Exploring Affordances in Robot Grasping through Latent Structure Representation

Carl Henrik Ek, Dan Song, Kai Huebner, and Danica Kragic[*]

KTH – Royal Institute of Technology, Stockholm, Sweden, Computer Vision & Active Perception Lab., Center for Autonomous Systems

Abstract. An important challenge in robotic research is learning by imitation. The goal in such is to create a system whereby a robot can learn to perform a specific task by imitating a human instructor. In order to do so, the robot needs to determine the state of the scene through its sensory system. There is a huge range of possible sensory streams that the robot can make use of to reason and interact with its environment. Due to computational and algorithmic limitations we are interested in limiting the number of sensory inputs. Further, streams can be complementary both in general, but more importantly for specific tasks. Thereby, an intelligent and constrained limitation of the number of sensory inputs is motivated. We are interested in exploiting such structure in order to do what will be referred to as *Goal-Directed-Perception* (GDP). The goal of GDP is, given partial knowledge about the scene, to direct the robot's modes of perception in order to maximally disambiguate the state space.

In this paper, we present the application of two different probabilistic models in modeling the largely redundant and complementary observation space for the task of object grasping. We evaluate and discuss the results of both approaches.

1 Introduction

A major challenge in the research field of robot grasping and manipulation is to automatically plan a grasp on an object that affords a specific manipulation task. To do so, robots need to obtain a large range of physical attributes of objects through their sensory system, reason about their own motor capabilities, as also understand how the task requirements constrain the specific features of objects and motor actions. In recent years, *learning by imitation* has been one important approach to these problems[1–4]. The goal is to design a system whereby a robot learns to perform a task by imitating a human teacher. More specifically, the robot learns the affordances [5] of an object and the requirements of a task through observing human tutors.

Though proved to be an effective approach, learning by imitation presents more challenges to the robot sensory system. To observe a human demonstration, robots need to obtain the state of the entire scene, not only the objects, but also the human actions. The problem becomes even harder when the goal is to perceive these features through vision systems [6]: human hands have many fingers that are often blocked by

[*] {chek,dsong,khubner,danik}@csc.kth.se, http://www.csc.kth.se/cvap. The code is available from http://www.cs.man.ac.uk/ neill/sgplvm/

the grasped object, and objects are also mostly occluded by the hands. Understanding the scene from this huge range of noisy and uncertain sensory data is a formidable challenge, both in terms of computational resources and real-time applications. On the side of the motor system, another challenge in imitation learning is how to map a human grasp to a robot grasp, particularly when their hands are very different. This is a common problem in imitation learning, referred as the *correspondence problem* [7]. To address this, a concept of *goal-directed imitation* is inspired from the imitation studies in developmental psychology: infants are able to infer the intention of others and understand and reproduce the task through their own actions [8]. In analogy to *goal-directed imitation*, we coin a complementary concept of *goal-directed perception* (GDP). The aim of GDP is, given a specific goal and partial knowledge of the scene, to direct the sensory system to gain information by maximum disambiguation in the state-space of the robot.

As an initial path towards the goal of GDP, we are interested in learning structures in the data which are relevant for classifying the task. That is, the system should reason about the structure of its several different continuous sensory streams, in close association with discrete and categorical variables for the task of grasping.

To reach this goal, we consider models capable of modeling the uncertainty in the data, $i.e.$ probabilistic models. The idea is to factorize the joint distribution of the observed data into some form of structure. Different models embed different assumptions about this structure, each associated with different advantages and disadvantages. Generally, there are two different approaches to create an accurate factorization of the joint distribution, by the structure of the nodes or by the relationship between specific nodes. In the first approach, the relationship between dependent nodes takes a simple form and the complexity in the data is handled by the structure of the dependencies. The opposite approach is to use a simple structure which allows the use of more complex models of the dependencies.

In our recent work [9], we applied the Bayesian network (BN) model [10], a probabilistic graphical model to learn the task constraints in robot grasping. This work addresses the problems in both sensory and motor systems in imitation learning: the probabilistic framework can deal with uncertainty in the sensory data, and the graphical structure of BN imposes a task-related representation of the object and grasp features. By training two embodiment-specific BNs for the human and the robot, we circumvent the correspondence problems in grasp mapping and realize goal-directed imitation. However, the BN is limited in its ability to model large range of sensory streams. When the number of nodes which model the sensory streams is high, the training becomes intractable.

