# Variable Selection in GPDMs Using the Information Bottleneck Method

**Jesse St. Amand**[1], **Martin A. Giese**[1]
[1]Section Computational Sensomotorics, Department of Cognitive Neurology,
Tübingen University Hospital, CIN, HIH, and the University of Tübingen
{jesse.st-amand, martin.giese}@uni.tuebingen.de

## Abstract

Accurate real-time models of human motion are important for applications in areas such as cognitive science and robotics. Neural networks are often the favored choice, yet their generalization properties are limited, particularly on small data sets. This paper utilizes the Gaussian process dynamical model (GPDM) as an alternative. Despite their successes in various motion tasks, GPDMs face challenges like high computational complexity and the need for many hyperparameters. This work addresses these issues by integrating the information bottleneck (IB) framework with GPDMs. The IB approach aims to optimally balance data fit and generalization through measures of mutual information. Our technique uses IB variable selection as a component of GPLVM back-constraints to reduce parameter count and to select features for latent space optimization, resulting in improved model accuracy.

## 1 Introduction

Accurate real-time synthesis models for human motion are of high relevance for cognitive science, computational neuroscience, computer graphics, and robotics. Although neural networks are frequently used in these models, their precision and robustness depend on vast training data sizes. Probabilistic graphical models are often more suitable for this task as they have a strong theoretical foundation and allow for the inclusion of regularization methods appropriate for inference on smaller data sets. However, due to their higher computational complexity, often such models are ill-suited for real-time control and training on large data sets.

The Gaussian process latent variable model (GPLVM)—which can be formulated as a neural network with an infinite number of hidden units (Neal [1995])—is an effective tool for modeling high-dimensional human motion. They have been successfully applied to the acquisition of low-dimensional models (Levine et al. [2012]), inverse kinematics (Grochow et al. [2004]), kinematic modeling, and motion interpolation (Ye and Liu [2010]). While inherently computationally complex, sparse approximation techniques make GPLVMs adaptable for real-time uses and large data sets (Snelson and Ghahramani [2005], Lawrence [2007]). They also perform exceptionally well on data-limited tasks compared to neural networks. The Gaussian process dynamical model (GPDM) is an extension on the GPLVM, adding a dynamical prior for trajectory smoothing and sequence prediction (Wang et al. [2005, 2008]).

A high computational cost and an excess of hyperparameters limit the utility of the GPDM. To compensate, we integrate GPDMs and the information bottleneck (IB) framework to enforce optimal dynamical latent space development. The IB balances data fit with generalization using mutual information for cost and regularization (Tishby and Zaslavsky [2015]). Our objective is to simplify optimization by tightly constraining the set of learnable parameters and decreasing the number of necessary hyperparameters. We present here a novel technique that integrates IB variable selection with GPLVM back-constraints to enhance the model's ability to produce task-suitable latent spaces.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

## 2 Background

### 2.1 GPLVMs

The Gaussian process latent variable model (GPLVM) is a non-linear dimensionality reduction technique closely related to probabilistic PCA (PPCA) (Lawrence [2005]). In GPLVMs, we learn a reduced-dimensional representation of the observed data over a latent space.

Consider a data set $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T \in \Re^{N \times D}$. We define the latent set as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \Re^{N \times Q}$, where $N$ represents the number of data points, $D$ is the number of dimensions in the data space, and $Q$ is the number of dimensions in the latent space. These latent values are learned through likelihood optimization, satisfying the mapping function:

$$y_d = f_d(\mathbf{x}) + \varepsilon_d, \tag{1}$$

where $\mathbf{y} = [y_1, \ldots, y_D]^T$, $\varepsilon_d$ describes the noise, and $f_d(\mathbf{x})$ is drawn from a Gaussian process (GP) prior, $f_d(\mathbf{x}) \sim GP(0, k_{\mathbf{f}}(\mathbf{x}, \mathbf{x}'))$. The kernel function $k_{\mathbf{f}}(\mathbf{x}, \mathbf{x}')$ is chosen respective to the use case.

