Task Modeling in Imitation Learning using Latent Variable Models

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

Abstract—An important challenge in robotic research is learning and reasoning about different manipulation tasks from scene observations.

In this paper we present a probabilistic model capable of modeling several different types of input sources within the same model. Our model is capable to infer the task using only partial observations. Further, our framework allows the robot, given partial knowledge of the scene, to reason about what information streams to acquire in order to disambiguate the state-space the most.

We present results for task classification within and also reason about different features discriminative power for different classes of tasks.

I. INTRODUCTION

A major challenge in robot grasping is to automatically plan a grasp on an object that affords a specific task. Though proved to be an effective approach, learning by imitation [1], presents significant 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 [2]. The problem becomes even harder when the goal is to perceive these features through vision systems [3]: human hands have many fingers that are often blocked by 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.

These challenges are not new in the field of robotics. To cope with the uncertainty in the sensorimotor systems, [4] proposed a coherent control, trajectory optimization, and action planning architecture. They applied the probabilistic inference-based methods and the dynamic Bayesian networks to integrate across all levels of representations. Another work [5] addresses the challenges in the robotics application with high degrees of freedom. They identified, from high-dimensional movement data, a latent space representation for tasks, such as drawing on a table plane. A control policy learned in this latent space is powerful to regenerate new movements.

For the robot manipulation task, a recent work [6], adopt a self-supervised, developmental approach where the robot first explores its sensory motor capabilities, and then interacts with objects to learn their affordances. A discrete Bayesian network (BN) [7] is used to capture the statistical dependencies between actions, object features and the observed effects of the actions. Our recent work [8], extended this idea by incorporating task information. The work propose a task constraint model based on a mixed BN, which links the symbolic task requirements to continuous real robot sensory data of attributes on both objects and grasping actions. The framework successfully realizes goal-directed imitation in robot grasping tasks. The limitation, however, lies in the inability of the BNs to model high dimensional, multi-channeled sensory data. When the number of nodes which model the sensory streams is high, the training of BNs becomes intractable. Further, the complexity in the structure of the conditional dependencies modeling significantly limits the flexibility of the model.

In current framework, we focus on building a latent task observation model (see the Latent Variable Model in Fig. 1). The work relates to that of [5] in that, we want to model a reduced dimensional representation of task. The difference, however, lies in the problem domain: we do not address the movement regeneration as in [5], rather focusing on modeling the task related affordances of objects and robot embodiments. The work can be integrated with the embodiment-specific BNs, as presented in [8] and Fig. 1, to realize goal-directed imitations.

In this paper, we employ the Shared Gaussian Process Latent Variable Model (SGP-LVM) [9], which assumes a factorization of the joint distribution of the data into independent conditional distributions given a shared latent variable. The advantage of such a simple structure is that it allows us to model the conditional dependencies using flexible Gaussian Process mappings (GP) which can be viewed as infinite mixture models whereas in the BN approach we are limited to a fixed set of mixture components. Further, using the SGP-LVM we are not forced to limit the observation space but can leverage the advantage of using the full observed data.

A. Notation

Upper-case letters identify set variables, where \( Y \) specifies the full set of all observed variables \( Y = \{ O, A, C \} \). Superscript \( S \) is used to refer to a sub-set of the variables, \( Y^S \subseteq Y \), the letter \( M_S \) is the cardinality of the feature subset, \( M_S = |Y^S| \). 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 sub-script index \(i\). For example, \(Y = (y_1, \ldots, y_N)^T\) represent \(N\) cases of instantiations of all the variables in the set \(Y\). Then, all the \(N\) instantiations of variable \(Y_i\) will be \(Y_i\), and the \(i^{th}\) specific instantiation of \(Y_i\) will be \(y_i\). The \(i^{th}\) specific instantiation of all the variables in \(Y\) will be a concatenation of the variables \(y_i = [y_i, \ldots, y_i]\), where \(M = |Y|\). Last, we introduce \(D\) to be the dimensionality of an instantiation. Note that \(D \geq M\), since \(M\) represents the number of possibly multi-dimensional variables.