These problems in BN-based task constraint learning motivate a framework that can reason about the underlying structure of the sensory input, and select a reduced set that efficiently disambiguates the tasks. In Fig. 1 a schematic figure of our over-all framework is shown.

1.1 Notation

Upper-case letters identify set variables, where Y specifies the full set of all observed variables $Y = \{O, A, C\}$. Super-script ^{S} is used to refer to a sub-set of the variables,

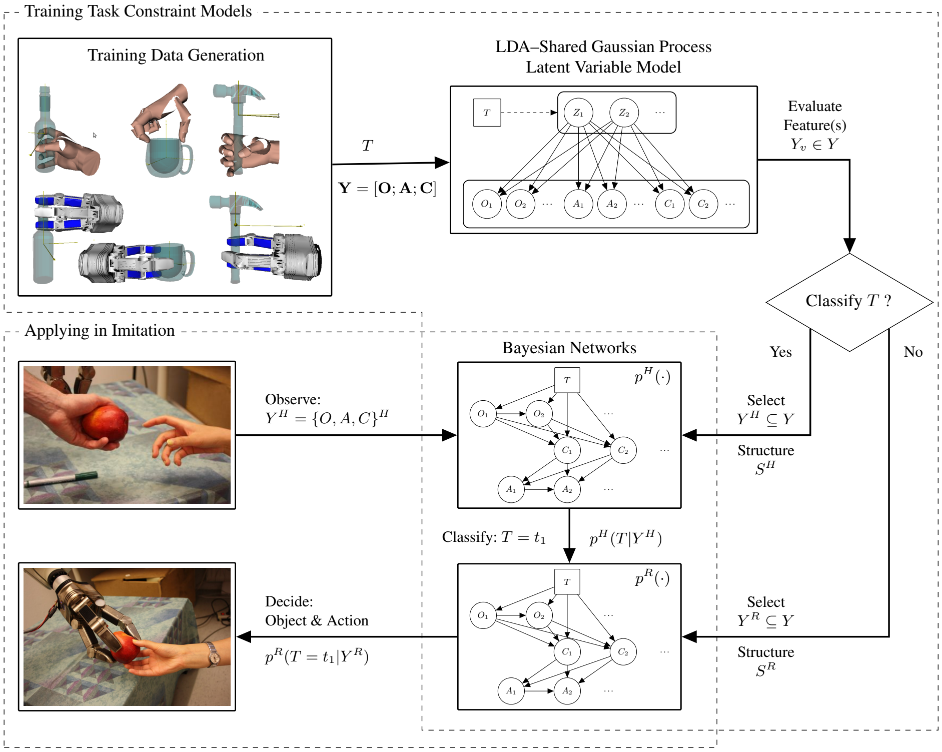


Fig. 1. System Diagram.

$Y^S \subseteq Y$. The letter M is the cardinality of the feature set, *i.e.* $M = |Y|$ and $M_S = |Y^S|$. We use letter v to refer to individual variables within a set or a sub-set of the variables. For example $Y_v \in Y$ can be $size$, one of the object features. We use bold letters to imply instantiations of a specific variable, where upper-case refers to "all" (or N) instantiations and lower-case refers to a specific instantiation identified by the subscript index i. For example, $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T$ represent N cases of instantiations of all the variables in the set Y. Then, all the N instantiations of variable Y_v will be $\mathbf{Y}^v$, and the i^{th} specific instantiation of Y_v will be $\mathbf{y}_i^v$. The i^{th} specific instantiation of all the variables in Y will be a concatenation of all the variables $\mathbf{y}_i = [\mathbf{y}_i^1; \ldots; \mathbf{y}_i^M]$. Last, we introduce D to be the dimensionality of an instantiation.

1.2 Training Data Generation

In this subsection, we will briefly introduce how the variables, *i.e.* real feature values, are practically extracted in our system. As it is out of the focus of the paper, we will not describe our grasp planner and the implementation of feature extraction, but refer the reader to [9]. In the notation and Fig. 1 we distinguish between three different types of observed variables defining the training data: *object features* (O) are directly extracted from the object representation, *action features* (A) are directly extracted from the grasp planner, and *constraint features* (C) emerge from the complementation of both, *e.g.* from the resulting contacts. We note that we shortcut the problem of using real world perception for all the features by using a simulation-based architecture and grasp planner. For details on those modules as also on the tutor-based task labeling process, in which the task variable T is provided, see [9].

