Multivariate Discretization for Bayesian Network Structure Learning in Robot Grasping

Dan Song, Carl Henrik Ek, Kai Huebner and Danica Kragic

Abstract— **A major challenge in modeling with BNs is learning the structure from both discrete and multivariate continuous data. A common approach in such situations is to discretize continuous data before structure learning. However efficient methods to discretize high-dimensional variables are largely lacking. This paper presents a novel method specifically aiming at discretization of high-dimensional, high-correlated data. The method consists of two integrated steps: non-linear dimensionality reduction using sparse Gaussian process latent variable models, and discretization by application of a mixture model. The model is fully probabilistic and capable to facilitate structure learning from discretized data, while at the same time retain the continuous representation. We evaluate the effectiveness of the method in the domain of robot grasping. Compared with traditional discretization schemes, our model excels both in task classification and prediction of hand grasp configurations. Further, being a fully probabilistic model it handles uncertainty in the data and can easily be integrated into other frameworks in a principled manner.**

I. Introduction

In the research field of robot grasping and manipulation, an important challenge is to automatically plan a grasp on an object that affords an assigned manipulation task. This requires a robot to integrate a large range of sensory streams in order to estimate the state of the scene, understand the task requirements, and reason about its ability to plan and control in an uncertain, open-ended environment. In recent years, learning by imitation has been one important approach to these problems [1], [2], [3], [4]. The goal is to design a system whereby a robot learns a task by imitating a human teacher. In order to observe a human demonstration, the robot needs to obtain the state of the entire scene, not only the objects, but also the human actions which often are characterized using many variables (e.g. hand kinematics). In addition, the robot also needs to recognize the intention of the human to be able to reproduce the grasp in a goal-directed way. This poses a significant modelling challenge.

To correctly describe a manipulation task, both conceptual high-level information and continuous low-level sensorimotor variables are needed. In previous work [5], we have applied Bayesian Networks (BN) [6] to model the grasping scenario. The aim is to exploit the strengh of BNs in modeling complex joint distribution of these variables, so that the robot is able to reason at both high-level task representations and low-level planning and control using its sensorimotor systems. Though the results are promising, the approach suffers from inheritant disadvantages of BN, being able to use only a small sub-set of the available sensory streams.

These difficulties arise from several sources. When many nodes are included, manually encoding the structure of the network is non-trivial and poses significant challenges. Therefore, we are motivated to learn the BN structure automatically from the data. A large body of work exists (see review in [7]) learns the structure of BN in scenarios where all the variables are discrete. However, our observations are represented using both discrete and multivariate continuous variables. In such scenarios, no directly applicable method for learning the structure exists. To learn structure from continuous data, several approaches have sought to discretize continuous data as a prestep (see reviews in [8]). However, as data-dimensions increases discretisation becomes significantly more challenging. Recently, a multivariate discretization scheme was developed in [9]. However, the method suffers from ineffective clustering in high-dimensional spaces and slow convergence.

The contribution in this paper is to introduce a novel multi-variate discretization approach, which makes use of high correlation of the high dimensional data, combining a non-parametric dimensionality reduction approach (Gaussian process latent variable model (GP-LVM) [10]), with a Gaussian mixture model. In addition, the GP-LVM being a generative model, when combined with BNs, can improve the data reconstruction in the high-dimensional *continuous* space. We evaluate our proposed discretization approach by examining its effect on learning BNs. When compared with other alternative approaches, the GP-LVM-based method significantly improves the task classification accuracy, and intuitive data reconstruction in high-dimensional space.

II. Related Work

Graphical models such as Bayesian networks [6] aim at exploiting conditional independencies in the data in order to factorize the joint distribution of the data. For interpretable scenarios such as object manipulation, the factorization encodes the semantic relationships between groups of variables that represents the entire scene of manipulation tasks. Further, being a generative model, a BN can be directly applied for reasoning and planning of new grasps outside the training data. In [3] a discrete BN is applied to model affordances of objects while a robot is interacting with them in simple manipulation tasks to explore its sensorimotor capabilities. In our recent work [5], we extended the domain to complex,

D. Song, C.H Ek, K. Huebner and D. Kragic are with KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, www: `http://www.csc.kth.se/cvap`, e-mail addresses: {`dsong,chek,khubner,danik`}`@csc.kth.se`.

task-oriented grasp planning problems, and tackled challenges in both object/task recognitions and grasp selection under uncertain environments. The model allows not only to reason about high-level task representations, but also to make detailed decisions on which object to use and which grasp to apply in order to fulfill the requirements of an assigned task. However, the BN models both discrete and continuous variables, which presents the limitations when a large number of variables are to be modeled, and when the structure of the network is to be learned.

