Joint Learning of Sparse Gaussian Processes and Gaussian Process Latent Variable Models for Semi-supervised Tasks

Ana Alice Ximenes Mota Peres [1] **, César Lincoln Cavalcante Mattos** [1]

[1]Department of Computer Science, Federal University of Ceará, Brazil

aliceximenes@alu.ufc.br, cesarlincoln@dc.ufc.br

Abstract. *The speed and variety of collected data have increased in a surprising way, corroborating with the creation of large and diverse datasets. When using such data for training supervised machine learning models, it is necessary to annotate the available samples. However, labeling instances can be challenging, expensive, and time-consuming. In this context, semi-supervised learning models have been extensively researched over the past decades. Among the supervised learning methods, models based on Gaussian Processes (GPs) offer the advantage of quantifying uncertainties and providing significant modeling flexibility. Nevertheless, like several learning strategies, they cannot be directly applied to semi-supervised scenarios. To overcome this issue, the current work proposes a GP-based approach to perform semi-supervised learning. The proposal consists in simultaneously training an unsupervised GP latent variable model (GPLVM) and a supervised sparse GP model. The approach leverages both labeled and unlabeled data to create a more effective final classifier. Additionally, a neural network is included to reproduce the latent variables learned by the GPLVM, which enables scaling and eases its use with unseen data. The introduced solution is evaluated on public datasets and compared with standard semi-supervised approaches from the literature, in both inductive and transductive settings. The experiments indicate that the proposed technique, despite some mixing results, is competitive, especially in transductive learning problems.*

1. Introduction

The Artificial Intelligence field has introduced various approaches to modeling thinking and reasoning. Among them, Machine Learning stands out by building models that extract knowledge from observed data, rather than relying on queries or manual inspections. This work focuses on semi-supervised learning, a branch of machine learning widely studied in the literature [Chapelle et al. 2006a], and seen as an innovative solution where supervised and unsupervised learning face limitations.

On one hand, labeling data can be hard, expensive, and time consuming. Not having enough labeled data can hinder the use of supervised machine learning models, worsening their performance. On the other hand, unsupervised methods do not leverage the available labeled instances. The semi-supervised approach allows the learning model to take advantage of both a small labeled dataset and a large unlabeled one, with the goal of improving the model pattern recognition capability [Prakash and Nithya 2014].

Standard supervised machine learning methods usually focus on the functional or conditional dependency between a target variable and the corresponding input variables.

The estimation is calculated based on a training dataset that reflects this dependency. After that, the trained model is evaluated with data that was not observed at the training process. This is known as *inductive learning*, where the model generalizes the learned patterns from the training set to perform predictions for the new data [Le et al. 2006].

Alternatively, methods based on *transductive learning* focus on predicting over samples inside a specific dataset, without the need to generalize the patterns for the whole data domain. It uses the whole dataset, both labeled and unlabeled, without a train-test split, in the learning process. Both inductive and transductive approaches can be considered in semi-supervised learning.

A promising perspective for semi-supervised tasks is the use of probabilistic models. In this case, Gaussian Processes (GP) models stands out due to its modeling flexibility and ability to quantify prediction uncertainty [Williams and Rasmussen 2006]. Standard GP models are widely used in supervised settings, while sparse GPs (SGPs) extend its application to larger datasets and non-Gaussian likelihood settings [Titsias 2009, Hensman et al. 2013]. Moreover, a variation named GP Latent Variable Model (GPLVM) enables the use of GPs in unsupervised learning scenarios [Lawrence 2003, Titsias and Lawrence 2010]. The GPLVM learns a latent representation of observed data without explicit labels, crucial for dimensionality reduction. While GP is valued for accurate prediction with labeled data, GPLVM stands out for revealing hidden patterns in unlabeled datasets. However, it cannot represent unseen data without extra optimization, limiting its use.

Motivated by the high cost and effort of data labeling and the abundance of unlabeled data, this work proposes a semi-supervised learning approach that fully leverages all available data. The main contributions are: (i) a novel GP-based method combining sparse GPs and GPLVM; (ii) a joint learning strategy for supervised and unsupervised GP components; (iii) a neural network that learns the latent projection from the trained GPLVM, allowing generalization to new data. The approach is experimentally evaluated and compared with standard methods in both inductive and transductive settings.

2. Related Work

There is a variety of semi-supervised approaches in the literature. One that is widespread and simple is called Self-training [Scudder 1965]. In it, a supervised model is trained with the labeled data, then the model predicts pseudo-labels for the unlabeled data, and, finally, the model is re-trained with the basis of more reliable pseudo-labels, repeating the process iteratively. Co-training [Blum and Mitchell 1998] is a method derived from Self-training, in which the data is divided into two sets with different attributes and it is assumed that each group provides different and complementary information about the observations. The approach starts with training a model on each set using the labeled data. Then, the most confident predictions are used to iteratively build more labeled data. This algorithm and its variants have been successful in Natural Language Processing, among other fields [Van Engelen and Hoos 2020]. A common way to represent labeled and unlabeled data is using graphs. Examples of techniques that pursue this direction are Label Propagation (LP) and Label Spreading (LS) [Zhu and Ghahramani 2002, Zhou et al. 2003], which has several variations [Fujiwara and Irie 2014].