For each good grasp i provided by the grasp planner, one training dataset $\mathbf{y_i}$ is generated. While in [9], we used a small network with M=7 features (exemplified in Fig. 2, right) to analyze Bayesian Network learning, we here use M=21 features (sketched in Fig. 5, right). This increase of features is not only accompanied by a drastic increase in the dimensionality of the whole feature vector (from D=15 to D=293), but also with strong redundancy in the data. For instance, size (*size*), eccentricity (*ecce*) and shape (*zern*) keep redundant information. These two characteristics (high dimensionality and redundancy) allow us to evaluate our models in terms of dimensionality reduction, dependency detection, and structure learning.

2 Models

2.1 Gaussian Process Latent Variable Model

The Gaussian Process Latent Variable Model (GP-LVM) [11] is a probabilistic model for dimensionality reduction. The observed data $\mathbf{Y}$ is assumed to have been generated through a mapping f from a low-dimensional latent variable $\mathbf{Z}$ corrupted by additive Gaussian noise, $y_i = f(\mathbf{z}_i) + \epsilon_i$. By placing a Gaussian Process (GP) prior, with parameters ϕ, over the generative mapping f leads to the marginal likelihood,

$$p(\mathbf{Y}|\mathbf{Z},\phi) = \int p(\mathbf{Y}|f)p(\mathbf{f}|\mathbf{Z},\phi)df. \tag{1}$$

The latent locations and the hyper-parameters can be found by maximizing the marginal-likelihood, in order to remove additional degrees of freedom we incorporate non-informative priors in order to proceed. This leads to the following objective,

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \sum_i \ln \phi_i + \sum_i \frac{1}{2}||\mathbf{z}_i||^2, \tag{2}$$

where $\mathcal{L}_{\text{data}} = p(\mathbf{Y}|\mathbf{Z},\phi)$.

One advantage of the GP-LVM framework is that it is straight-forward to include additional constraints and priors on the latent representation to replace the uninformative prior used in the original formulation Eq. 2. This has been exploited in order to learn representation with a specific topology [12] and to learn spaces respecting class constraints [13].

An extension to the GP-LVM framework for modeling two correlated observation spaces was presented in [14] referred to as the SGP-LVM. The SGP-LVM learns a single latent representation from which the two observed data spaces are generated by separate GP's. In this paper, we extend the original SGP-LVM to model the 21

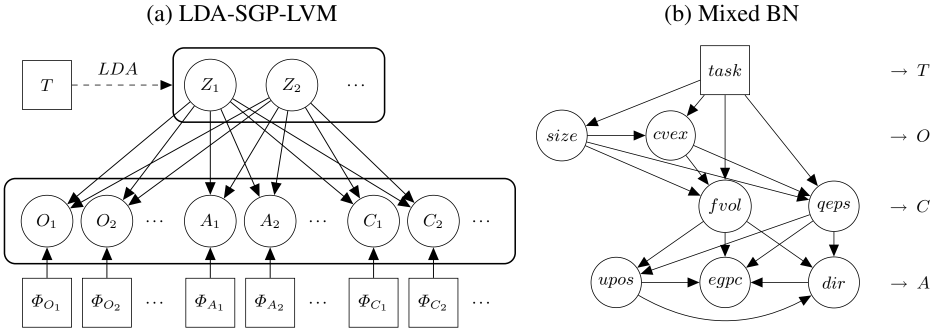


Fig. 2. *Left:* Graphical model of the LDA-SGP-LVM model where each observation stream $[\mathbf{o}_i; \mathbf{a}_i; \mathbf{c}_i]$ is generated from a shared latent representation Z through a mapping specified by a Gaussian Process with individual hyper-parameters ϕ_i. *Right:* Graphical model of a Bayesian Network used in [9]. The two models exemplify two different modeling strategies and factorizations of the joint distribution of the observed data. The LDA-SGP-LVM model has a simple structure assuming each observed feature to be independent given the latent variable but uses a strong and flexible Gaussian Process to describe this relationship. In comparison the BN applies a simple model of the relationship between the nodes but is capable to use a much more complex structure to model the data.

different observation spaces $Y = \{O, A, C\}$. This means that we will assume each of the observed data spaces to be independent given the latent representation giving the following factorization,

$$p(\mathbf{Y}|\mathbf{Z}, \mathbf{\Phi}) = \prod_v^{M_O} p(\mathbf{O}^v|\mathbf{Z}, \phi_{O_v}) \cdot \prod_v^{M_A} p(\mathbf{A}^v|\mathbf{Z}, \phi_{A_v}) \cdot \prod_v^{M_C} p(\mathbf{C}^v|\mathbf{Z}, \phi_{C_v}). \quad (3)$$

We will refer to $\mathcal{L}_{\text{data}}^{\text{SGP-LVM}} = \ln p(\mathbf{Y}|\mathbf{Z}, \mathbf{\Phi})$ as the data-term of the SGP-LVM.