Learning BN structure from both continuous and discrete data is difficult, particularly when continuous data is high-dimensional and has complex distributions. Most algorithms for structure learning only work with discrete variables. Therefore, a common approach is to convert the mixed modeling scenario into a completely discrete one by discretizing the continuous variables [8]. In data discretization, most techniques are based on either heuristic evaluation [3], or equal-frequency and equal-width binning methods [7], [8], [11]. In latter, the data is divided into a set of bins, and then the number and the location of the bins are optimized based on some information criterion such as Akaike's criterion [7], [12]. However, the method is only applicable for continuous variables with only one dimension. Multivariate discretization has been explored for association rule discovery in the field of data mining. A recent work in [9] introduced a novel approach that is based on density-based clustering techniques. The method assumes that high density data clusters often occur when there are high associations between the continuous variables. Once these clusters were identified, a genetic algorithm was applied to optimize the cut points for the discretization intervals.

The idea behind the method in [9] is similar to ours in that we also use the clusters discovered in the data as the class information used for discretization. However, [9] explored the clustering in the original observation space of the data which will suffer when the variable is very high-dimensional and thus density learning becomes inefficient. In robotic applications, many variables of interest such as hand joint kinematics are high-dimensional. However, the intrinsic dimensionality or the number of degrees of freedom are often much smaller due to significant between-dimension correlation in the observed space. In dimensionality reduction the aim is to exploit such correlation in order to find a more compact and efficient representation.

The field of dimensionality reduction has received significant attention over the last decade. Originating from simple linear algorithms such as Multi-Dimensional-Scaling [13] and Principle Component Analysis, several non-linear extensions were suggested [14], [15]. However, these methods often rely on assumptions that significantly limits their application. Therefore more general, generative interpretations such as Probabilistic PCA (PPCA) [16] were proposed. Of particular success has been the GP-LVM being a non-linear generalization of PPCA based on non-parametric regression using Gaussian Processes (GP). In next two sections, we will first introduce GPs and the GP-LVM. Based on this background, we will proceed to explain our proposed discretization model.

III. Gaussian Processes

A GP [17] is defined as a collection of random variables, any finite number of which follows a joint Gaussian distribution. GPs are commonly used to model distributions over functional spaces making probabilistic treatment of such possible.

Given a set of data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$, $\mathbf{x}_i \in \Re^q$ and $\mathbf{Y} = [y_1, \ldots, y_N]^{\mathrm{T}}$, $y_i \in \Re$, we assume that y_i are related to $\mathbf{x}_i$ through an underlying functional relationship f where the observations $\mathbf{Y}$ have been corrupted by additive Gaussian noise,

$$y_i = f(\mathbf{x}_i) + \epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We assume that the unknown function values f can be modeled using a GP,

$$p(f(\mathbf{X})|\mathbf{X}) = \mathcal{N}\big(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X})\big), \tag{2}$$

where $\mu(\cdot)$ is the mean and $k(\cdot, \cdot)$ the covariance function respectively. Without loss of generality we can remove the bias from the data and set the mean funtion to be constant zero. The covariance function defines what types of functions are more prominent in the prior, and are specified using a set of parameters $\mathbf{\Phi}$ which will be referred to as the hyper-parameters of the GP. The hyper-parameters are difficult to know a-priori, so we want to learn them from data. Combining the prior with the likelihood, which is Gaussian due to the noise assumption, and marginalizing,

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{\Phi}) = \int p(\mathbf{Y}|f(\mathbf{X}))p(f(\mathbf{X})|\mathbf{X}, \mathbf{\Phi})df(\mathbf{X}), \tag{3}$$

leads to the marginal likelihood of the observed data. We can then find the hyper-parameter $\mathbf{\Phi}$ that maximizes Eq.3.

Through the definition of a GP, the joint distribution of $\mathbf{X}$ and a set of new input locations $\mathbf{X}^*$ can be formulated. Marginalization of the observed data $\mathbf{X}$ leads to the predictive distribution,

$$p(f(\mathbf{X}^*)|\mathbf{Y}, \mathbf{X}, \mathbf{\Phi}) = \mathcal{N}\big(k(\mathbf{X}^*, \mathbf{X})\left(k(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}\right)^{-1}\mathbf{Y},$$
$$k(\mathbf{X}^*, \mathbf{X}^*) - k(\mathbf{X}^*, \mathbf{X})\left(k(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}\right)^{-1}k(\mathbf{X}, \mathbf{X}^*)\big), \tag{4}$$

from which inference can be performed.

As can be seen above, learning in the GP framework requires invertion of a $\mathrm{N} \times \mathrm{N}$ matrix, an operation of cubic complexity. This places significant constraints on the size of the datasets for which the framework can be applied. This has led to a significant amount of work on methods which aim to reduce the computational cost associated with the model. In specific, a set of methods which introduce an additional set of input variables $\mathbf{U}$ referred to as the *inducing* variables have been proposed. By assuming the observed data to be independent given the inducing points, the prior can be written as $p(f(\mathbf{X})) = \int p(f(\mathbf{X})|\mathbf{U})p(\mathbf{U})d\mathbf{U}$. When substituted into the derivation of the predictive distribution, the inducing conditional takes the following form,

$$p(f(\mathbf{X})|\mathbf{U}) = \mathcal{N}\big(k(\mathbf{X}, \mathbf{U})k(\mathbf{U}, \mathbf{U})^{-1}\mathbf{U},$$
$$k(\mathbf{X}, \mathbf{X}) - k(\mathbf{X}, \mathbf{U})k(\mathbf{U}, \mathbf{U})^{-1}k(\mathbf{U}, \mathbf{X})\big). \tag{5}$$

Note that the matrix inversion in Eq. 5 is only applied to the covariance matrix on the inducing points.

