonstrated in [15].

3. **Obj + DT**: to demonstrate the superiority of our method in terms of motion and object association, we compare with a *naive* combination of dense trajectory features and object detection features. Namely, we calculate the histogram of different object detection scores over the sequence and augment this histogram with the aforementioned **BOW + DT** feature representation. SVM is used for classification with the same kernel as in [15].

4. **Obj + Hand**: to demonstrate our method's advantages in terms of 1) the richer motion information encoded by the learned manipulation pattern dictionary and 2) the probabilistic framework of motion and object association, we also compare with the simultaneous action and object detection method proposed by Kjellström et al. [7]. In [7], hand is detected explicitly and only its $2D$ position and pose information are extracted, and objects are deterministically associated to the hand based on spatial distance.

For all comparing algorithms, their corresponding parameters (e.g., dictionary size, SVM parameters etc.) are set optimally based on the validation on a subset of the training data, if not otherwise specified. Leave-one-person-out classification accuracies are shown in Table 1. For MSRDA3D, some action classes do not contain manipula-

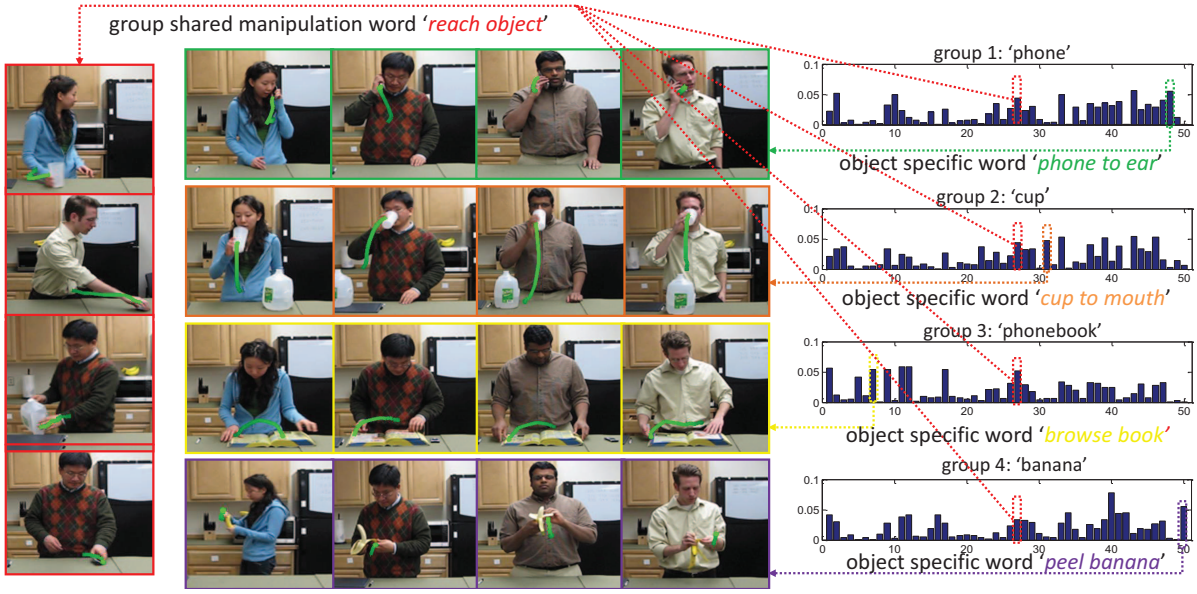


Figure 4. Examples of posterior distributions on $z_{ij}$ (i.e., mixture proportions) on URADL for different object groups (right most column, due to limited space, we only show distributions of the first 50 manipulation words. Horizontal and vertical axes correspond to word index and probability, respectively). Example frames with motion trajectories according to different manipulation words (in different colors) are shown, in terms of both group shared words (left most column) and object specific words (middle four rows). Best view in color.

Table 1. Recognition accuracy comparison for URADL and MSRDA3D datasets. For MSRDA3D, we report accuracies on the whole dataset and on the subset including 8 classes of manipulation actions (i.e., in the bracket).

| Method | URADL | MSRDA3D (subset) |
|---|---|---|
| BOW + DT | 87.3% | 79.4% (81.3%) |
| pLSA + DT | 85.3% | 77.5% (80.0%) |
| Obj + DT | 92.7% | 80.6% (87.5%) |
| Obj + Hand | 82.7% | 74.4% (85.0%) |
| Our Method | **98.0%** | **86.9%** (**96.9%**) |

Table 2. Recognition accuracy comparison for URADL and MSRDA3D datasets.

| URADL | | |
|---|---|---|
| Messing et al. [10] | Wang et al. [16] | Our Method |
| 89% | 96% | **98.0%** |
| **MSRDA3D** | | |
| Wang et al. [17] | Our Method | Our Method + Joints |
| 85.8% | **86.9%** | **91.3%** |

tion actions. Therefore, besides reporting accuracies on the whole dataset, we also report accuracies on the subset which only contains 8 classes of manipulation actions including *drink*, *eat*, *read book*, *call cellphone*, *use laptop*, *use vacuum cleaner*, *play game* and *play guitar*. From Table 1, we can observe: 1) dense trajectory feature based methods generally outperform hand trajectory based method because missing or false detection and tracking of hand and object degrades recognition performance and subtle local manipulation motion patterns cannot be captured by hand movement; 2) using object detection scores to complement motion features improves classification accuracy, which is **NOT** a surprise since objects provide important contextual information; however, our proposed method significantly outperforms this naive combination. The reason is that our derived probabilistic model provides a much better motion and object association scheme and our method can automatically discover representative common and object specific manipulation patterns thus achieving higher discriminative capability; and 3) in URADL, when a subject is manipulating an object (e.g., phone), other objects (e.g., phonebook, cup, banana) are also in the scene, therefore the performance of directly augmenting motion features with histograms of object detection scores is degraded by the presence of these not-in-use objects. However, our method takes the paired motion and object association features, and not-in-use objects cannot affect our method as their association scores with any motions are low.

