Philipp Zech, Justus Piater (Eds.)
Proceedings of the Austrian Robotics Workshop 2018
 ISBN 978-3-903187-22-1, DOI 10.15203/3187-22-1

Estimating a Sparse Representation of Gaussian Processes Using Global Optimization and the Bayesian Information Criterion

Wilfried Wöber[1], Georg Novotny[1], Mohamed Aburaia[1], Richard Otrebski[1] and Wilfried Kubinger[1]

***Abstract*— Localization in mobile robotics is an active research area. Statistical tools such as Bayes filters are used for localization. The implementation of Gaussian processes in Bayes filters to estimate transition and measurement models were introduced recently. The non-linear and non-parametric nature of Gaussian processes leads to new possibilities in modelling systems. The high model complexity and computation expense based on the size of the dataset are shortcomings of Gaussian process Bayes filters. This work discusses our approach of a sparsing process of a dataset based on Bayesian information criterion model selection and global optimization. The developed approach combines the idea of avoiding model overfitting and Bayesian optimization to estimate a sparse representation of a Gaussian process. Based on visual odometry data of a mobile robot, the method was evaluated. The results show the operability of the system and unfold limitations of the current implementation such as random-initialization.**

I. INTRODUCTION

Bayes filters have been used frequently in mobile robotics. Different textbooks discuss the main aspects of different implementations of Bayes filters, namely Kalman filter or extended Kalman filter (EKF) [1]. Unfortunately, known restrictions limit the accuracy of Bayes filter implementations.

A Gaussian processes is a method for non-linear and non-parametric regression, which can be implemented in Bayes filters (EKF or particle filter) as a motion or measurement model [2], [3], [4]. The main benefit of a Gaussian process are estimations based on a dataset $\mathcal{D}$ including uncertainty. This leads to Bayes filter implementations, where prediction and correction are based on data [4] with minor model restrictions. The main shortcoming of Gaussian processes is the usage of the whole dataset for each estimation step. Therefore, the size of the dataset limits the processing speed.

This work tackles this problem by estimating pseudo-data for a sparse representation of a Gaussian process. This leads to the estimation of a new dataset $\mathcal{D}^*$, which consists of less data elements than the original dataset $\mathcal{D}$ without significant loss of model accuracy. This work is structured as follows: The next section discusses previous work. Section III discusses our method for optimization. Section IV evaluates our experiments. Finally, section V summarizes this work and gives an overview concerning future work.

II. PREVIOUS WORK

Bayes filters are well known methods for state estimation in mobile robotics [1, p. 23]. Doing so, $p(\vec{x}_t|\vec{x}_{1:t-1}, \vec{z}_{1:t}, \vec{u}_{1:t-1})$ must be evaluated using different approximations for motion models $p(\vec{x}_t|\vec{u}_t, \vec{x}_{t-1})$ as well as measurement models $p(\vec{z}_t|\vec{x}_t)$. This can be done using linear Gaussians in case of Kalman filter, or taylor approximation in case of EKF. To overcome approximation problems, non-parametric regression can be used to estimate models based on data. Based on that, models can be described using real system behavior. A method for such tasks is Gaussian process regression. This model is fully described using a mean and a covariance function [4], [5]:

$$GP_{\vec{\mu},\mathcal{D}}(\vec{x}_{new}) = \vec{k}^T \left[\mathbf{K} + \sigma_n^2 \mathbf{I}\right]^{-1} \vec{y} \quad (1)$$

$$GP_{\Sigma,\mathcal{D}}(\vec{x}_{new}) = k(\vec{x}_{new}, \vec{x}_{new}) - \vec{k}^T \left[\mathbf{K} + \sigma_n^2 \mathbf{I}\right]^{-1} \vec{k} \quad (2)$$

Where $GP_{\vec{\mu},\mathcal{D}}(.)$ predicts the output (mean) based on the input $\vec{x}_{new}$, the dataset $\mathcal{D}$, a kernel vector $\vec{k}$, a kernel matrix $\mathbf{K}$, the identity matrix $\mathbf{I}$ and the measurement noise σ_n^2. $GP_{\Sigma,\mathcal{D}}(.)$ predicts the inherent uncertainty using the additional scalar value $k(.)$, the kernel function. Note, that a detailed description of Gaussian processes and kernel methods can be found in [6].