The motivation behind this work is to evaluate the knowledge of the scene that the robot can robustly extract to be relevant for task classification. To this end we want to learn a shared latent representation of all the different observation streams that respects the class information in the data. This can be achieved by incorporating the class based prior over the latent space as described in [13] which leads to the following modified objective function,

$$\begin{aligned} \mathcal{L}^{\text{LDA-SGP-LVM}} &= \mathcal{L}_{\text{data}}^{\text{SGP-LVM}} + \underbrace{\sum_v^{M_O} \ln \phi_{O_v} + \sum_v^{M_A} \ln \phi_{A_v} + \sum_v^{M_C} \ln \phi_{C_v}}_{\mathcal{L}_{\text{hyper-prior}}} + \lambda \text{tr}(\mathbf{S}_w^{-1}\mathbf{S}_b) \\ &= \mathcal{L}_{\text{data}}^{\text{SGP-LVM}} + \mathcal{L}_{\text{hyper-prior}} + \lambda \mathcal{L}_{\text{LDA}}, \end{aligned} \quad (4)$$

where $\mathbf{S}_w$ and $\mathbf{S}_b$ are the within and between class scatter matrices of the latent representation respectively. The scalar λ controls the trade-off between reconstruction and class separation.

The term $\mathcal{L}_{\text{LDA}}$ will encourage a latent representation with large class separation and low within class variance. As noted in [13], the objective function Eq. (4) can be interpreted from two different views. One interpretation is to see $\mathcal{L}_{\text{LDA}}$ as a regularizer on the data-term $\mathcal{L}_{\text{data}}^{\text{SGP-LVM}}$. However, an equally valid view is to see the data-term as a regularizer on the LDA term. In this paper, we take the later view. By setting λ to a large value we force the latent representation to strongly reflect the shared class correlated information in the data. This allows us to evaluate the relation between the class and the non-class information contained in each separate information space.

Inference. Having learnt a model of the observed data means that we have found the latent representation $\mathbf{Z}$ of the training data and the hyper-parameters ϕ specifying the generative mappings. Factorizing the joint probability of the observation streams by the latent representation $\mathbf{Z}$ allows us to specify a distribution over any input feature given the latent representation. This similarly means that given any subset $\mathbf{Y}^S \subseteq \mathbf{Y}$ of the observed features we can infer the latent location by finding the location that maximizes the marginal likelihood,

$$\hat{\mathbf{z}} = \text{argmax} \prod_{v}^{M_S} p(\mathbf{y}^v|\mathbf{z}, \phi_v). \tag{5}$$

The maximum of Eq. (5) is found using gradient based methods. This means that the latent locations need to be initialized. In this paper, we initialize the latent location by taking the nearest-neighbors in the feature training data and initializing using the associated latent location. Our final estimate is the solution corresponding to the highest likelihood solution.

We are interested in inferring the class label t_i associated with a specific subset of the feature observations. To do so, we learn a Gaussian Mixture Model over the latent space from which the posterior distribution over each class given a latent location can be evaluated. This means that given a latent location $\mathbf{z}$ we can evaluate the conditional distribution for this point to be associated with each class t.

2.2 Bayesian Network Model

A Bayesian network [10] is a probabilistic graphical model that encodes the probabilistic distribution of a set of random variables $X = \{X^1, X^2, \ldots, X^m\}$. Each node in the network represents one variable, and the directed arcs represent conditional independencies. Given a structure of the network S and a set of local conditional probability distributions (CPDs) of each variable X^v, the joint distribution of all the variables can be decomposed as

$$p(\mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}, S) = \prod_{v=1}^{m} p(\mathbf{x}^v|\mathbf{pa}_v, \boldsymbol{\theta}_v, S) \ , \tag{6}$$

where $\mathbf{pa}_v$ denotes the parents of node X^v, and the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$ specifies the CPDs. Learning a BN includes discovering from a dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

1) how one variable X^v depends on others $X^{l \neq v}$ (the CPDs encoded by $\boldsymbol{\theta}$), and *2)* what the conditional in-dependencies between different variables in X are (the structure of the network S). The former is an instance of parameter learning and the latter of structure learning. Various algorithms and techniques have been developed to learn a BN in different model and data conditions (see [15, 16] for a review).