We also compare the state-of-the-art recognition accuracies on both datasets in Table 2. For all comparing methods, as the experimental settings (train/test partition) are exactly the same, we directly cite the reported accuracies in their respective publications. For URADL, the comparing methods include: 1) Messing et al. [10] which uses velocity histories of tracked keypoints for recognition; and 2) Wang et al. [16] which utilizes multi-scale spatio-temporal contexts. For MSRDA3D, we compare the method by Wang et al. [17] which is based on the mined *actionlets*. This method heavily relies on 3D body joints information extracted by depth camera (a strong feature), and therefore for fair comparison, we also test our method by augmenting the *only joint position features* in [17] with our representation to boost the performance. Results shown in Table 2 demonstrate the superior performances of our method.

Table 3. Recognition performance (mean F-score for all classes) comparison for KSCGR dataset.

| Method | BOW + DT | pLSA + DT | Obj + DT | Obj + Hand | Doman and Kuai [1] | Packer et al. [13] | Our Method |
|---|---|---|---|---|---|---|---|
| Mean F-score | 0.61 | 0.55 | 0.64 | 0.59 | 0.74 | 0.71 | **0.79** |

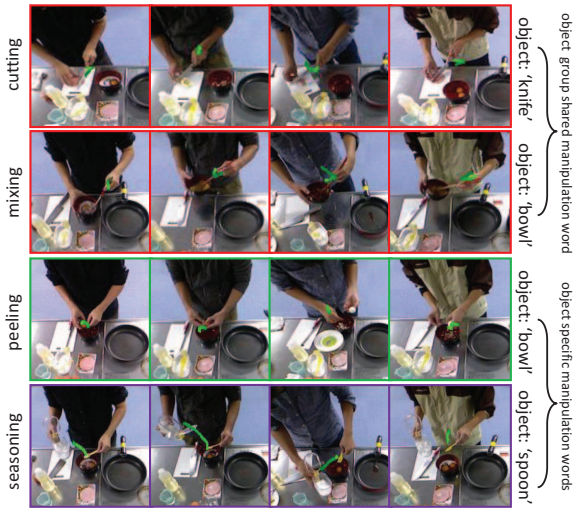


Figure 5. Example frames with motion trajectories according to different manipulation words (in different colors). Both group shared words and object specific words examples are shown.

### 4.2. Cooking Motion Recognition

We also apply our method on cooking motion recognition which contains rich and fine-grained object manipulations. We use the ICPR 2012 Kitchen Scene Context based Gesture Recognition (KSCGR) contest dataset [1], which contains pre-partitioned training and testing set. There are five candidate cooking menus cooked by five different actors. The task is to recognize eight types of cooking motions including: *baking*, *boiling*, *breaking*, *cutting*, *mixing*, *peeling*, *seasoning*, and *turning*. The objects of interest are *pan*, *oil bottle*, *salt bottle*, *bowl*, *knife*, *spoon*, *chopping board*, *egg*, *ham*, i.e., $M = 9$. Examples of learned manipulation words are illustrated in Figure 5 and the inferred $K = 538$. As defined in the contest, the evaluation metric is the mean recognition $F$-score over all motion categories. Besides aforementioned methods, we also compare our method to the best reported result in the contest by Doman and Kuai [1] and state-of-the-art cooking action recognition method developed in [13]. As human skeleton data (3D pose) is not available, to implement [13], we instead use a standard skin color based hand detector to estimation hand position. The list of objects to recognize is the same as in our method. The results are summarized in Table 3. Note that as most not-in-use objects are in the kitchen table, the method based on naive combination of motion trajectory features and object detection score histogram is severely affected. Also, unreliable detection of hand or objects and incapability in encoding fine-grained motions degrade performances of both **Obj + hand** and [13]. In contrast, our method can well models representative manipulation patterns and therefore it achieves the best performance.

## 5. Conclusion

We propose an unsupervised learning framework for discovering representative object manipulation patterns based on nonparametric Bayesian. The learned manipulation pattern dictionary is used for action representation on two assisted daily living benchmarks and a cooking motion dataset. The superiority of our method in representing manipulation action sequence for recognition is demonstrated.

## Acknowledgment

This study was supported by a research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

## References

[1] http://www.murase.m.is.nagoya-u.ac.jp/kscgr/index.html.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[5] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, pages 1022–1029, 2009.

[6] A. Gupta and L. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, pages 1–8, 2007.

[7] H. Kjellström, J. Romero, D. Martínez, and D. Kragić. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, pages 336–349, 2008.

[8] V. Kyrki, I. Vicente, D. Kragic, and J.-O. Eklundh. Action recognition and understanding using motor primitives. In *International Symposium on Robot and Human interactive Communication*, pages 1113–1118, 2007.

[9] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[10] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, pages 104–111, 2009.

[11] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV*, Corfu, Greece, 1999.

[12] J. C. Niebles, H. Wang, and L. Fei-fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[13] B. Packer and D. Koller. A combined pose, object, and feature model for action understanding. In *CVPR*, 2012.

[14] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[15] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[16] J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, pages 3185–3192, 2011.

[17] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297, 2012.

[18] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *T-PAMI*, 31(3):539–555, 2009.

[19] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. Huang. Regression from patch-kernel. In *CVPR*, pages 1–8, 2008.

[20] Y. Yang, C. Fermuller, and Y. Aloimonos. Detection of manipulation action consequences (mac). In *CVPR*, 2013.

[21] B. Yao, A. Khosla, and L. Fei-fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In *ICML*, 2011.