### 2.2 GPDMs

Dynamics in the latent space are modeled using a Gaussian process dynamical model (GPDM). This model follows a mapping of the form:

$$\mathbf{x}_n = f_{\mathbf{x}}(\mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-z}) + \xi, \tag{2}$$

where $z$ changes to match application in first- or second-order dynamics (Wang et al. [2005, 2008]). The function $f_x$ is drawn from a GP prior with an associated auto-regressive kernel function that generalizes a hidden Markov model (HMM) (Li et al. [2000]). It serves to model state transitions and helps to smooth trajectories in the latent space.

### 2.3 Back-Constraints

Back-constraints (BCs) are functions that inversely map from the data space into the latent space of GPLVMs. In addition to offering an efficient way to project into the latent space, BCs impose additional constraints on its formation. These constraints aim to preserve local distances and can be used to shape the space into specific geometries (Urtasun et al. [2007]).

We apply BCs to prevent overfitting the GPLVM and to encourage the formation of a latent space suitable for dynamics, e.g., by having latent points that make up coherent trajectories.

Below, we describe a new form of back-constraint named the GP BC which allows us to incorporate the IB framework for selecting latent features.

### 2.4 Information Bottleneck Framework

The IB framework models a probabilistic mapping between the variable $X$ and $Y$ through an intermediate bottleneck variable $Z$. It is expressed as the minimization of $I(Z, X) - \beta \cdot I(Y, Z)$, where the mutual information $I(X, Z)$ measures the interdependence between random variables X and Z. Essentially, the inclusion of $I(Z, Y)$ encourages Z to predict Y, while the $I(X, Z)$ enforces a minimal knowledge of X, lowering redundancy. The hyperparameter $\beta > 0$ moderates the trade-off between the compression and fit (Tishby and Zaslavsky [2015]).

## 3 Methods

### 3.1 Gaussian Process Back-Constraints

The GPDM can converge sub-optimally when overfitting the compression mapping relative to the state transitions. Geometric constraints Urtasun et al. [2007] and smooth latent space initial conditions can correct this imbalance. BCs that simply maintain the latent space close to good initial conditions, like a kernel PCA, can dramatically improve performance for this reason.

The GP BC is a method that ensures the convergence of the latent space close to its initial conditions. The model's latent variables are first initialized with an arbitrary dimension reduction of the training

data. A radial basis function (RBF) GP mapping from the data to the latent space is then optimized. We use the regressional posterior of the GP to provide our constraint during learning,

$$X = K_{Y,Y} \cdot K^{-1} \cdot X_0, \tag{3}$$

where $X$ is a matrix of latent variables, $K^{-1}$ is the inverse learned GP kernel matrix, $K_{Y,Y}$ is the kernel mapping of the training data with itself, and $X_0$ is the initialized latent matrix. Optimization of the latent space is thus performed through learning the kernel parameters in $K_{Y,Y}$ (and not in $K^{-1}$).

After optimization, predictive mapping into the latent space is performed by equation (3) updated as,

$$X^* = K_{Y,Y^*} \cdot K^{-1} \cdot X_0, \tag{4}$$

where $Y^*$ designates a set of testing data.

### 3.2 Information Bottleneck Latent Optimization

IB latent optimization (IBLO) begins by defining a latent space with more than the expected number of task-optimal features. Here, we designate $Y$ as the data space and $X$ as the high-dimensional latent space. We aim to optimize an auto-regressive mapping from $y_t$ to $y_{t+1}$.

Within this framework, we optimize a GP RBF mapping from $X$ to $Y$. We then establish an auto-regressive mapping in $X$ using the same type of kernel as in our GPDM dynamics. These GPs furnish us with the probability distributions: P(X) and P(Y|X). The former characterizes the auto-regressive dynamics of $X$, while the latter provides a GP regression from $X$ to $Y$.