In this paper, we use the BN-based task constraint model that has been developed in our previous work [9] (shown in Fig. 2). The variables in the network include four groups $X = \{T, O, A, C\}$, where we choose a subset of all the O, A, C features to instantiate the network. Given that X includes both discrete (T) and continuous variables (O, A, C), the model is a mixed Bayesian network. We model the discrete variable T as multinomial distribution. For the continuous variables, we use Gaussian Mixture Model (GMM) to represent the relatively complex distributions. The number of components of the GMM for a variable X_v is determined from its training instances $\mathbf{X}^v$ based on a Bayesian information criterion. In the BN, the node with GMM distribution has a discrete latent parent to store the mixture coefficients.

The structure, as exemplified in Fig. 2, is determined by human experts. The coarse structure of this BN is reflected by the connection between the four groups of feature variables: T is the root node that parent all the features in O and C. O features parent all the C features, and C features parent all the A features. The fine structure is shown as the intra-group connections. For ease of interpretation, the BN model in Fig. 2 does not show the latent variables for the nodes with GMM distributions.

Inference. Having learned the parameter $\boldsymbol{\theta}$ of a BN from training data $\mathbf{X}$, the model encodes the joint distribution of all the variables X. Since the structure S of the BN encodes the probabilistic relations between T and all the features of O, A and C, we can now compute the posterior distribution of one or group of variables given the observation on others. A common way of this computation is to convert the graph into a tree, then apply the junction tree algorithm [17], an efficient algorithm of local message passing to compute the distribution of the interests.

Our interests in inference using BNs can be illustrated by the *Applying in Imitation* block in Fig. 1. We can apply a BN model in estimating the posterior distribution of a task given observations of the object and action features of a human demonstration, *i.e.* to classify T. We can also find, given an assigned task, the posterior distribution of the object features. This provides an evaluation of the task affordance of a given object, hence allows the robot to select object in complex environments.

In Section 3 we present the task classification results generated by the BN model from our previous work [9].

2.3 Comparison

As stated previously, the two different models presented above describe two different modeling strategies. The BN uses a complex structure to factorize the joint distribution which allows handling involved inter-dependencies between the different variables. In comparison, the LDA-SGP-LVM model uses a simple structure where each observed variable is assumed to be independent given a single shared latent variable. However,

(a) O, A, C

0.96	0.01	0.03	0.00
0.60	0.37	0.01	0.02
0.52	0.00	0.48	0.00
0.44	0.00	0.00	0.56

(b) $A_6 : fcon$

0.48	0.14	0.34	0.04
0.09	0.40	0.11	0.40
0.30	0.13	0.53	0.04
0.14	0.34	0.01	0.51

(c) $C_7 : gbvl$

0.95	0.00	0.05	0.00
0.23	0.61	0.02	0.14
0.70	0.00	0.29	0.01
0.06	0.21	0.00	0.73

Fig. 3. Confusion matrices on classification of 4 tasks: *moving, hand-over, pouring, tool-use*, which (a) all features (O, A, C) are observed; (b) when only $fcon$ is observed, and (c) when $gbvl$ is observed. The result is from LDA-SGP-LVM.

the structural freedom of the BN comes with a penalty as each factor of the joint distribution can only be modeled using a relatively simple model. The BN in our experiments uses a mixture model with a pre-defined set of mixture components to represent each conditional distribution.

In comparison, the simpler structure defined by the SGP-LVM model allows us to leverage the advantage of non-parametric Bayesian modeling which can be interpreted as the mixture model with a infinite number of components.

3 Experimental Results

For the experiments with the LDA-SGP-LVM model we selected 800 randomly sampled observation instances $\mathbf{Y}_{\text{train}}$ uniformly distributed over task and object type. We used 400 instances for training and the remaining to test the model.

Due to the complexity associated with training in the BN framework we selected by hand a sub-set of features we thought were relevant for task classification.

In Fig. 3 we show the confusion matrices associated with the LDA-SGP-LVM model over the 4 tasks. As can be seen the model cannot discriminate between moving and the remaining three classes for a significant portion of the data. This is not surprising as there is hierarchal structure to the classes where *hand-over*, *pouring* and *tool-use* are subsets of *moving*.