By using a much smaller number of inducing points compared to the original data, several different approximations have been suggested. Explaining the different approximation in detail is beyond the scope of this paper. However, we briefly note that one kind of approximation method adds the inducing points as parameters of the covariance function and estimates their location jointly with the remaining hyper-parameters. Intuitive as this might be, it significantly increases the number of parameters to be estimated which makes such an approach prone to over-fitting. A recent work by [18] suggested a variational framework to circumvent this problem. The method treats the inducing points as parameters, and proceeds by minimizing the Kullback-Leibler divergence between the exact posterior distribution and the variational distribution. In this paper we will make use of this variational approximation in [18].

The GP-LVM [10] is a generative model for dimensionality reduction based on GPs. We assume a set of observed data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}}$, with $\mathbf{y}_i \in \Re^D$, is generated from a low-dimensional variable $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, with $\mathbf{x}_i \in \Re^q$, through a mapping f corrupted by additive Gaussian noise, $\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Each dimension of the output is modeled as an independent GP,

$$p(f(\mathbf{X})|\mathbf{X}) = \prod_i^D p(f(\mathbf{X})^{\mathrm{T}}\mathbf{e}_i|\mathbf{X})) \tag{6}$$

where $\mathbf{e}_i$ is standard basis vector. The GP-LVM proceeds in analog with the regression framework with one significant difference. In the GP-LVM the input variables $\mathbf{X}$ are treated as random variables, and the solution to $\mathbf{X}$ is found together with the hyper-parameters through maximum likelihood,

$$\{\hat{\mathbf{\Phi}}, \hat{\mathbf{X}}\} = \mathrm{argmax}_{\{\mathbf{\Phi},\mathbf{X}\}} p(\mathbf{Y}|\mathbf{X}, \mathbf{\Phi}). \tag{7}$$

This might seem non-sensical as clearly the combination of mappings f and input locations $\mathbf{X}$ that could have generated the observed data $\mathbf{Y}$ is infinite. However, by fixing the dimensionality of the latent space q to be significantly smaller compared to the observed data, and by the regularizing effect of the GP-prior, a solution can be found efficiently.

IV. GP-LVM Discretization

In this section, we explain a novel descretization approach, which, by exploiting recent advances in sparse GPs and the GP-LVM, is able to compactly represent high-dimensional continuous data using a low-dimensional discrete mixture model in a principled way.

Our approach is a straight-forward two-stage framework, as exemplified in Fig. 1. Given a set of observed high-dimensional data $\mathbf{Y}$, in this example the 20D feature $fcon$, we wish to find a compact discrete representation of the data which can be efficiently used in a BN. In the first stage we use a GP-LVM model where the generative mapping is formulated as a sparse GP from a small sub-set of inducing points. Using the sparse variational approximation of [18], the location of the inducing points (the red star markers in Fig. 1 a) and the hyper-parameters of the generative mapping can be found. Further, in order to avoid setting the dimensionality by hand and reduce reliance on initialization, we include the rank-regularization framework of [19].

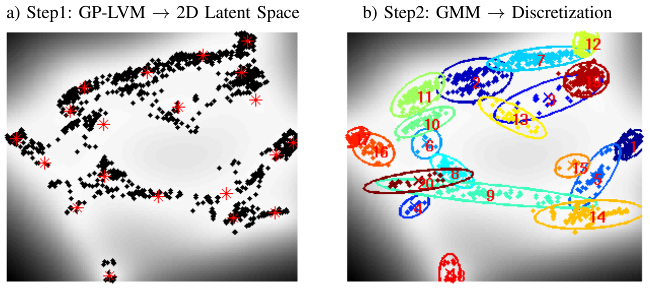


Fig. 1. GP-LVM-based discretization for $fcon$: a) 2D latent space learned using sparce GP-LVM, b) discretization using GMM in the 2D latent space.

In the second stage, one could directly use the inducing points of the GP as cluster-centers to discretize the data. However, when training GP-LVM, we use a spherical covariance function in order to reduce the number of parameters that need to be estimated from data. But this might lead to a less compact representation since variances in the dimensions with high correlation will be modeled as the same as those with low correlations. Our solution is, in the second stage, to learn a more compact representation using a mixture model that allows full covarances therefore more flexible component representations. More specifically, we learn a Gaussian mixture model (GMM) explaining the latent locations $\mathbf{X}$. The inducing points are used as the initial centers of all mixture components, and the model is optimized through the standard Expectaion Maximization approach. Fig. 1 b) shows the resulted mixture model, with the ellipsoids representing one standard deviation of the Gaussian components, and the colored data points being the resulted discretization.