An alternative approach is the use of Bayesian methods, which focus on posterior

distribution modeling. However, their common assumption is that the posteriori is not affected by unlabeled data [Chapelle et al. 2006b, Li et al. 2008]. The contribution of a not available label to the likelihood is obtained by marginalizing the corresponding variable. However, it is resolved as a constant, providing no new information. Therefore, naive use of Bayesian models for semi-supervised learning may be challenging.

One of its first uses of GPs for semi-supervised learning was proposed in [Lawrence and Jordan 2004] for a classification problem with labeled and unlabeled data. The Null-Category Noise Model (NCNM) excludes unlabeled data via a null region, using the cluster assumption without specialized kernels. [Rogers and Girolami 2007] generalize NCNM for multiclass problems with a multinomial probit GP.

In the field of transductive learning, the work in [Gärtner et al. 2005] compare the use of GPs on both inductive and transduction for multi-class classification. The authors in [Le et al. 2006] expand this analysis for regression problems, finding more competitive results. The work in [Srijith et al. 2013] focus on ordinal regression, pointing out that transduction offers significant advantages when used on smaller labeled datasets.

Inspired by [Kingma et al. 2014], the work in [Damianou and Lawrence 2015] builds a framework that uses GPLVM on the whole data (with or without label) and then applies the learned embedding as input to any other supervised model (e.g., logistic regression). Our work follows a similar strategy, but the proposed methodology aims to jointly train a GPLVM and a sparse GP model for semi-supervised problems. The joint training is expected to enable information sharing between both components along the single optimization procedure, as it will be detailed later.

3. Background

Gaussian Process (GP). A GP is a flexible Bayesian nonparametric method usually applied for supervised learning [Williams and Rasmussen 2006]. They are a class of probabilistic models that learn a distribution over functions, being entirely defined by its mean and covariance functions.

To model a function $f(\cdot)$ means to represent the relation between its inputs and outputs. Considering N observations, let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be the input data and $\boldsymbol{y} \in \mathbb{R}^N$ be the output data. Then, the value of an output y_i based on its input $\boldsymbol{x}_i$ and a random Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \boldsymbol{I})$ is given by $y_i = f(\boldsymbol{x}_i) + \epsilon_n$.

The function values $\boldsymbol{f} = f(\boldsymbol{X})$ can be approximated using a multivariate Gaussian, so $\boldsymbol{f} \sim \mathcal{N}(m(\boldsymbol{X}), \boldsymbol{K})$, where $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, $K_{ij} = k(x_i, x_j)$, being the covariance matrix obtained by the covariance (or kernel) function $k(\cdot, \cdot)$, and $m(\boldsymbol{X}) = \mathbb{E}[f(\boldsymbol{X})]$ the mean function, which is commonly chosen to be $m(\boldsymbol{X}) = \mathbf{0}$ a priori.

Since we have considered a Gaussian likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_y^2 \boldsymbol{I})$, due to the Gaussian noise assumption, the marginal likelihood is also Gaussian: $p(\boldsymbol{y}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \boldsymbol{K} + \sigma_y^2 \boldsymbol{I})$. Importantly, the posterior is also Gaussian: $p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{K}(\boldsymbol{K} + \sigma_y^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \boldsymbol{K} - \boldsymbol{K}(\boldsymbol{K} + \sigma_y^2 \boldsymbol{I})^{-1}\boldsymbol{K})$. The latter can be used to perform predictions, while the former is used to optimize the kernel hyperparameters following analytical gradients [Williams and Rasmussen 2006].

Gaussian Process Classifiers. Both regression and classification can be seen as function approximation tasks. However, GP expressions in the classification setting

are not analytical. In the binary classification case, the output is a discrete variable $y_i \in \{0, 1\}$ that follows a Bernoulli distribution. In the multiclass case, $y_i \in \{1, ..., C\}$ follows a Categorical distribution. In both cases, the model prediction (probability of a class) is related to the latent function values by a non-Gaussian likelihood function. There are several strategies to perform approximated inference in those scenarios, such as Laplace approximation [Williams and Barber 1998], Expectation Propagation (EP) [Minka 2001], Markov Chain Monte Carlo (MCMC) [Neal 1999], and Variational Inference (VI) [Wainwright et al. 2008]. In this work, we pursue VI-based methods.

Sparse Gaussian Process (SGP). There has been great progress in scaling GPs for large datasets. The key point that allowed advancement was the use of induction points [Snelson and Ghahramani 2005, Titsias 2009, Hensman et al. 2013]. The work in [Titsias 2009] proposes to consider M induction variables $\boldsymbol{u} \in \mathbb{R}^M$ with the same GP prior of the vector $\boldsymbol{f}$. Each one of these variables are related to M inducing inputs (or *pseudo-inputs*), $\boldsymbol{\xi}_j|_{j=1}^M \in \mathbb{R}^D$ that are on the same domain of the input $\boldsymbol{x}_i|_{i=1}^N$.