The Gaussian process is based on the dataset $\mathcal{D} = \{(\vec{x}_0, y_0), ..., (\vec{x}_n, y_n)\}$, where $\vec{x} \in \mathbb{R}^{p\times 1}$ and $\vec{y} = (y_1, ..., y_n)^T$ and thus $\vec{y} \in \mathbb{R}^{n\times 1}$. Due to n examples in $\mathcal{D}$, $\mathbf{K} \in \mathbb{R}^{n\times n}$ and $\vec{k} \in \mathbb{R}^{n\times 1}$. Based on the dimensions of the Gaussian process parameters $\mathbf{K}$, $\vec{k}$ and $\vec{y}$, the size of the dataset $\mathcal{D}$ itself is critical facing real time constraints.

Gaussian process sparsing focuses on the generation of $\mathcal{D}^* = \{(\vec{x}_0^*, y_0^*), ..., (\vec{x}_m^*, y_m^*)\}$, where m is the number of examples in the new dataset $\mathcal{D}^*$ and

$$m \ll n \quad (3)$$

$$GP_{\vec{\mu},\mathcal{D}}(.) \approx GP_{\vec{\mu},\mathcal{D}^*}(.) \quad (4)$$

$$GP_{\Sigma,\mathcal{D}}(.) \approx GP_{\Sigma,\mathcal{D}^*}(.) \quad (5)$$

Recently, different approaches for Gaussian process sparsing and their applications have been discussed. In [7] a greedy sample selection is performed, where likelihood approximation is done. The subset is selected analysing the information gain. A stop criterion must be defined in terms of fixed set size or square error value. [8] generates new data points (pseudo points) to estimate $\mathcal{D}^*$ based on [7] and a maximum likelihood approach. [9] and [10] use a sparsed Gaussian process based on [8] to estimate stochastic differential equations.

Different to the previous work, the estimation of the sparse representation of a Gaussian process in this work is calculated based on the Bayesian information criterion (BIC) for pseudo input generation and global optimization for Gaussian process hyperparameter optimization.

[1] Department of Advanced Engineering Technologies, University of Applied Science Technikum Wien, Vienna, Austria, {woeber, novotny, aburaia, otrebski, kubinger}@technikum-wien.at

III. OUR APPROACH

The developed approach combines the idea of preventing model overfitting and global optimization in two stages. In the model selection stage, the sparsing of the dataset $\mathcal{D}$ using clustering and model selection is done. After that, the optimization stage optimizes a new Gaussian process to accomplish the constraints in equation 3 - 5. The remaining part of this section introduces the two stages.

A. Model Selection

The idea of sparsing in this work is based on avoiding overfitting of model selection. In this case, a finite gaussian mixture model (fGMM) was chosen to model the data. The optimal model dimension can be estimated using model selection based on the BIC [11] and a fGMM analysing $1, 2, ..., n$ mixture components. Our approach estimates the number of components using the BIC and estimates $\mathcal{D}^*$ using the expectation maximisation (EM) algorithm based fGMM fitting [12]. This is achieved using

$$p(\vec{x}|\vec{\theta}^*) = \sum_{k=1}^{m} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k) \tag{6}$$

$$\text{where } m = \underset{j=1:n}{\operatorname{argmin}}(\text{BIC}_{\text{fGMM}}(\mathcal{D}, j)) \tag{7}$$

Where $p(\vec{x}|\vec{\theta}^*)$ describes $\mathcal{D}^*$ using a fGMM. π_j, $\vec{\mu}_j$ and Σ_j are the parameters of the j-th fGMM component, which are summarized in $\vec{\theta}^*$. m is the optimized number of pseudo-inputs based on the BIC analysis. Typically, the number of relevant samples will be smaller than the raw dataset ($m \ll n$). Note, that this assumption is based on a high number of samples. $p(\vec{x}|\vec{\theta}^*)$ is estimated using the EM algorithm. Shortcomings of this approach are discussed in chapter IV.