For runtime evaluation, an unoptimized Matlab implementation with 1800 data points, it takes about 100 seconds to train the discretization model for the worst case data.

V. Data Generation

In this section, we will briefly introduce how the variables, i.e. the feature values, are extracted. Tab. I shows the features used in this work. The features describing each grasp are divided into three sub-sets: *object features* (O) extracted from the object representation, *action features* (A) extracted from the planned grasps, and *constraint features* (C) resulting from the complementation of both, i.e. the hand-object configuration. Each grasp is visualized in GraspIt! to a human tutor who associates it with a task label (T).

Fig. 2 shows the schematic of the data generation process. To extract those features, we first generate grasp hypotheses using the grasp-planner BADGr [20], and evaluate them as scenes of object-grasp configurations in a grasp simulator, GraspIt! [21]. BADGr includes extraction and labeling modules to provide the set of variables presented in Tab. I. The interested reader is referred to [5], [20] for more details on the feature extraction. We emphasize that the grasp representation does not have to be non-redundant, e.g. *cvex* and *shcv* are allowed variables to both represent object shapes. Such an "over-representation" of the featured

TABLE I

USED FEATURES WITH THEIR DIMENSIONALITY D (FOR CONTINUOUS) AND NUMBER OF STATES N AFTER DISCRETIZATION.

	Name	D	N	Description
T	$task$	-	3	Task Identifier
O_1	*obcl*	-	6	Object Class
O_2	$size$	3	8	Object Dimensions
O_3	$cvex$	1	4	Convexity Value $[0, 1]$
O_4	$shcv$	3	7	Shape Class Vector (Zernike Similarity)
A_1	$fcon$	20	20	Final Hand Configuration
A_2	dir	4	20	Approach Direction (Quaternion)
A_3	pos	3	14	Grasp Position
A_4	$egpc$	2	6	Eigengrasp Pre-Configuration
A_5	$upos$	3	11	Unified Spherical Grasp Position
C_1	$fvol$	1	6	Free Volume
C_2	$gbvl$	1	4	Volume of Grasped Boxes
C_3	$pshcv$	3	7	Part Shape Class Vector (Zernike Similarity)
C_4	$pecce$	1	3	Part Eccentricity $[0, 1]$
C_5	$g1bx$	1	2	Grasped-1-Box Value $[0, 1]$
C_6	$qeps$	1	5	Grasp Stability Measure (eps)
C_7	$qvol$	1	3	Grasp Stability Measure (vol)

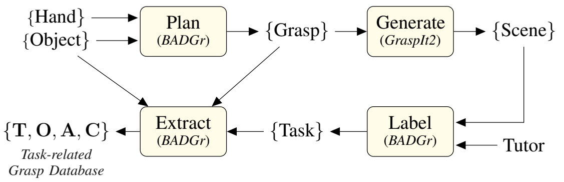


Fig. 2. Schematic diagram for generating task-related grasp database.

variables allows us to use BNs to identify the importance of, and dependencies between these variables in the scenarios of robot grasping and manipulation.

VI. RESULTS

In this section, we will describe four experiments to evaluate our framework. In order to compare the suggested approach with other methods we also present results obtained by discretizing using a mixture model in either the observed data space (NoReduce), or in the space spanned by the dominant principle components (PCA). To compare them under the same level of model complexity, the dimensionality of the mixture model and the number of principle components are set to be the same as for those learned by the GP-LVM.

To generate data, the grasp hypotheses are produced on 24 object models evenly covering six object classes (*obcl*) including knifes, hammers, screwdrivers, glasses, bottles and mugs. The grasps are labeled according to three manipulation tasks: *hand-over, pouring* and *tool-use*. The total data-set consists of 1800 data points uniformly divided over the three tasks. We use 80% of the data for training and the remainng 360 instances for testing.

In all the experiments, when feature set O is observed, all object features except the object class *obcl* are observed. We assume that *obcl* is unknown in order to simulate the real-world scenarios where recognizing object categories from its raw features is still a hard problem for robot sensor systems.

A. Experiment I: Structure Learning

The first experiment is to learn structure of BNs from the discretized data by using the three different discretization schemes. We use a greedy search algorithm to find the structure, or the directed acyclic graph (DAG), in a neigborhood of graphs that maximizes the network score (Bayesian information criterion [22]). The search is local and in the space of DAGs, so the effectiveness of the algorithm relies on the initial DAG. As suggested by [7], we use another simpler algorithm, the maximum weight spanning tree [23], to find an oriented tree structure as the initial DAG. We assume the task class variable is the 'cause' of the systems thus the root node of the network.