The work in [Hensman et al. 2013] introduced the use of stochastic variational inference (SVI) in the context of GPs. The authors introduce an evidence lower bound (ELBO) which can be factorized along the observations, enabling mini-batch optimization. Let the variational distribution for the inducing points be $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{S})$, with $\boldsymbol{m} \in \mathbb{R}^M$ and $\boldsymbol{S} \in \mathbb{R}^{M \times M}$, the ELBO obtained in [Hensman et al. 2013] is given by

$$\mathcal{L} = \sum_{i=1} \mathcal{L}_i - \mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u})), \tag{1}$$

$$\text{where } \mathcal{L}_i = \log \mathcal{N}(y_i|\boldsymbol{k}_i^T \boldsymbol{K}_u^{-1} \boldsymbol{m}, \sigma_y^2) - \frac{1}{2\sigma_y^2}([\boldsymbol{K}_f]_{ii} - \boldsymbol{k}_i^T \boldsymbol{K}_u^{-1} \boldsymbol{k}_i) - \frac{1}{2}\mathrm{Tr}\left(\frac{1}{\sigma_y^2} \boldsymbol{S} \boldsymbol{K}_u^{-1} \boldsymbol{k}_i \boldsymbol{k}_i^T\right),$$

and $\boldsymbol{k}_i^T = [k(\boldsymbol{x}_i, \boldsymbol{\xi}_1), \cdots, k(\boldsymbol{x}_i, \boldsymbol{\xi}_M))]$, $\mathrm{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence an $\mathrm{Tr}(\cdot)$ is the trace operator. All the global variational parameters and the kernel hyperparameters can be optimized using scalable stochastic gradient methods and mini-batches.

Gaussian Process Latent Variable Model (GPLVM). Given an unsupervised setting, where we only have the observations $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, the GPLVM assumes that the observed data was generated by the transformation of a latent variables set $\boldsymbol{\Lambda} \in \mathbb{R}^{N \times W}$ by a GP, where we usually choose $W < D$. The straight mapping $(\boldsymbol{\Lambda} \rightarrow \boldsymbol{X})$ is defined by GPs independently defined on each dimension of $\boldsymbol{X}$. The model focuses on keeping distant in the latent space the observations that are distant in the observed space [Lawrence and Quinonero-Candela 2006]. Thus, the GPLVM aims to build a W-dimensional latent space for the D-dimensional observed data [Quirion et al. 2008]. It is a technique that can be considered as a GP regression model with multiple outputs where only the output data is available [Titsias and Lawrence 2010].

The work in [Titsias and Lawrence 2010] proposed an improved version, entitled Bayesian GPLVM, which after defining a prior for the latent variables $\boldsymbol{\Lambda}$, finds an approximate posterior given the available data. The authors follow a variational approach and uses a factorized Gaussian as the approximate posterior of the latent variables. Similar to the SGP, all the variational parameters are optimized by the means of a tractable ELBO.

4. Proposed GP Approach for Semi-Supervised Learning

The demand for semi-supervised methods is on constant growth as the volume of unlabeled data continues to increase. The combination of these methods with probabilistic approaches results in a more flexible and adaptable model, which is essential for solving real-world problems. To that end, this work adopts a hybrid approach that combines elements from supervised and unsupervised learning methods. The goal is to exploit the complementarity between labeled and unlabeled data to improve the learning capability of the resulting model.

4.1. Learning projections and mappings

Our main contribution consists of jointly training unsupervised and supervised probabilistic models (GPLVM and SGP, respectively) using all the available data, labeled or not. First, the D-dimensional data $\boldsymbol{X}$ with N samples is masked and divided into a labeled set $\boldsymbol{X}^L$ and an unlabeled set $\boldsymbol{X}^U$. Both $\boldsymbol{X}^L$ and $\boldsymbol{X}^U$ are used to learn a latent space $\boldsymbol{\Lambda}$ using a Bayesian GPLVM to obtain an approximate posterior distribution $q(\boldsymbol{\Lambda}) = \prod_i^N q(\boldsymbol{\lambda}_i)$, where $q(\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\mu}_\Lambda, \mathrm{diag}(\boldsymbol{\sigma}_\Lambda))$ and $\mathrm{diag}(\cdot)$ builds a diagonal matrix from a vector. The variational parameters $\boldsymbol{\mu}_\Lambda$ and $\boldsymbol{\sigma}_\Lambda$ related to each instance are optimized during training.