$BIC_{\text{fGMM}}(.)$ uses the original dataset $\mathcal{D}$ and the number of mixing components to calculate a BIC trend. This function is defined using the log-likelihood at the maximum likelihood estimation, the number of used mixture components, the sample size and the number of estimated parameters [12]. Analysing n mixing components using the BIC, the optimal model can be chosen using the minimum BIC_{fGMM} value. The sparsing is done using the mean values $\vec{\mu}_{1:m}$ of the optimized fGMM. Due to that, the sparsed dataset is $\mathcal{D}^* = \{\vec{\mu}_1, ..., \vec{\mu}_m\}$. The vectors $\vec{\mu}_{1:m}$ are called pseudo-inputs.

Note, that the discussed sparsing process tackles the optimization of the mean function of Gaussian processes. As a result of the BIC based dataset sparsing, the estimation functions are going to change. To overcome this problem, the Gaussian process hyperparameters need to be adapted. This procedure is discussed in the remaining part of this section.

B. Gaussian Process Hyperparamter Optimization

After dataset sparsing, the new Dataset $\mathcal{D}^*$ affects the mean and variance function (see equations 1 and 2). To minimize the difference between the original and sparsed Gaussian process, global Bayesian optimization was used to adapt the hyperparameters. Hyperparameter optimization is critical because of high computational effort. Simultaneously, optimization is necessary for algorithm performance. Bayesian optimization [13], [14], [15], [16] tackles this problem by reformulating the optimization to a regression problem.

Doing so, a Gaussian process again is used for this regression formulation. The main idea of Bayesian optimization is step-wise optimization based on an initialized regression model using initial samples of the optimization function. Based on those samples and a regression model, functions like the expected improvement [14], [15] evaluates the expectation and uncertainty of the regression model. The expected improvement a_{EI} is defined as [15]:

$$a_{\text{EI}}(\vec{x}|\mathcal{D}^*) = \mathbb{E}\left[\max(f^* - f(\vec{x}), 0)\right] \tag{8}$$

Where f^* is the current maximum value of the regression model and $\mathbb{E}$ is the expectation value. The function $f(.)$ returns the regression value of the regression model. Note, that different implementations extend the idea of expected improvement to control exploitation and exploration [17]. Sequential optimization is done adding an evaluation of the model to optimize at the highest a_{EI} value. In this work, we use the r^2 of the variance for model comparison. The hyperparameters of the Gaussian process are optimized in terms of optimizing the r^2.

IV. EXPERIMENTAL RESULTS

Our experiments based on measurements on a mobile robot called "Robotino"[18]. The dataset $\mathcal{D}$ is based on visual odometry calculations of five experiments. We extracted the velocity (v_x) and transition (Δx) based on those measurements. Because this paper discusses the Gaussian process sparsing, our experiments discuss the movement model sparsing in detail. Note, that the used movement model is trivial. From a machine learning perspective, the model could be represented using linear regression. Even though the model itself is simple, the Gaussian process adds uncertainty estimation, which is needed for Bayes filters.