Fig. 3 shows the results of learned DAGs from the three models. We note that learning the structure from data reveals complicated relationships among these large pool of variables, which will otherwise be very difficult to encode by human experts. From an initial inspection, the DAGs associated with the different discretization schemes share much of their structure. And they conform to our intuitive knowledge of the dependency relations between the variables. For example, the three action features – pos, $upos$ and dir – are fully connected because the unified spherical grasp position $upos$ is directly derived from the grasp position pos and the hand orientation with respect to the object dir. We also see that $task$ has a direct connection to *obcl* which in turn directly determines the other object features. This make perfect sense as among the six object classes, 3 classes are tools therefore afford the *tool-use* task, and the other 3 classes are containers therefore afford the *pouring* task.

When comparing the DAG of GP-LVM to those of PCA and NoReduce models, the main difference lies in the connections involving $fcon$. This is not surprising. As $fcon$ represents the final grasp configuration of the hand which is high-dimensional and contains a lot of information, it is therefore expected to be most affected by the discretization methods. As we can see, $fcon$ in the GP-LVM model has more parents thus more family members than that of the other two models. This indicates that the GP-LVM-based discretization of $fcon$ captured a different representation in the continuous, high-dimensional space of the variable, therefore, the dependencies of $fcon$ with other variables are different. In specific, in the GP-LVM model, $fcon$ is connected to $pecce$, the eccentricity of the grasped part, which clearly has a big impact on the final grasp configuration. This dependency is discovered when using GP-LVM-based discretization scheme, but is lost by both PCA and NoReduce models.

B. Experiment II: Task Classification

As we do not have the true structure of the BN to evaluate the learned DAGs, in this section we evaluate the effectiveness of the structure learning using the three different discretization schemes by comparing their task classification performance. As shown in Fig. 4, this task classification is based on the inference of $task$ variable given observation of different set of other variables that form a complete permutation of the 3 feature sub-sets: O, A and C.

There are three major observations from the result of this

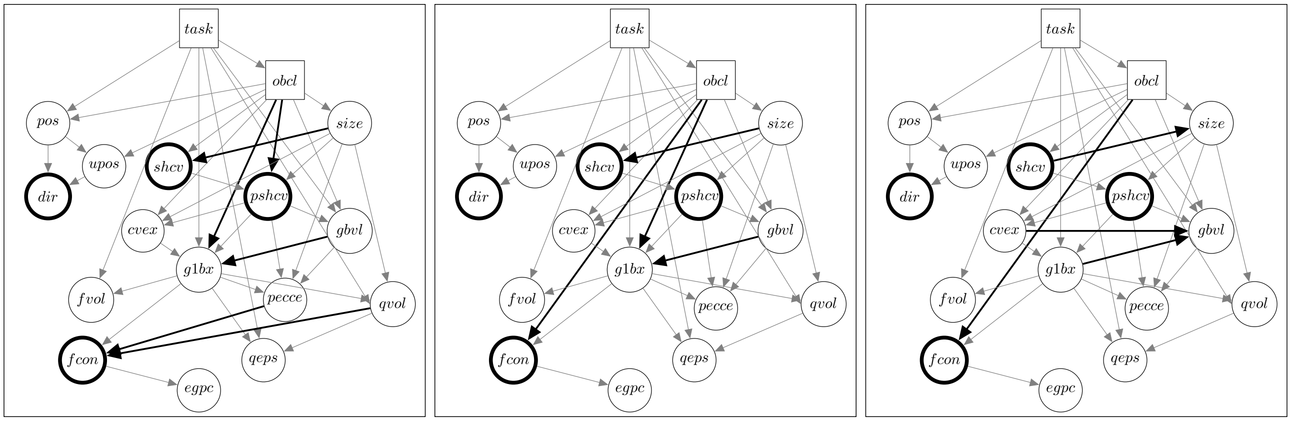


Fig. 3. Experiment I: The three resulting DAGs by applying structural learning on (left) GP-LVM, (middle) PCA and (right) NoReduce discretization schemes. The differences in DAGs are highlighted by thick arrows. Square nodes represent discrete variables and circled nodes continuous. Continuous nodes with thick boundaries identify high-dimensional variables onto which different discretization schemes have been applied.