Our central objective is to define $Z$ as the random variable to be optimized, serving as the bottleneck between $y_t$ and $y_{t+1}$. We aim to identify a subset, denoted as $x_1, x_2, x_3, ...x_d \subseteq X$, for $Z$ that includes only the dimensions essential for describing $Y$. To assess the relevance of dimension $x_j$, we employ the following indicator function, inspired by the work in Pan et al. [2023]:

$$S_j = [I(Y_t, Z_{t,+j}) - \beta I(Z_{t+1,+j}, Z_{t,+j})] - [I(Y_t, Z_{t,-j}) - \beta I(Z_{t+1,-j}, Z_{t,-j})] \tag{5}$$

Here, the subscripts $-j$ and $+j$ signify whether $z_j$ is not in $Z$ or is included in $Z$, respectively. $I(N, M)$ represents the mutual information between $N$ and $M$, calculated using the multivariate Gaussian entropy. $S_j$ is a measure of how well $Z$ serves as a bottleneck when either retaining or eliminating dimension $x_j$. The greater the value of $S_j$, the more Z will benefit from the removal of $x_j$. $S_j$ is evaluated iteratively on successive removals of dimension $j$. We set $\beta$ by solving for $S_j = 0$ on the evaluation of the first $j$ in the first iteration. This enabled the algorithm to optimally balance features that were pertinent to both compression and dynamical prediction.

## 4 Results

We evaluated the IBLO method by its ability to accurately categorize actions under different conditions. This classification was accomplished by comparing the posterior probabilities of a given sequence for different GPDMs. The data set comprised infrared motion capture of five bimanual manipulation tasks encompassing 117 degrees of freedom.

Our initial trials examined the ability of our IB method to select optimal RBF kernel PCA (kPCA) features compared to the model's performance when utilizing features selected according to the highest eigenvalues. In these cases, our method produced no significant improvement as it nearly always selected the highest eigenvalue features.

As a second comparison, we ran simulations where IBLO selected 10 features from a mixture containing 10 RBF kPCA features and 30 features produced by random projection—a dimensionality reduction technique wherein new features are produced by a linear combination of the features (in our case, $Y$) with weights sampled from a Gaussian distribution. We compared this against a control containing an equal number of features produced only by random projection. From these results, we derived two confusion matrices and accuracy scores based on the number of sequences correctly classified. A t-test comparing these accuracy scores gave a t-statistic of -5.83 with a p-value of $6 \cdot 10^{-7}$. Figure 1 displays a differential confusion matrix constructed from the IBLO scores minus those of the control. We further tested whether the IBLO method could select features better than expected for an arbitrary dimensionality reduction method. We tested the highest 10 eigenvalue

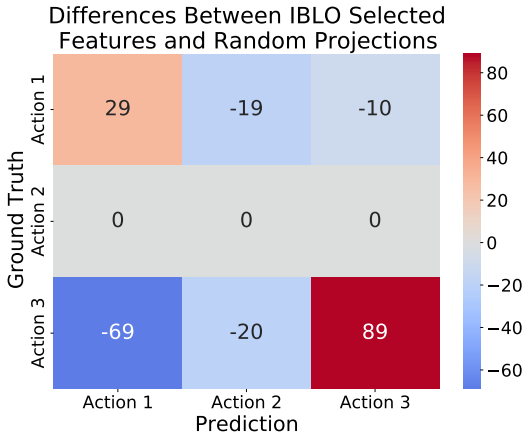


Figure 1: A heat map displaying the difference in sequence classification accuracy for three actions when comparing IBLO against the control. We constructed this heat map by subtracting the confusion matrix of the control from the confusion matrix for IBLO. Positive and negative values indicate areas where IBLO and the control, respectively, had greater scores. The numbers on the diagonal represent scores for the correct outcome, indicating IBLO had improved accuracy when classifying actions 1 and 3. For this test, IBLO selected 10 features from a mixture of 10 RBF kPCA and 30 random projections while the control used 10 random projections. Comparison of the two accuracy scores produced a significant difference with a p-value of $6 \cdot 10^{-7}$

features from PCA and ICA against a selection by IBLO from a set of the first 30 features of both techniques. In this case, IBLO did not use the same features as the control. Figure 2 shows the comparison for both ICA and PCA. In both cases IBLO improved the accuracy, precision, recall, and F1 score of the model.