The approximate posterior characterizes the learned latent space, while also considering the uncertainty inherent to the task. Thus, inspired by the work in [Damianou and Lawrence 2015], to leverage the knowledge obtained from the GPLVM, we sample from the latent space in the regions which are related to the labeled data $\boldsymbol{X}^L$. More specifically, for a given $\boldsymbol{x}$ we sample from the corresponding $\boldsymbol{\lambda} \sim \mathcal{N}(\boldsymbol{\mu}_\Lambda, \boldsymbol{\sigma}_\Lambda)$. The samples taken considering all the available labeled data are associated to the same corresponding labels and collected in the matrix $\boldsymbol{Z}$, which is fed as the input of the SGP model. In summary, the labeled examples are projected in the latent space, learned from both labeled and unlabeled data, before being fed to the supervised SGP model, which learns the final mapping to the observed output $\boldsymbol{y}$. It is worth noting that, due to the variational variances $\boldsymbol{\sigma}_\Lambda$, we are able to generate multiple samples for each labeled data, which constitutes a form of data augmentation.

However, the use of GPLVM brings a limitation to the solution, since the model can only find the projection on the latent space for the data used during the training phase. Given new inputs $\boldsymbol{X'}$, where $\boldsymbol{X'} \cap \boldsymbol{X} = \emptyset$, the GPLVM will not be able to find a projection without additional optimization steps, limiting its use to perform new predictions.

To overcome the above limitation, we follow an approach similar to the work in [Dai et al. 2016, Mattos and Barreto 2019, Lalchand et al. 2022], where a neural network is used to learn the projection performed by the GPLVM. In our case, a Multi-Layer Perceptron (MLP) is trained using $\boldsymbol{x}_i|_{i=1}^N$ as input and the corresponding $\boldsymbol{\mu}_\Lambda$ as the output. Since the SGP model is fed with $\boldsymbol{Z}$, which comes from random samples taken from the learned distribution $q(\boldsymbol{\Lambda})$, the expected value for each latent projection is given by $\boldsymbol{\mu}_\Lambda$, which turns it in a sensible output for training the network. The trained MLP allows the projection of unseen data to the latent space learned by the GPLVM, which enables generalization and significantly amplifies its applicability. A detailed overview of the entire methodology is presented in Fig. 1.