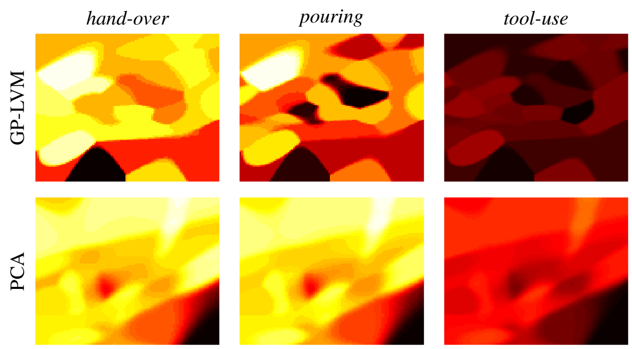
	O	A	C	O, A	O, C	A, C	O, A, C
GP-LVM	0.70 0.12 0.17	0.58 0.10 0.32	0.75 0.10 0.15	0.74 0.09 0.17	0.73 0.04 0.23	0.78 0.12 0.10	0.86 0.04 0.10
	0.12 0.88 0.00	0.03 0.38 0.59	0.33 0.57 0.10	0.05 0.95 0.00	0.29 0.71 0.00	0.16 0.82 0.02	0.15 0.85 0.00
	0.16 0.00 0.84	0.01 0.05 0.94	0.38 0.00 0.62	0.03 0.00 0.97	0.11 0.00 0.89	0.03 0.03 0.94	0.04 0.01 0.95
PCA	0.71 0.13 0.16	0.58 0.13 0.29	0.73 0.06 0.21	0.77 0.12 0.11	0.78 0.04 0.18	0.79 0.08 0.13	0.87 0.06 0.07
	0.11 0.89 0.00	0.02 0.59 0.39	0.47 0.46 0.07	0.03 0.97 0.00	0.41 0.57 0.02	0.32 0.65 0.03	0.26 0.73 0.01
	0.40 0.00 0.60	0.02 0.10 0.88	0.62 0.00 0.38	0.14 0.00 0.86	0.36 0.00 0.64	0.18 0.00 0.82	0.13 0.00 0.87
NoReduce	0.58 0.28 0.14	0.69 0.19 0.12	0.58 0.07 0.35	0.68 0.21 0.11	0.72 0.11 0.17	0.77 0.12 0.11	0.79 0.11 0.10
	0.28 0.44 0.28	0.12 0.53 0.35	0.14 0.17 0.69	0.11 0.50 0.39	0.55 0.22 0.23	0.16 0.44 0.40	0.32 0.51 0.17
	0.09 0.19 0.72	0.17 0.36 0.47	0.46 0.01 0.53	0.06 0.20 0.74	0.37 0.02 0.61	0.45 0.17 0.38	0.29 0.14 0.57

Fig. 4. Experiment II: Confusion matrices for task classification given different observations spaces: permutations of O, A, C features. For each 3×3 matrix, from left to right (top to down), the 3 tasks are: *hand-over, pouring, tool-use*.

experiment: *i)* Object features contain a lot information that is task-relavant. By just observing O using GP-LVM model, we see quite good classification rate. But A and C features by themselves are close to random. When more information is observed, the classification rate improves. Particularly, for GP-LVM and PCA models, O and A compliments each other and improve the classification rate significantly for the *pouring* and *tool-use* tasks. When C is also added in the observation space (O, A, C), the classification for the *hand-over* task is improved in both models. However, we notice that the accuracy on classifying *pouring* task decreases from 95% to 85%. This may be explained by the imbalanced training data in many of the C variables. For example, when there is lack of data points for some states of the C variables when the task is *pouring*, the conditional probabilities learned in the BNs will be under-determined. In such situations, clamping (observing) these C variables at these states will certainly hurt the inference on the posterior distribution on task. *ii)* When comparing the three models, NoReduce, which discretizes the variables in original high-dimensional space, has close-to-random classification rate. The two models with dimensionality reduction (GP-LVM, PCA) greatly outperform the NoReduce model. GP-LVM is clearly the best in most of the observation conditions. *iii)* When only A features are observed, we see quite high classification rate only for *tool-use* (PCA: 88%, GP-LVM: 94%), while the other two tasks are random. This can be explained by that since the tools are mostly very slim compared to other objects, thus the fingers are very close to each other when grasping them. As a result some A features such as $fcon$ contain much information to separate *tool-use* from other tasks.

C. Experiment III: Prediction of Grasp Final Configuration

Since the Bayesian network is a generative model, in addition to evaluate its performance in discrimitive power (i.e. task classification) in the first experiment, we would also like to see how well the model can sucessfully reconstruct any variable given an assigned task. Particularly, we want



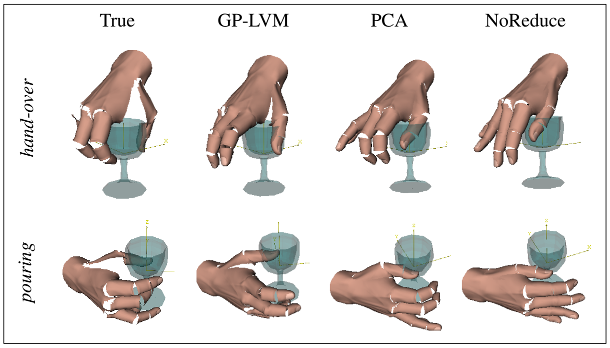


Fig. 5. Experiment III: Top panel shows the likelihood maps of $fcon$ in latent space given tasks and glass's object features $P(fcon|T,O)$; Bottom panel shows prediction of $fcon$ in original space, visualized in GraspIt!.

to show the strength of the proposed discretization scheme GP-LVM on reconstructing the high-dimensional variables. Due to the space limit of the paper, we choose the grasp final configuration $fcon$ as our target variable. But note that the similer effects have been observed on other variables. In this experiment, the goal is to demonstrate that using the discretization scheme GP-LVM, as compared to PCA and NoReduce, we can *i)* better model the constraining effect of the manipulation tasks on $fcon$, and *ii)* predict more intuitive $fcon$ in the original hand configuration space.