II. DATA GENERATION

In this subsection, we will briefly introduce how our 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 [8]. 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 our features by using a simulation-based architecture and grasp planner. For details on those modules and the tutor-based task labeling process see [8].

For each good grasp \(i\) provided by the grasp planner, one training dataset \(y_i\) is generated. While in [8], we used a small network with \(M=7\) features to analyze Bayesian Network learning, in here we use \(M=21\) features. 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.

III. GAUSSIAN PROCESS LATENT VARIABLE MODELS

In generative dimensionality reduction the observed data \(Y\) is assumed to have been generated from a low-dimensional latent variable \(Z\) through a mapping \(f\), \(y_i = f(z_i)\). Assuming the observed data to have been corrupted by additive noise leads to the likelihood of the data,

\[
p(Y) = p(Y|Z,f)p(Z)p(f).
\]

The Gaussian Process Latent Variable Model (GP-LVM) [10] is a generative model for dimensionality reduction where the generative mapping \(f\) is modeled as a product of independent Gaussian Processes (GP) which leads to the
following marginal likelihood,

\[ p(Y|Z, \phi) = \int p(Y|f)p(f|Z, \phi)df, \quad (2) \]

where \( \phi \) is the hyper-parameters specifying the GP.

In the GP-LVM framework we wish to find the latent locations and the hyper-parameters that maximizes the posterior distribution of the data,

\[ p(Z, \phi|Y) \propto p(Y|Z, \phi)p(Z)p(\phi). \quad (3) \]

However, as estimating partitioning function is intractable we in practice proceed to minimize the negative log of the product of the margin likelihood the latent prior and the prior over the hyper-parameters. This leads to the following objective,

\[ \mathcal{L} = \mathcal{L}_{\text{data}} + \sum_i \ln \phi_i + \sum_i \frac{1}{2} |z_i|^2 + C, \quad (4) \]

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

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. 4.

1) \textit{Shared GP-LVM:} The Shared GP-LVM (SGP-LVM) [9] is an extension which learns a single latent representation from which two observed data spaces are generated by separate GP’s. In this paper, we extend the original SGP-LVM to model the 3 different observation spaces \( Y = \{O, A, C\} \). This means that we assume each of the observed data spaces being independent given the latent representation resulting in the following factorization,

\[ p(Y) = p(Y|Z, \Phi) = p(O|Z, \phi_O)p(A|Z, \phi_A)p(C|Z, \phi_C). \quad (5) \]

Being a generative model with a data-term reflecting reconstruction all the variance in the observed data needs to be represented within the model. The SGP-LVM model models each of the observed data-spaces to have been generated from the same underlying latent variable, this implies an assumption that all the variance in each of the observation spaces is shared, variance which is not needs to be “explained away” as Gaussian noise using the noise model. However, for many data-sets it is unlikely for the non-shared variance to be well described using this model leading to a bad fit of the model to the data.

In order to proceed in such situations it was suggested in [9] to extend the SGP-LVM model to also include “private” latent spaces. These spaces models variance in a single observation space which is not shared by the others can be interpreted as “structured noise models”. Including such private spaces into our model leads to the following marginal likelihood,

\[ p(Y) = p(Y|Z^S, Z^A, Z^O, Z^C, \Phi) = p(A|Z^S, Z^A, \phi_A)p(O|Z^S, Z^O, \phi_O)p(C|Z^S, Z^C, \phi_C)\delta \]

In this model the variance which is shared between \( \{A, O, C\} \) will be represented using \( Z^S \) while the non-shared variance in each space will be represented by \( Z^A, Z^O \) and \( Z^C \) for \( A, O \) and \( C \) respectively.

In this project we are not just interested in factorizing the latent representation into variance that is shared from such which is private. Specifically we are interested in variance which is shared \textit{and} that contains task correlated information. Such a factorization can be found by replacing the uninformative prior \( p(Z) \) with a distribution that encourages class separation. Such an extension to the standard GP-LVM model was suggested in [11] by including a term based on Linear Discriminant Analysis (LDA).