For the analysis of our approach, we simplified the data using gathered movement information of the mobile robot. The Gaussian process based transition model was used to predict the movement of the mobile robot Δ_x along the X-axis at time t based on the velocity v_x. Additional, the implementation of our method includes data pre-processing. The data pre-processing was done using outlier elimination and data normalization. Based on our BIC based pseudo-input generation, outlier detection is critical. The used implementation uses the expectation maximization algorithm to estimate the model [12]. Due to that, implemented random cluster initialization can result in unwanted sample elimination. This would make the evaluation of $GP_{\vec{\mu},\mathcal{D}}(.)$ and $GP_{\vec{\mu},\mathcal{D}^*}(.)$ respectively $GP_{\Sigma,\mathcal{D}}(.)$ and $GP_{\Sigma,\mathcal{D}^*}(.)$ impossible.

For outlier detection, hierarchical clustering was used [19]. The software implementation is based on the hierarchical clustering functions of [20] based on euclidean distances. The visualization of the outlier detection is shown in figure 1. The algorithm classifies 26 data elements out of 4458 data elements as outliers. For further discussion, the resulting normalized 4432 data elements describe $\mathcal{D}$. The Gaussian

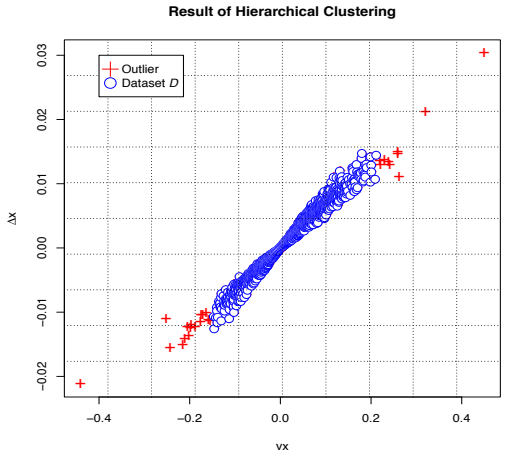

Fig. 1. Visualisation of hierarchical clustering for outlier detection

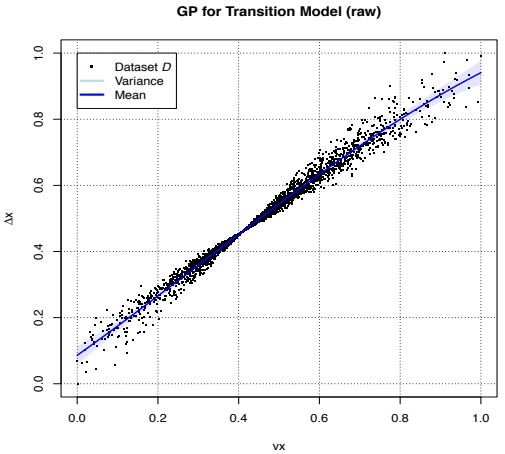

Fig. 2. Gaussian process without outliers. Note, that the data is normalized.

process based on $\mathcal{D}$ is shown in figure 2. The model sparsing was done analysing 20 to 500 pseudo-inputs using a stepsize of 50. The BIC based model selection is shown in figure 3. Note, that the implementation uses a BIC approximation which leads to a maximization instead of minimization [12]. The result of the BIC model selection is a fGMM using 170 pseudo-inputs. Those pseudo-inputs represents the dataset $\mathcal{D}^*$. Note the compression of the dataset to 170 datapoints.

Our experiments showed, that the random initialization of the fGMM clustering is critical for further optimization. The random initialization can result in a dataset $\mathcal{D}^*$, where areas with low frequency disappear. This leads to poor results of the sparsed Gaussian process. Currently, we can overcome this problem by increasing the number of datapoints in $\mathcal{D}^*$. A non-random initialization of the BIC based model selection is part of our recent research. Further, the penalty term in the function BIC_{fGMM} can be adapted for this application. The kernel used in this paper is the so-called 'rbf' kernel [4]. The hyperparameters of the kernel are the signal noise variance σ_n^2 and the smoothness factor ω [4], [6].