However, it can be seen from the two right-most confusion matrices that by only using a sub-set of the feature for observation improve the classification results for certain task. In specific the constraint feature *gbvl* performs significantly better compared to the full observation space in classifying *hand-over* and *tool-use*. The action feature *fcon* is particularly good at separating *moving* from the three other classes but cannot disambiguate between *hand-over* and *tool-use*.

The intuition behind this is that when inferring the latent location from the observed feature subset, we wish to use a combination of features which reduces the entropy over the joint distribution.

To respect the class-hierarchy we will in the reminder of the paper focus on the subset of the data which have not been associated with label *moving* under the models. To this end we will use the row-normalized 3-by-3 sub confusion matrices associated with *hand-over*, *pouring* and *tool-use*.

	O			O, A			O, A, C		
LDA-SGP-LVM	0.71	0.06	0.23	0.74	0.07	0.19	0.93	0.02	0.05
	0.00	1.00	0.00	0.02	0.98	0.00	0.00	1.00	0.00
	0.21	0.00	0.79	0.10	0.00	0.90	0.00	0.00	1.00
Mixed BN	0.51	0.15	0.34	0.56	0.35	0.09	0.70	0.21	0.09
	0.22	0.78	0.00	0.13	0.87	0.00	0.11	0.89	0.00
	0.15	0.10	0.75	0.12	0.00	0.88	0.11	0.00	0.89

Fig. 4. Confusion matrices on classification of 3 tasks: *hand-over, pouring, tool-use*. Three different amount of observations are shown in three columns. The first row are the results from *LDA-SGP-LVM*. In the second row the result is from *Mixed BN* from [9].

3.1 $\mathbf{O, A, C}$ Complement Each Other for Task Classification

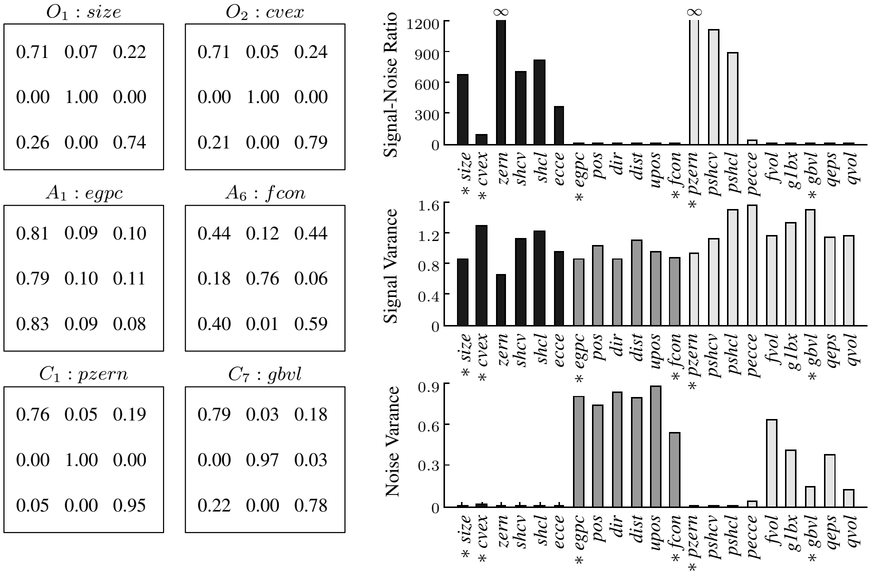
In Fig. 4 the 3-by-3 confusion matrices for LDA-SGP-LVM model and the BN is shown. Comparing the two different models we see that the classification results for the LDA-SGP-LVM model which uses all the available sensory streams outperforms the BN model. Further, it is interesting to note that given only the object features the LDA-SGP-LVM model is capable to perfectly classify the pouring task. This is because the objects that can be poured have a smaller variance in the object features compared to the non-pour-able objects. This implies that the conditional distribution over object feature will be "tighter" when generated from a latent location associated with pouring compared to the other classes therefore rendering a higher likelihood.

intuitively this might seem strange, however, there is a clear rational behind the models decision. For example, say that all the pour-able objects are of the same size, *i.e.* its a hard-constraint for the task. Given an object of this size, the model knows that objects of this size can also be associated with different tasks. However, assuming each task to be equally probable, satisfying the hard-constraint of the *pouring* task means it will be associated with the highest likelihood.