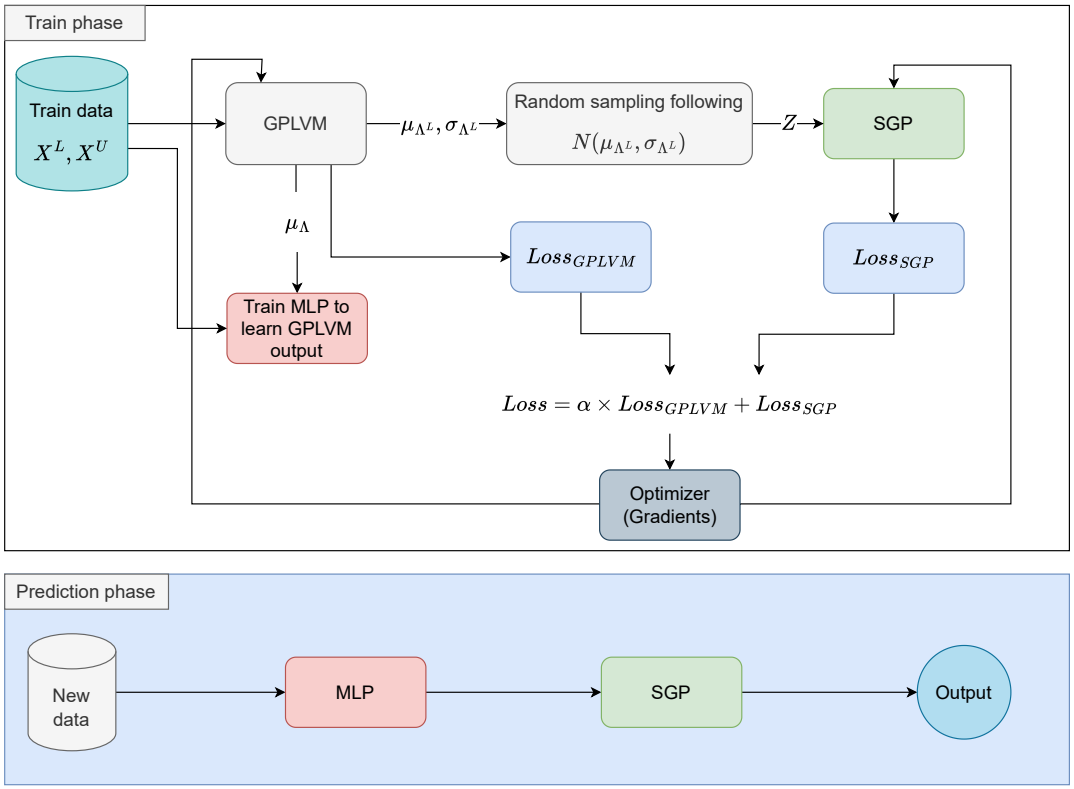


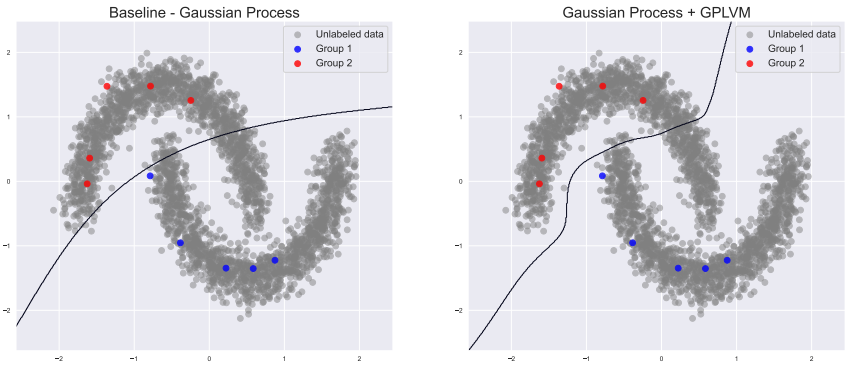
Figure 1. Workflow for the proposed semi-supervised methodology.

4.2. Joint training of SGP and GPLVM

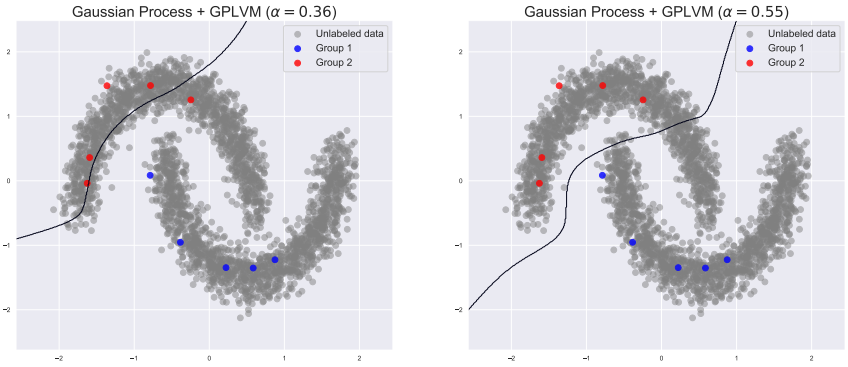
It is worth emphasizing that GPLVM and SGP serve distinct purposes: SGP is supervised (regression or classification), while GPLVM is unsupervised (or regression with latent inputs), each with its own loss function. Unlike [Damianou and Lawrence 2015], we do not train the models separately. As SGP performance depends on the latent space learned by GPLVM, we propose joint training using the gradients obtained by the sum of both losses. This enables information sharing during training. Since GPLVM uses more data (labeled and unlabeled), its loss is scaled by a factor $\alpha > 0$ to balance with the SGP loss, a common strategy in problems involving multiple objective functions.

Another thing to consider is that, at the beginning of the training procedure, the GPLVM may not have a good initial latent representation, since it is initialized with a simple prior. We noticed empirically that it is worth training only the GPLVM for the first few epochs. Thus, we introduce the variable $\mathrm{gpFreeze}$ to control for how many epochs the training is restricted to the GPLVM.

Once the joint training of the GPLVM and the SGP is concluded, we train the MLP network to learn how to project the original D-dimensional input data into the W-dimensional latent space learned by the GPLVM. Predictions for unseen data can then be performed by projecting the new input with the MLP and using the SGP to obtain the final prediction, as depicted at the bottom of Fig. 1.



(a) Decision boundaries obtained by the single SGP (left) and the proposed solution (right). The unlabeled data is colored in grey.



(b) Decision boundaries obtained by the proposed solution with different α values to scale de GPLVM loss. The distinct results highlight the importance of optimizing the α hyperparameter.

5. Experiments

As follows we perform some initial experiments with synthetic data, to better understand the behavior of the proposed solution. Then, we run additional benchmarks with real-world data and compare the proposal with methods from the literature.

5.1. Initial investigation

We first consider a synthetic dataset named MOONS, inspired by [Sindhwani et al. 2007], with two classes and two-dimensional inputs, as shown in Fig. 2a. Gray points are unlabeled, and colored points are labeled samples from both classes. Fig. 2a also shows the decision boundaries learned by a standard SGP (left) and by the joint training of GPLVM and SGP (right). The joint model's boundary (right) is influenced by unlabeled data, staying further from the red class, despite limited labeled data on its left side.

We observed that results were sensitive to the hyperparameter α. Fig. 2b shows boundaries for different α values. Using $\alpha = 0.55$ (right) yields a boundary similar to Fig. 2a with $\alpha = 0.43$, while decreasing it to 0.36 worsens the result.

5.2. Real-world benchmarks - inductive learning

To evaluate the proposed solution over real-world data, we have chosen four public datasets: cancer ($N = 569, D = 30$, 2 classes) [Street et al. 1993], vehicle ($N = 840, D = 18$, 4 classes) [Zouhal and Denoeux 1998], oil ($N = 1000, D = 12$, 3 classes) [Bishop and James 1993], mocap ($N = 78000, D = 36$, 5 classes) [Dua and Graff 2017].