The behavior of the variance function is based on the hyperparameters of the Gaussian process, namely σ_n^2 and ω. Those hyperparameters were optimized using Bayesian optimization [17]. The results of the optimization are visualized in table I. The hyperparameters are optimized in 20 steps. The optimum is found at $r^2 = 0.9625$. Further, the r^2 of the Gaussian process mean values (raw and sparsed) using the optimized hyperparameters is 0.9998. Note, that due to the random initialization of the optimization algorithm, the optimization results differ. The analysis of 100 optimization procedures proves, that the exploitation/exploration tradeoff is not optimized yet and current part of further optimization. Further, due to processing limitations, 20 optimization steps and five initialization steps were used. A histogram of 100 optimization steps analysing the r^2 of $GP_{\Sigma,\mathcal{D}}(.)$ and $GP_{\Sigma,\mathcal{D}^*}(.)$ is shown in figure 4.

V. SUMMARY & OUTLOOK

We introduced a novel procedure for Gaussian process sparsing. The sparsing procedure is based on Bayesian information criterion model selection followed by hyperparameter optimization.

The model selection uses finite Gaussian mixture models to find pseudo-inputs, which represent a sparsed dataset $\mathcal{D}^*$. The hyperparameters are optimized using Bayesian optimization and focus on model difference minimization.

Our results proves that the method is applicable. Limitation, namely random initialization of model selection and optimization, are discussed. Those limitations are currently part of ongoing research. This research focuses on non-random algorithm initialization and BIC calculation adaption. Based on the results of our optimized approach, Gaussian process

TABLE I
THE OPTIMIZATION PROCEDURE IN THIS EXAMPLE.

#	σ_n^2	ω	r^2	#	σ_n^2	ω	r^2
1	3.3619	0.0171	0.7824	2	4.8541	0.0174	0.7306
3	4.0077	0.0043	0.7020	4	2.4200	0.0143	0.8156
5	0.0922	0.0086	0.9401	6	0.0050	0.0199	0.7952
7	0.0050	0.0010	0.8036	8	4.7598	0.0087	0.7048
9	0.0050	0.0121	0.7955	10	0.1486	0.0092	**0.9625**
11	0.3107	0.0041	0.9387	12	0.5753	0.0081	0.9251
13	0.4405	0.0196	0.9488	14	0.4870	0.0140	0.9407
15	0.8008	0.0190	0.9243	16	0.9876	0.0013	0.8275
17	0.3334	0.0081	0.9451	18	1.6964	0.0087	0.8340
19	2.8853	0.0093	0.7781	20	0.6128	0.0199	0.9378

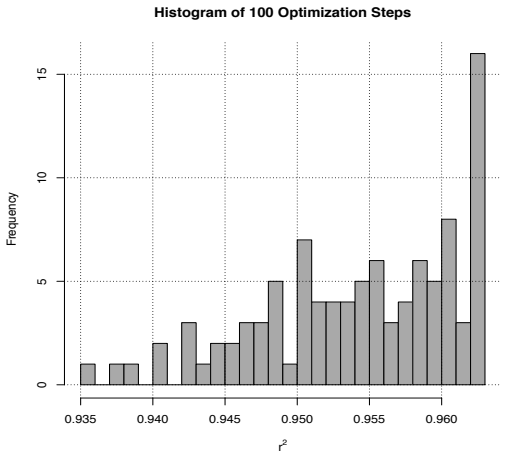

Fig. 4. Histogram of 100 optimization procedures (r^2 of $GP_{\Sigma,\mathcal{D}}(.)$ and $GP_{\Sigma,\mathcal{D}^*}(.)$).