By noting that our model factorises the latent prior into 4 distinct parts, \( p(Z) = p(Z^S)p(Z^A)p(Z^O)p(Z^C) \). We replace the uninformative prior over the shared space with the LDA prior from [11].

\[ p(Z^S) = \frac{1}{C_S} \exp - \text{tr} \left(S_w^{-1}S_b\right), \quad (7) \]

where \( S_w \) and \( S_b \) are the within and between class scatter matrices of the latent representation respectively. In this way we will encourage the shared space to contain information which is shared between the observation spaces and contains class correlated information. Figure shows the graphical model.

2) \textit{Training:} The objective function above is non-convex and as the solution is sought through gradient based optimization we are dependent on a good initialization of the latent spaces. Further, the dimensionality of the latent spaces are free variables which needs to be estimated from data. These two characteristics of the model poses significant limitations as it for many types of data is non-trivial to estimate the dimensionality and a initialization, specifically for factorized models such as ours.

In recent work [12] a set of regularizers to the SGP-LVM framework has been suggested referred to as FOLS. The FOLS regularizer reduces the demands on the initialization (in practice the latent space is initialized with the observed data) and is capable of learning the dimensionality in addition to factorizing the latent space into non-redundant subspaces. It does so by encouraging solutions with a low-rank covariance matrix and penalize solutions where the shared and the private spaces are non-orthogonal. For a more complete description of the model see [12]. We add the FOLS regularizers to our model which leads to the following objective,

\[ \mathcal{L}_{\text{LDA-SGP-LVM-FOLS}} = \mathcal{L}_{\text{SGP-LVM}} + \lambda \cdot \mathcal{L}_{\text{LDA}} + \beta \cdot \mathcal{L}_{\text{FOLS}}. \quad (8) \]

The scalar \( \lambda \) controls the trade-off between reconstruction and class separation and \( \beta \) controls the relative scale of the FOLS regularizer.

The term \( \mathcal{L}_{\text{LDA}} \) will encourage a latent representation with large class separation and low within class variance. As noted in [11], the model can be interpreted from two different views. One interpretation is to see \( \mathcal{L}_{\text{LDA}} \) as a regularizer on the data-term. 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 \( \lambda \) to a large value we
force the shared latent representation to strongly reflect the
shared class correlated information in the data. This allows us
to evaluate the relative amount of the variance in each
observation space which is task correlated.

3) Inference: Training the above presented model means that we have learnt the latent representation \( Z = \{ Z^O, Z^A, Z^C \} \) and the hyper-parameters \( \Phi = \{ \phi_A, \phi_O, \phi_C \} \) specifying the three generative mappings.

The factorization of joint probability of the observations specified by the model allows us to specify a distribution over any input space given the known latent representation. This means that given any subset of the observation spaces \( Y^S \subseteq Y \) of the observed features we can infer the latent location by finding the location that maximizes the marginal likelihood,

\[
\hat{z} = \arg \max_z \prod_v p(y^v | z, \phi_v).
\] (9)
The maximum of (9) 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 (throughout the experiments in this paper we use 10 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 task label associated with a specific subset of the feature observations. To do so, we learn a Gaussian Mixture Model over the shared class-constrained latent space from which the posterior distribution over each class given a latent location can be evaluated. This means that given a location \( z^S \) we can evaluate the conditional distribution for this point to be associated with each task.

IV. EXPERIMENTAL RESULTS

For the experiments with the LDA-SGP-LVM model we selected 900 randomly sampled observation instances \( Y \) uniformly distributed over task. We used 450 instances for training and the remaining to test the model.

A. Task Classification

We first test our models performance in classifying the observed task, we use the MAP solution under the GMM model over the latent space. To understand the effect of the classification performance with respect to different observation domains we perform the classification on each permutation of the observed features, the results are compared to a set of baseline algorithms in Tab. IV-A.