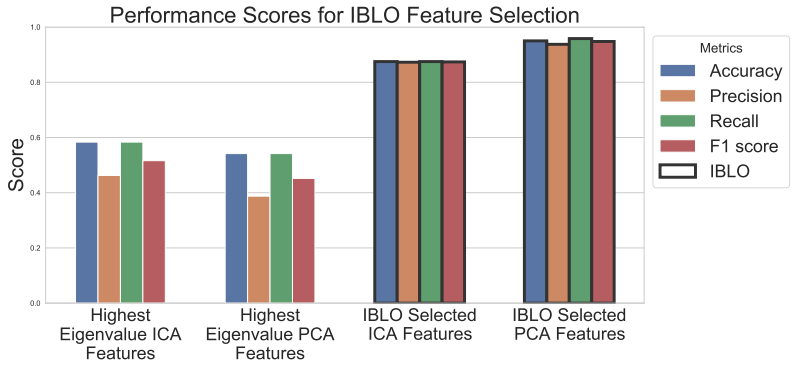


Figure 2: Performance scores for model classification of five sequences using PCA and ICA features. The control model used the first 10 features with the highest eigenvalues, while IBLO selected 10 from the first 30. In both PCA and ICA, IBLO improved performance on metrics of accuracy, precision, recall, and F1 score. Further simulations must be run to determine a significance score, but results suggest a continuation of the displayed trend.

## 5 Conclusion

The results are promising, showing an ability for IBLO to select highly informative kPCA features from sets intermixed with random projections. It also shows a trend towards improved performance when comparing PCA and ICA features chosen according to the highest eigenvalues to those selected by IBLO. Future research will continue to test the performance and utility of the model.

## 6 Acknowledgements

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting JS. This research was funded through the European Research Council ERC 2019-SYG under EU Horizon 2020 research and innovation programme (grant agreement No. 856495, RELEVANCE).

## References

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popovic. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004.

Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models, 2005.

Neil D. Lawrence. Learning for larger datasets with the gaussian process latent variable model. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 243–250, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL `https://proceedings.mlr.press/v2/lawrence07a.html`.

Sergey Levine, Jack M. Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. Continuous character control with low-dimensional embeddings. *ACM Trans. Graph.*, 31(4):28, 2012.

Xiaolin Li, Marc Parizeau, and Réjean Plamondon. Training hidden Markov models with multiple observations-a combinatorial method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(4):371–377, 2000.

R.M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Dept. of Computer Science, University of Toronto, 1995.

Junlong Pan, Weifu Li, Liyuan Liu, Kang Jia, Tong Liu, and Fen Chen. Variable selection using deep variational information bottleneck with drop-out-one loss. *Applied Sciences (Switzerland)*, 13, 3 2023. ISSN 20763417. doi: 10.3390/app13053008.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL `https://proceedings.neurips.cc/paper_files/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf`.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015. URL `http://arxiv.org/abs/1503.02406`.

R. Urtasun, D. J. Fleet, and N. D. Lawrence. Modeling human locomotion with topologically constrained latent variable models. Workshop on Human Motion: Understanding, Modeling, Capture and Animation, 2007.

Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL `https://proceedings.neurips.cc/paper_files/paper/2005/file/ccd45007df44dd0f12098f486e7e8a0f-Paper.pdf`.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:283–298, 2 2008. ISSN 01628828. doi: 10.1109/TPAMI.2007.1167.

Yuting Ye and C. Karen Liu. Synthesis of responsive motion using a dynamic model. *Computer Graphics Forum*, 29(2):555–562, 2010.