3.2 Different Features Show Different Properties

The LDA-SGP-LVM model provides a powerful way of evaluating the underlying properties of each individual features, which provide a basis for realizing GDP. The four feature properties we will present in this section in Fig. 5) include: *1)* confusion matrix, *2)* signal-noise ratio, *3)* signal variance, and *4)* noise variance. We will first explain the meaning of the properties *2)–4)* from the underlying principle of SGP-LVM model.

Learning a SGP-LVM means that we are trying to find a latent structure and a set of hyper-parameters that can best regenerate the observed data under the GP prior. By the application of a class based prior over the latent space we are in addition learning a representation which respect a class separation. In this paper we are interested in task classification, therefore we want a model that "extracts" relevant variance and "explains



$O_1 : size$

$$\begin{bmatrix} 0.71 & 0.07 & 0.22 \\ 0.00 & 1.00 & 0.00 \\ 0.26 & 0.00 & 0.74 \end{bmatrix}$$

$O_2 : cvex$

$$\begin{bmatrix} 0.71 & 0.05 & 0.24 \\ 0.00 & 1.00 & 0.00 \\ 0.21 & 0.00 & 0.79 \end{bmatrix}$$

$A_1 : egpc$

$$\begin{bmatrix} 0.81 & 0.09 & 0.10 \\ 0.79 & 0.10 & 0.11 \\ 0.83 & 0.09 & 0.08 \end{bmatrix}$$

$A_6 : fcon$

$$\begin{bmatrix} 0.44 & 0.12 & 0.44 \\ 0.18 & 0.76 & 0.06 \\ 0.40 & 0.01 & 0.59 \end{bmatrix}$$

$C_1 : pzern$

$$\begin{bmatrix} 0.76 & 0.05 & 0.19 \\ 0.00 & 1.00 & 0.00 \\ 0.05 & 0.00 & 0.95 \end{bmatrix}$$

$C_7 : gbvl$

$$\begin{bmatrix} 0.79 & 0.03 & 0.18 \\ 0.00 & 0.97 & 0.03 \\ 0.22 & 0.00 & 0.78 \end{bmatrix}$$

Fig. 5. *Left*: Confusion matrices of 6 individual features. The 3 tasks are: *hand-over, pouring, tool-use*. The results are from the LDA-SGP-LVM model. *Right*: Signal-Noise Ratio, Signal Variance and Noise Variance for all the 21 features.

away" the non-class correlated variance in the data. Therefore by enforcing a strong class-separation the non-class correlated variance in each observation space has to be explained by the noise model. This means, as we are estimating the noise variance, that we can evaluate the amount of variance in the data that is relevant for task classification by the signal-noise ratio of the generative mapping.

So in the result Fig. 5, the *confusion matrix* of each feature gives a sense of how well this feature can disambiguate the three tasks; the *signal-noise ratio* indicates the task-relevance of the feature; the *signal variance* implies how "smoothly" the feature varies with the task class, *i.e.* a large signal variance means that the feature generalize well over class; though the *noise variance* is redundant given the former two, we choose to include it for completion.

Examining results in Fig. 5, we have the following interesting observations. Firstly, $O_2 : cvex$, $A6 : fcon$ and $C_7 : gbvl$ are good for task classification, as reviewed by their confusion matrices, but they both have low signal-noise ratios. This tells us that the main information in the features is noise, *i.e.* non-class correlated. In other words, the features are good for task classification, but do not for generalize well over tasks which means they are likely to be non-robust. Connecting to the concept of GDP (*e.g.* selecting features for BN models Fig. 1), if the purpose of the model is only to classify the task (*e.g.* Human BN p^H), these features are good. But if, for example, we want to apply the BN model to select objects for a task (*e.g.* Robot BN p^R), $cvex$ will not be a good candidate to represent the object. On contrary, we notice O_1 : $size$ and O_2 : $pzern$ are good for both.

Secondly, the hand preshape A_1 : $egpc$ can not classify the three tasks well (see its confusion matrix). This is one surprising result as hand preshape is commonly seen as an output of the grasp planning system. This finding implies the preshape is not task relevant.

Finally, the constraint features have slightly higher signal variance compared to object and action features. This implies they generalize better over tasks. This justify the introduction of constraint features to model the tasks.