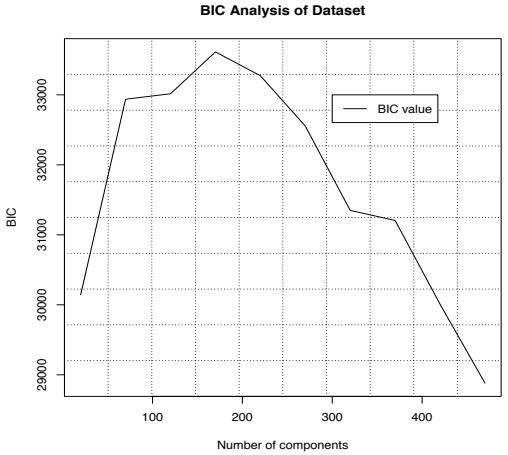

Fig. 3. Result of (approximated) BIC analysis of the transition model [12].

optimization approaches can be applied without the need of processing clouds. Currently, mobile robot localization algorithms based on sparsed Gaussian processes are implemented. This task includes the analysis of the processing workload.

Further, the expected improvement can be used to estimate the "completeness" of motion models as a preceding analysis step.

The next steps include the merging of the sparsing and optimization steps to a single optimization task. Based on the planned method extensions, non-trivial Gaussian process sparsing will be analysed. This will be used in further research areas such as example generation in object recognition.

REFERENCES

[1] Thrun, S.; Burgard, W.; Fox, D., *Probabilistic Robotics*. Massachusetts Institute of Technology: MIT Press, 2006.

[2] Hartikainen, J.; Srkk, S., "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2010.

[3] Reece, S.; Roberts, S., "An introduction to Gaussian processes for the Kalman filter expert," in *2010 13th Conference on Information Fusion (FUSION)*. IEEE.

[4] Ko, J.; Fox, D., "GP-BayesFilters: Bayesian Filtering Using Gaussian Process Prediction and Observation Models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008.* Nice, France: IEEE, 2008.

[5] ——, "GP-BayesFilters: Bayesian filtering using Gaussian Process prediction and observation models," 2009, (online) https://rse-lab.cs.washington.edu/papers/gp-bayesfilter-arj-09.pdf (Last access: 18.2.2018).

[6] Bishop, C.M., *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.

[7] Seeger, M.; Williams, C.K.I.; Lawrence, N.D., "Fast forward selection to speed up sparse Gaussian process regression." Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003.

[8] Snelson, E.; Ghahramani, Z., "Sparse Gaussian Processes using Pseudo-inputs." Advances in Neural Information Processing Systems 18 (NIPS 2005), 2005.

[9] Garcia, C.A.; Otero, A.; Felix, P.; Presedo, J.; Marquez, D.G., "Non-parametric Estimation of Stochastic Differential Equations with Sparse Gaussian Processes." Physical Review E, 2017.

[10] Archambeau, C.; Cornford, D.; Opper, M.; Shawe-Taylor, J., "Gaussian Process Approximations of Stochastic Differential Equations." JMLR: Workshop and Conference Proceedings, 2007.

[11] Schwarz, G., "Estimating the dimension of a model." Annals of Statistics, 1978.

[12] Scrucca L., Fop M., Murphy T.B., Raftery A.E., "mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models." The R Journal, 2016, pp. 289–317.

[13] Osborne, M.A., Garnett, R., Roberts, S.J., "Gaussian processes for global optimization." 3rd International Conference on Learning and Intelligent Optimization (LION3), 2009.

[14] Bergstra, J.; Bardenet, R.; Benigo, Y.; Kegl, B., "Algorithms for hyper-parameter optimization." NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems, 2011.

[15] Klein, A.; Falkner, S.; Bartels, S.; Henning, P.; Hutter, F., "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets." Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[16] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959. [Online]. Available: http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

[17] Y. Yan, *rBayesianOptimization: Bayesian Optimization of Hyperparameters*, 2016, r package version 1.1.0. [Online]. Available: https://CRAN.R-project.org/package=rBayesianOptimization

[18] Festo. (2018) Robotino. [Online]. Available: http://www.festo-didactic.com/int-en/services/robotino/

[19] Liang, B., "A Hierarchical Clustering Based Global Outlier Detection Method." 5th International Conference on Bio-Inspired Computing: Theories and Applications, 2010.

[20] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: https://www.R-project.org/