Each dataset was randomly split into train (70%) and test (30%). Then, the train set was further divided into labeled $\boldsymbol{X}^L$ and unlabeled $\boldsymbol{X}^U$. The model hyperparameters were tuned using the Random Search Cross Validation (RSCV) strategy, with a 3-fold strategy in the inner loop. Given the semi-supervised setup, only small percentages of labeled data is used in each training scenario. For each dataset, we consider 4 different labeled data regimes. The hyperparameters optimized via RSCV were the learning rate (lr) for the GP models, the number of epochs, the mini-batch size, the value for $\mathrm{gpFreeze}$ and the GPLVM loss scaling factor α.

The experiments trained the GPLVM and SGP simultaneously using mini-batches. Each dataset had different labeled/unlabeled splits to compute the mean and standard deviation of metrics. For each, we show the mean and a 2 standard deviation confidence margin. To assess efficiency with varying labeled data, models used 3 training set sizes. Small labeled sets were chosen to test performance in challenging scenarios where supervised learning struggles and semi-supervised learning stands out.

The MLP was trained to predict the GPLVM projection using MSE loss, with hyperparameters tuned via RSCV. Adam optimizer was used. We optimized mini-batch size, epochs, learning rate, and architecture. The network has 1 to 3 layers with 32, 64, or 128 hidden units. The best configuration is selected in RSCV using the highest F1-score from K-fold CV. For multi-class tasks, the weighted average F1-score is considered.

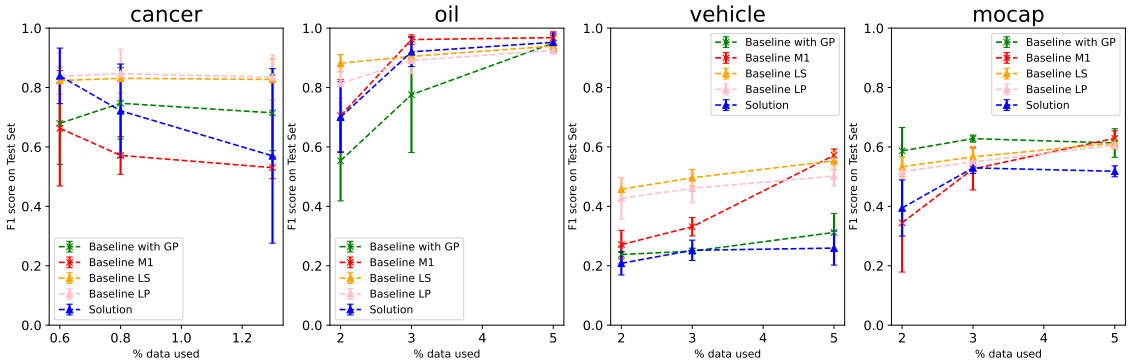
For the sake of comparison, we also evaluate other models from the literature: a standard SGP model with a Radial Basis Function (RBF) kernel trained only with the labeled data; the M1 strategy from [Kingma et al. 2014], which first train an unsupervised model (here, the GPLVM) with all the data and then trains a classifier (here, the SGP) on the learned projections for the labeled data; the semi-supervised Label Propagation (LP) [Zhu and Ghahramani 2002] and Label Spreading (LS) models [Zhou et al. 2003]. Since the M1 method also uses a GPLVM, it would not be possible to run it on unseen test data. Therefore, we also train a MLP to predict the projections learned by the GPLVM.

Fig. 3a presents the F1-score computed over the test sets for all models and datasets, with different percentages of training labeled data. Overall, the LS and LP methods achieved the best results, but not for all cases. For the cancer dataset, using only 0.6% of labeled data, our solution had a similar value, and it had a better performance using 3% and 5% of labeled data in the oil dataset. It is important to highlight that LS and LP are not probabilistic models. Thus, since we focus on probabilistic approaches and their application to semi-supervised tasks, the main competitors are the SGP and M1 methods.

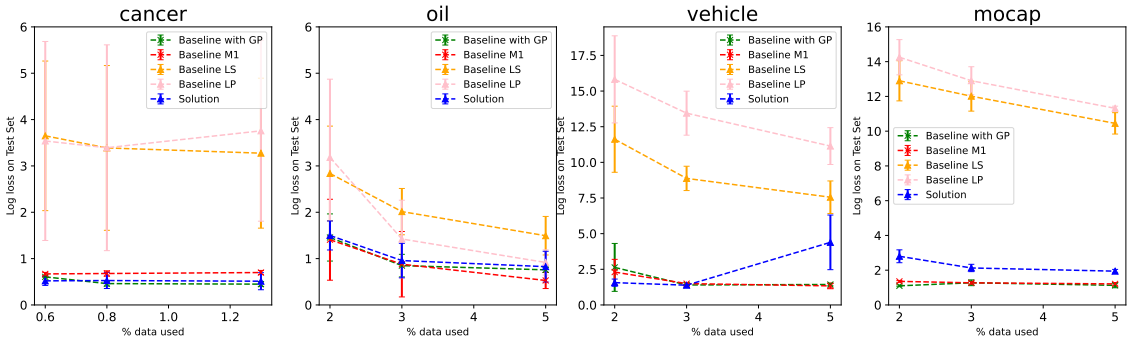
There is no best model for all datasets. On the cancer dataset, our solution has a better F1-score using the small amount of labeled data. The M1 baseline outperforms the other models in the vehicle dataset on all amounts of labeled data. However, it is worth mentioning that it can only perform predictions for the test data due to the proposed approach of using a neural network to learn the GPLVM projection. Fig. 3b shows the log-loss, where lower is better. In this case, non-probabilistic models (LP and LS) had the worst results in all experiments.