4 Discussion and Future Work

The suggested LDA-SGP-LVM model is a generative model where we encourage a strong class-based representation on the shared latent representation of the data. Each observed data-space is assumed to have been generated by a mapping modeled using a GP corrupted by gaussian noise. Due to the strong class based prior representing any non-task correlated information in the latent representation will result in a significant penalty to the objective. Further, due to the completely shared structure of the model variance that is contained in a subset of the observation spaces but not all will if represented on the latent space "pollute" the model for the other observation models. This means that dominantly non-shared and non-task correlated variance needs to be "explained away" in the using the noise model. However, the noise is assumed to be Gaussian something which is most likely a much to "crude" assumption to make.

In [18] and [19] the SGP-LVM model was extended to include *private* latent spaces, $i.e.$ spaces which are used to only model a single observation space. These spaces were [18] interpreted as "structured" noise models explaining away non-shared variance in the data. Similarly, we would like to adopt such a strategy but in addition use the class prior only over the shared latent representation. This we hope would lead to a better generalization of the data allowing us in addition to task-classification further use the models generative capabilities.

In the next phase towards GDP we aim to use the knowledge and insight about the structure in the observed data in order to create a more efficient BN. Further, we are also currently exploring non-parametric approaches to directly learn the structure of Graphical Models from data such as presented in [20].

5 Conclusion

The motivation behind this work was to evaluate the possibility of using a significantly different graphical model to gain knowledge about the task relevant structure in the data in order to support designing a new BN structure. To do so, we extended the SGP-LVM model to handle more than two observation spaces and to incorporate class based priors.

Using the LDA-SGP-LVM model in the paper we show an improved task classification performance compared to the previously proposed BN model.

By evaluating the ratio of class correlated to the non-class correlated variance of each feature we have gained a notion of what specific parts contain the most relevant class information.

References

1. Kang, S., Ikeuchi, K.: Toward Automatic Robot Instruction from Perception-Mapping Human Grasps to Manipulator Grasps. Robotics and Automation, IEEE Transactions on (1997)
2. Nehaniv, C.L., Dautenhahn, K., eds.: Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions. Cambridge University Press (2004)
3. Montesano, L., Lopes, M., Bernardino, A., Santos-Victor, J.: Learning Object Affordances: From Sensory–Motor Coordination to Imitation. Robotics, IEEE Transactions on (2008)
4. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and Generalization of Motor Skills by Learning from Demonstration. In: ICRA. (2009)
5. Greeno, J.: Gibson's Affordances. Psychological Review (1994)
6. Romero, J., Kjellström, H., Kragic, D.: Monocular real-time 3d articulated hand pose estimation. IEEE-RAS International … (2009)
7. Alissandrakis, A., Nehaniv, C., Dautenhahn, K.: Correspondence Mapping Induced State and Action Metrics for Robotic Imitation. Systems, Man, and Cybernetics, Part B, IEEE Transactions on (2007)
8. Meltzoff, A.N. In: Elements of a Developmental Theory of Imitation. Cambridge University Press, Cambridge, MA, USA (2002) 19–41
9. Song, D., Huebner, K., Kyrki, V., Kragic, D.: Learning Task Constraints for Robot Grasping using Graphical Models. In: IEEE International Conference on Intelligent Robots and Systems 2010. (2010) To appear.
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
11. Lawrence, N.D.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. The Journal of Machine Learning Research **6** (2005) 1783–1816
12. Urtasun, R., Fleet, D., Geiger, A., Popović, J., Darrell, T., Lawrence, N.: Topologically-constrained latent variable models. (2008) 1080–1087
13. Urtasun, R., Darrell, T.: Discriminative gaussian process latent variable model for classification. Proceedings of the 24th international conference on Machine learning (2007) 927–934
14. Shon, A., Grochow, K., Hertzmann, A., Rao, R.: Learning shared latent structure for image synthesis and robotic imitation. Proc. NIPS (2006) 1233–1240
15. Heckerman, D.: A Tutorial on Learning With Bayesian Networks. Technical report, Microsoft Research (1996)
16. Cowell, R., Dawid, P., Lauritzen, S., Spiegelhalter, D.: Probabilistic Networks and Expert Systems. Springer-Verlag New York (1999)
17. Huang, C., Darwiche, A.: Inference in Belief Networks: A Procedural Guide. International Journal of Approximate Reasoning **15** (1994) 225–263
18. Ek, C.H.: Shared Gaussian Process Latent Variable Models. PhD thesis (2009)
19. Salzmann, M., Ek, C.H., Urtasun, R., Darrell, T.: Factorized orthogonal latent spaces. (2010)
20. Adams, R.P., Wallach, H.M., Ghahramani, Z.: Learning the structure of deep sparse graphical models. (2010)