To do so, we first obtain the likelihood maps of $fcon$ in the 2D latent space for the GP-LVM and PCA models (see top panel of Fig. 5). The light color of the map indicates that the point has high likelihood for the task. The maps are generated by evenly sampling the posterior distribution $P(\cdot|T,O)$ under the BN model for each task using the same object (glass). This object comes from the test data in order to investigate performance on generalization outside the training data.

The main observations here include: *i)* for both GP-LVM and PCA models, the *tool-use* task has much darker maps, indicating that the glass is clearly not tool-usable; *ii)* when comparing the maps between the two models, we see there are clear differences between *hand-over* and *pouring* tasks for GP-LVM, whereas for PCA the likelihood patterns are almost same. This implies that the GP-LVM model has captured the potential constraining effects of the tasks on the final configuration of the hand. On the contrary, the PCA model is almost 'blind' on the potential task constraints.

As the 2D latent space representation can not give us any intuition about how good is the $fcon$ predicted by the model, we would like to project the prediction into the original 20D space and visualize it in the simulator. This way, we can also visualize the results from the NoReduce model. The bottom panel in Fig. 5 shows the results with the same object, for the *hand-over* and *pouring* tasks. From the left to right columns, the images represent the ground truth $fcon$, predicted $fcon$ from GP-LVM, PCA and NoReduce models respectively. We can see that the reconstruction of the GP-LVM model is closer to true $fcon$, and importantly the hand configurations are more natural compared to other models. Both the PCA and NoReduce models have unintuitive, even 'impossible' hand configurations that are far away from the true data.

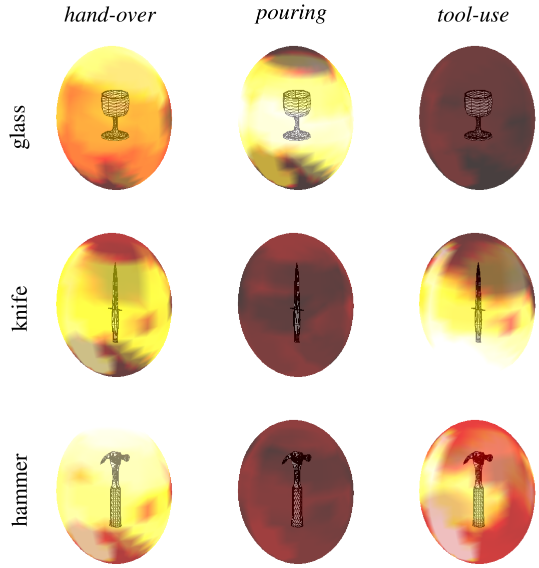


Fig. 6. Experiment IV: Likelihood of Grasp Position Given Tasks and Object Features $P(pos|T,O)$, resulted from the GP-LVM model.

The reason lies on the foundamental differences in the discretization schemes. The GP-LVM model provides a generative discretization scheme, while PCA and NoReduce models do not. As a result, GP-LVM-based BNs can model $P(fcon|T,O)$, where $fcon$ represents the original 20D continuous data, whereas the other models can not. In other words, the proposed GP-LVM-based discretization scheme allows us to construct a *full generative framework* that includes BN, GP-LVM and GMM. This framework is very powerful to model the constraints for robot grasping and manipulation tasks.

D. Experiment IV: Inference on Grasp Position

From the first three experiments, we have shown that the GP-LVM-based discretization scheme outperforms others in both task classification and data reconstruction. The goal of the last experiment is to confirm that GP-LVM model can successfully encode some task constraint on different objects and tasks. Notice that the constraint of a given task is often encoded by a combination of multiple features, e.g. one should not grasp this object from this position pos, in this orientation dir, and with this joint configuration $fcon$. However due to space limit and for the purpose of easier evaluation by the readers, we choose a single intuitive variable, the grasp position pos, to visualize the task constraint.

Similar to obtaining the likelihood map of $fcon$ in experiment III, we sample 625 grasping positions evenly on an ellipsoid around the object. The size of the ellipsoid is determined by the training data so that the ellipsoid envelops the outer surface of all the grasping positions. As shown in Fig. 6, for each sampled position, the likelihood is obtained given the 3 tasks, and the object features for 3 unknown objects: a glass, a knife and a hammer, i.e. $P(pos|T, O)$.

Fig. 6 shows that the model sucessfully rules out the glass for *tool-use*, and the knife and hammer for *pouring*. For *pouring*, the glass can not be grasped from the top as it will block the opening of the glass; similarly, when using the knife or hammer as a tool, the grasp should avoid the sharp blade or the head of the hammer as functional parts. All the 3 objects afford the *hand-over* task, and the likelihood maps of pos for *hand-over* also capture the intuitive constraints for these objects. Similar results are also observed in PCA model but not in NoReduce model. The reason is revealed by previous task classification results as shown in Fig. 4: while PCA model has good classification rate given the observation on O, A features, the NoReduce model is almost random.