5.3. Real-world benchmarks - transductive learning

Inspired by the work in [Sindhwani et al. 2007], we also perform experiments in the transductive scenario. In this case, we did not divide the datasets into train and test. Instead, we



(a) Induction: Comparison of F1-score values (higher values are better) obtained for different percentages of labeled data.



(b) Induction: Comparison of log-loss values (lower values are better) obtained for different percentages of labeled data.

create $\boldsymbol{X}^L$ and $\boldsymbol{X}^U$ from $\boldsymbol{X}$ and calculate the metrics using the labels from $\boldsymbol{X}^U$ (which are not used for training). Fig. 4a shows the F1-scores computed for each model and dataset and Fig. 4b shows the obtained log-loss values. The transduction improved the results for the probabilistic approaches, mainly in the oil dataset.

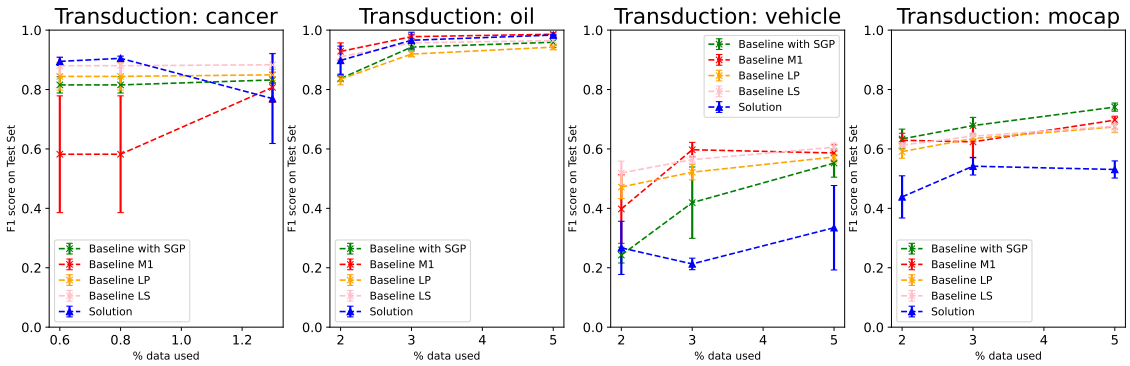
5.4. Discussion

Overall, it is possible to note that there is in fact no a single best model for all cases. Although LP and LS perform well in some scenarios, since they are not probabilistic models, they are less flexible in terms of handling uncertainty. Moreover, the proposed solution achieved a much better log-loss value than these methods, comparable to or better than the other probabilistic methods. The proposed solution achieved mixed results, worse results in some experiments, promising or competitive in others. It is worth mentioning that the M1 method stood out in several cases, which was only possible due to our introduction of an MLP network to enable projection of unseen data.

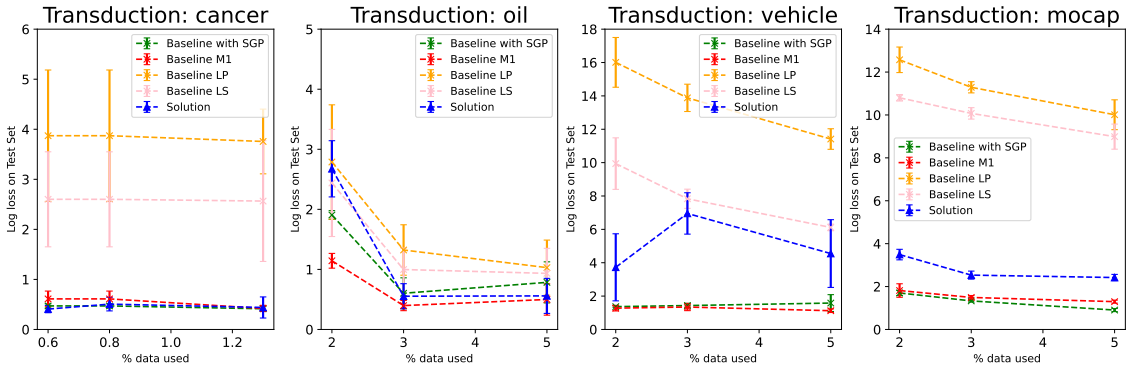
Comparing the results from the inductive and transductive experiments, it is possible to notice that in the transductive scenario the probabilistic models presented better overall results, which may indicate that the proposed solution may be more adequate for this learning setting.