Compared to the purely discriminative base-line algorithms we expect our model to perform worse as it is both learning a representation for task discrimination and for

<table>
<thead>
<tr>
<th>( A )</th>
<th>( O )</th>
<th>( C )</th>
<th>( A, O )</th>
<th>( A, C )</th>
<th>( O, C )</th>
<th>( A, O, C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-LR</td>
<td>78</td>
<td>94</td>
<td>88</td>
<td>90</td>
<td>95</td>
<td>92</td>
</tr>
<tr>
<td>SVM-RBF</td>
<td>84</td>
<td>88</td>
<td>84</td>
<td>92</td>
<td>94</td>
<td>94</td>
</tr>
<tr>
<td>LVM</td>
<td>88</td>
<td>94</td>
<td>88</td>
<td>92</td>
<td>94</td>
<td>94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( O )</th>
<th>( A )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.76</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>0.08</td>
<td>0.92</td>
<td>0.00</td>
</tr>
<tr>
<td>0.59</td>
<td>0.00</td>
<td>0.41</td>
</tr>
<tr>
<td>( O )</td>
<td>( O, A )</td>
<td>( O, A, C )</td>
</tr>
<tr>
<td>0.76</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>0.08</td>
<td>0.92</td>
<td>0.00</td>
</tr>
<tr>
<td>0.59</td>
<td>0.00</td>
<td>0.41</td>
</tr>
<tr>
<td>( O^S )</td>
<td>( O^S, A^S )</td>
<td>( O^S, A^S, C^S )</td>
</tr>
<tr>
<td>0.51</td>
<td>0.15</td>
<td>0.34</td>
</tr>
<tr>
<td>0.22</td>
<td>0.78</td>
<td>0.00</td>
</tr>
<tr>
<td>0.15</td>
<td>0.10</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Fig. 2. Confusion matrices on classification of 3 tasks, hand-over, pouring, tool-use, given observation on different sub-sets features in \( \{ O, A, C \} \). The first two row are the results from the Latent Variable Model. The third row is the results from the Mixed BN in previous work [8]. The superscript \( S \) represent subset of the features, e.g. \( O^S \subseteq O \) reconstruction and generation. Further, in our model it is possible to infer the task from any observed subspace while for the baseline algorithms a separate model needs to trained for each input combination.

Taking this into consideration we believe that our model performs competitively.

In Fig. 2 the confusion matrices for task classification for the proposed model and our previous BN based model is shown. The first row shows the results using each of the observed features in turn. We can see that the object features are generally good at discriminating the hand-over and pouring task but confuses tool-use with hand-over. The action features in comparison provides a good indications for each task but are worse than the object features in discriminating pouring. This is to be expected as the objects that are pour-able are quite specific while the pouring action itself can be ambiguous. Finally, for the constraint features we see that compared to object the are capable to clearly discriminate the task of tool-use. This is also something we would expect as this task poses quite specific constraints.

Comparing the result of the proposed model with our previous work we can see that when we only have a limited available our model performs significantly better while when given additional input features our models performance drops. This is an indication of the mismatch between the structure of the dependencies in our model compared to the data, where the single layer model is too simplistic while the more complex BN model is closer to the actual data.

B. Different Features Show Different Properties

By incorporating the FOLS regularizers into the learning framework we are able to infer the dimensionality of the latent factorization from data. In our experiments this resulted in a model with 2 shared dimensions and 5,2 and 2 private dimensions to represent \( A, O \) and \( C \) respectively.

In Fig. 3 we see the action features containing the largest proportion of non-task correlated variance among our three
feature classes. This is not surprising as we expect there to be a larger variability in the manner an action is performed compared to the object and constraints for a specific task.

For imitation learning we are interested in learning what features are relevant and what combination of features reduces the entropy of the task distribution the most. However, this is clearly task dependent as say for example knowing the object is not likely to be particularly discriminative between the task of pouring and hand-over while providing very useful knowledge for choosing between pouring and tool-use. If such information could be extracted we would be able to direct the robot's perception in such manner to acquire information to maximally reduce the entropy over tasks. Our model being generative we can exploit the shared-private latent structure to extract such information.