VII. Conclusion

In conclusion, the sparse GP-LVM-based discretization method excels over other methods in learning and inference with Bayesian networks. The compact, efficient data representation allows fast structure learning for BNs that model a large number of variables. And the resulted BN performs significantly better in both task classification and data reconstruction. In addition, since GP-LVM is a generative technique for dimensionality reduction, the model encodes the likelihood of each point in the latent-space, which, when combined with the prediction from BNs, can reproduce much more accurate and intuitive high-dimensional variables such as hand grasp configurations. This presents a major advantage of our proposed discretization method in the field of robotic applications, where many sensory and motor signals are high-dimensional with complex distributions.

In this paper, we only used human hand model to present the discretization techniques for learning Bayesian networks. However, the framework can be generalized to any hands, from which the training data should be generated. The goal of the hand-specific Bayesian network is to allow the task information to be transferred between different hands or embodiments, therefore the goal-directed grasp imitation can be performed [5].

There are some limitations in the current approach that need further research. Firstly, the number of discrete states are manually chosen to satisfy a trade-off between refined data representation and complexity of BNs. In the future, we would also like to learn this hyper parameter automatically from data. Secondly, the inducing points create a sparse model of the full GP, however, there is nothing in the model that encourages the inducing points to be sparse themselves, i.e. less inducing points. Sparseness among the inducing points would reduce the amount of shared explanation, which we believe can lead to better clustering.

We would also like to test our discretization based task constraint model in the real robot platforms where sensorimotor uncertainty is more prominant. We believe this will further exemplify the benifits of using a probabilistic model capable of dealing with uncertainty in real-world applications.

ACKNOWLEDGMENTS

This work was supported by EU IST-FP7-IP GRASP, EU IST-FP6-IP-027657 PACO-PLUS, and Swedish Foundation for Strategic Research.

References

[1] S. Kang and K. Ikeuchi, “Toward Automatic Robot Instruction from Perception-Mapping Human Grasps to Manipulator Grasps,” *IEEE Transactions on Robotics and Automation*, 1997.

[2] C. L. Nehaniv and K. Dautenhahn, Eds., *Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press, 2004.

[3] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, “Learning Object Affordances: From Sensory–Motor Coordination to Imitation,” *IEEE Transactions on Robotics*, 2008.

[4] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, “Learning and Generalization of Motor Skills by Learning from Demonstration,” in *IEEE International Conference on Advanced Robotics*, 2009.

[5] D. Song, K. Huebner, V. Kyrki, and D. Kragic, “Learning Task Constraints for Robot Grasping using Graphical Models,” in *IROS*, 2010.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[7] P. Leray and O. Francois, “BNT Structure Learning Package: Documentation and Experiments,” Université de Rouen, Tech. Rep., 2006.

[8] L. D. Fu and I. Tsamardinos, “A Comparison of Bayesian Network Learning Algorithms from Continuous Data,” in *AMIA*, 2005.

[9] H. Wei, “A Novel Multivariate Discretization Method for Mining Association Rules,” in *APCIP*, 2009.

[10] N. D. Lawrence, “Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models,” *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, November 2005.

[11] I. Ebert-Uphoff, “A Probability-Based Approach to Soft Discretization for Bayesian Networks,” Georgia Institute of Technology. School of Mechanical Engineering, Tech. Rep., 2009.

[12] O. Colot, P. C. Olivier, and A. El Matouat, “Information Criteria and Abrupt Changes in Probability Laws,” in *Signal Processing VII: Theorie and Applications*, 1994, pp. 1855–1858.

[13] T. Cox and M. Cox, *Multidimensional Scaling*, 2001.

[14] J. Tenenbaum, V. Silva, and J. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[15] K. Weinberger, F. Sha, and L. Saul, “Learning a Kernel Matrix for Nonlinear Dimensionality Reduction,” *ICML*, 2004.

[16] M. Tipping and C. Bishop, “Probabilistic Principal Component Analysis,” *Journal of the Royal Statistical Society*, 1999.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[18] M. Titsias, “Variational Learning of Inducing Variables in Sparse Gaussian Processes,” in *Artificial Intelligence and Statistics*, 2009.

[19] A. Geiger, R. Urtasun, and T. Darrell, “Rank Priors for Continuous Non-Linear Dimensionality Reduction,” in *CVPR*, 2009.

[20] K. Huebner, “BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems: Workshop on Grasp Planning and Task Learning by Imitation*, 2010, **URL:**http://www.csc.kth.se/∼khubner/badgr/.

[21] A. T. Miller and P. K. Allen, “GraspIt! A Versatile Simulator for Robotic Grasping,” *IEEE Robotics and Automation Magazine*, 2004.

[22] G. Schwarz, “Estimating the Dimension of a Model,” *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[23] C. Chow and C. Liu, “Approximating Discrete Probability Distributions with Dependence Trees,” *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.