6. Conclusion and Further Work

The present work explored the use of GP-based probabilistic models in the context of semi-supervised learning. We have proposed a new approach to jointly train a GPLVM,



(a) Transduction: Comparison of F1-score values (higher values are better) obtained for different percentages of labeled data.



(b) Transduction: Comparison of log-loss values (lower values are better) obtained for different percentages of labeled data.

with both labeled and unlabeled data, and a sparse GP, with labeled data. We also used an MLP network to enable the projection of unseen data in the latent space learned by the GPLVM. Experimentation in both inductive and transductive scenarios showed mixed performance for the proposed methodology, with promising results in some cases.

An interesting enhancement of this research is the joint training of the neural network along with the GP modules, which may enable even more change of information between the solution parts during training. This direction also opens up the possibility of fully amortizing the variational parameters of the GPLVM, similar to the work in [Lalchand et al. 2022], which would make the solution more scalable and capable of using more flexible variational approximations.

References

Bishop, C. M. and James, G. D. (1993). Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 327(2-3):580–593.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Learning Theory*, pages 92–100.

Chapelle, O., Schölkopf, B., and Zien, A. (2006a). Introduction to semi-supervised learning. In *Semi-Supervised Learning*, pages 1–12. The MIT Press.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006b). *Semi-Supervised Learning*. The MIT Press, Massachusetts, USA.

Dai, Z., Damianou, A. C., González, J., and Lawrence, N. D. (2016). Variational auto-encoded deep gaussian processes. In *Proceedings of the 4th ICLR, San Juan*.

Damianou, A. C. and Lawrence, N. D. (2015). Semi-described and semi-supervised learning with gaussian processes. In *Proceedings of the 31st UAI, Amsterdam, The Netherlands*, pages 228–237. AUAI Press.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fujiwara, Y. and Irie, G. (2014). Efficient label propagation. In *International conference on machine learning*, pages 784–792. PMLR.

Gärtner, T., Le, Q., Burton, S., Smola, A. J., and Vishwanathan, V. (2005). Large-scale multiclass transduction. *Advances in Neural Information Processing Systems*, 18.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27.

Lalchand, V., Ravuri, A., and Lawrence, N. D. (2022). Generalised gplvm with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, pages 7841–7864. PMLR.

Lawrence, N. (2003). Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16.

Lawrence, N. and Jordan, M. (2004). Semi-supervised learning via gaussian processes. *Advances in neural information processing systems*, 17.

Lawrence, N. D. and Quinonero-Candela, J. (2006). Local distance preservation in the gp-lvm through back constraints. In *Proceedings of the 23rd international conference on Machine learning*, pages 513–520.

Le, Q. V., Smola, A. J., Gärtner, T., and Altun, Y. (2006). Transductive gaussian process regression with automatic model selection. In *Proceedings of the 17th ECML, Berlin, Germany*, pages 306–317. Springer.

Li, H., Li, Y., and Lu, H. (2008). Semi-supervised learning with gaussian processes. In *2008 Chinese Conference on Pattern Recognition*, pages 1–5. IEEE.

Mattos, C. L. C. and Barreto, G. A. (2019). A stochastic variational framework for Recurrent Gaussian Processes models. *Neural Networks*, 112:54–72.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th UAI, Seattle, USA*, pages 362–369. Morgan Kaufmann.

Neal, R. (1999). Markov chain sampling using hamiltonian dynamics. In *Talk at the Joint Statistical Meetings, Baltimore, August*.

Prakash, V. J. and Nithya, D. L. (2014). A survey on semi-supervised learning techniques. *arXiv preprint arXiv:1402.4645*.

Quirion, S., Duchesne, C., Laurendeau, D., and Marchand, M. (2008). Comparing gplvm approaches for dimensionality reduction in character animation.

Rogers, S. and Girolami, M. (2007). Multi-class semi-supervised learning with the epsilon-truncated multinomial probit gaussian process. In *Gaussian Processes in Practice*, pages 17–32. PMLR.

Scudder, H. (1965). Adaptive communication receivers. *IEEE Transactions on Information Theory*, 11(2):167–174.

Sindhwani, V., Chu, W., and Keerthi, S. S. (2007). Semi-supervised gaussian process classifiers. In *IJCAI*, pages 1059–1064.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18.

Srijith, P., Shevade, S., and Sundararajan, S. (2013). Semi-supervised gaussian process ordinal regression. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 144–159. Springer.

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. SPIE.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR.

Titsias, M. and Lawrence, N. D. (2010). Bayesian gaussian process latent variable model. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 844–851. JMLR Workshop and Conference Proceedings.

Van Engelen, J. E. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, 109(2):373–440.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Williams, C. K. and Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1342–1351.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. (2003). Learning with local and global consistency. *Advances in neural information processing systems*, 16.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. *ProQuest number: information to all users*.

Zouhal, L. M. and Denoeux, T. (1998). An evidence-theoretic k-nn rule with parameter optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(2):263–271.