For an observed data point \( y_{Si} \), we can evaluate the marginal likelihood \( p(y_{Si} | z_i, Z) \) of the latent location under the model. In this model this factorizes as follows,

\[
p(y_{Si} | z_i, Z) = p(y_{Si} | z_i^S, Z^S, Z^O).
\]

As the location in the shared latent sub-space \( z_i^S \) represents task correlated variance in the data we can by fixing the location in the private subspace \( z_i^O \) (which represents the non-shared non-task correlated variance) evaluate the effect of the marginal likelihood with respect to the shared variable for different observation features.

In Fig. 4 the result of uniformly sampling the marginal likelihood over the shared latent space while keeping the private location fixed for the three different tasks and observation features are shown. The first row shows how the likelihood varies for an instance of the hand-over task. In the first column the likelihood for the objective features are shown, it can be seen that the probability mass is distributed over all training data locations. This is to be expected as the task hand-over can be performed on every object. The middle column shows the results for the action features, which have a large region associated with a high and two regions with low likelihood values. Identifying the training data location we can see that the region of high-likelihood corresponds to hand-over while the low-likelihood regions correspond to pouring. This is sensible as the action performed when pouring is significantly different to the one for hand-over. In the last column the distribution for the constraint model is shown. As we can see the distribution have a very high entropy being nearly completely flat which implies that there is very little information in the constraints for the task of hand-over.

By similar reasoning, we can see in the second row that for the task of pouring the object features are less discriminative while the action features places a high-probability mass around an area occupied by pouring and tool-use instances. More importantly, here, the constraint features are very discriminative with very dominant mode which identifies the task as pouring.

Finally, in the third row, the results for tool-use is shown. Here we can see that each of the three features are multimodal and all provide relevant information to reason about the task.

Further, for each of the three examples shown above we note that the distribution over the action features is a lot less peaked in comparison with the object and action features. This is to be expected as there is a much larger variability in realizing a grasp compared to object and constraint features.

The above results clearly exemplifies the strengths of our model, by learning a factorized model we can evaluate given an observation in one feature space which feature we should acquire in order to reduce the entropy in the decision of task the most. Further, observing how the probability mass is distributed over the shared latent space gives a notion of the generative qualities of our model. The object and constraint distributions are tightly centered around the data while for the action features it is more broadly distributed. This is good as the first two often poses next to hard constraints on the grasp while for the action this shows that there are several possible ways to realize a specific action.

V. CONCLUSION

In this paper we have presented an extension to the Shared Gaussian Process Latent Variable model to more than two observation spaces and by incorporating partial discriminative constraints. Further, we have including recently proposed latent regularizers in order to learn a non-redundant latent representation of the data.

The proposed model is applied to factorize the joint distribution of a set of different features for the robotic task of grasping. Our generative model performs comparatively to purely discriminative methods for task classification. Further, by exploiting the generative nature of our model we are able to evaluate the discriminative power of different features to different tasks, something which can be very useful in scenarios with limited observations.

Robots and humans have significantly different embodiments, this means that the features that are important for humans and robots in order to perform a specific task successfully are likely to be different. In our previous work we bridge this gap by using separate models for the robot and the human, where classification is done in the human model and generation in the robotic model. However, given corresponding instances of robot and human action, the framework developed above is capable of simultaneously learning a latent representation linking the two different embodiments making it possible to contain the full framework within a single model. This is something we would to evaluate in future work.
Fig. 4. Samples of the likelihood under the model. Each images is generated by fixing the location in the private-non-task correlated latent subspace and uniformly sampling over the shared-task correlated. Each row represents a specific task, top to bottom: Hand-Over, Pouring and Tool-Use. The columns represent the three different observation spaces, left to right: Object, Action and Constraint. The location of the training data is shown over the plots, where Magenta, Green and Blue indicates hand-over, pouring and tool-use respectively. Note due to the intractability in estimating the partitioning functions the actual value of the posterior should not be interpreted only the within observation space scale is relevant. For visualization purposes we have scaled and normalized the plots for ease of interpretation.

In this paper we have for clarity of presentation focused on the discriminative power of the model, however, the model is generative and capable of reconstruction outside the training data. As noted in the experimental section, there is indications of the good generative qualities. This is something we will evaluate in upcoming work.

REFERENCES