ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414960

ONLINE UNSUPERVISED LEARNING USING ENSEMBLE GAUSSIAN PROCESSES WITH RANDOM FEATURES

Georgios V. Karanikolas, Qin Lu, and Georgios B. Giannakis

Dept. of ECE and DTC, University of Minnesota, Minneapolis, MN

ABSTRACT

Gaussian process latent variable models (GPLVMs) are powerful, yet computationally heavy tools for nonlinear dimensionality reduction. Existing scalable variants utilize low-rank kernel matrix approximants that in essence subsample the embedding space. This work develops an efficient online approach based on random features by replacing spatial with spectral subsampling. The novel approach bypasses the need for optimizing over spatial samples, without sacrificing performance. Different from GPLVM, whose performance depends on the choice of the kernel, the proposed algorithm relies on an ensemble of kernels - what allows adaptation to a wide range of operating environments. It further allows for initial exploration of a richer function space, relative to methods adhering to a single fixed kernel, followed by sequential contraction of the search space as more data become available. Tests on benchmark datasets demonstrate the effectiveness of the proposed method.

Index Terms— Dimensionality reduction, Gaussian processes, ensemble learning, random features

1. INTRODUCTION

Dimensionality reduction (DR) refers to the task of obtaining meaningful low-dimensional vector representations (embeddings) of observations lying in a high-dimensional space. This key unsupervised learning task can aid with unveiling patterns in (unlabeled) data, or, serve as a feature extractor for downstream learning tasks [4].

GPLVM is a probabilistic approach to nonlinear dimensionality reduction, where the nonlinear mapping from latent embeddings to observations is modeled using Gaussian processes (GPs) [9]. GPLVM is perhaps the most widely used GP based approach in this context, with applications ranging from localization [5] to deconfounding time series from single cell assays [18]. Thanks to its probabilistic nature, it provides means for uncertainty quantification, a key advantage over methods such as kernel PCA.

Although quite powerful, the GPLVM yields the embeddings that maximize the likelihood of the observations. As evaluations of the likelihood and gradients thereof involve the inverse of the kernel matrix, the complexity scales cubically with the number of observations [9]. Scalability is achieved through the use of low-rank approximations to the kernel matrix. This is typically accomplished by relying on the covariance of a few (relative to the total number of observations) so termed inducing points [10, 3]. The locations of the inducing inputs are in turn jointly optimized alongside with the latent embeddings, thereby significantly increasing the number of optimization variables involved. Furthermore, it is well known that inducing point based schemes only approximate well the covariance of the (full) GP for points in the proximity of the inducing points [12]. Unfortunately, in online setups one cannot rely on obtaining the inducing point locations in the initialization phase. Rather, continuous optimization over the inducing point locations is required, so as to account for scenarios where different clusters of observations, and thus latent space regions, are being sequentially revealed. This in turn implies a recurring computational overhead.

To overcome these issues, we propose an online GPLVM scheme that relies on random (also known as spectral) features [17, 12]. Rather than subsampling the embedding space, the spectral samples are drawn prior to seeing any data, in a fashion dictated solely by the functional form of the kernel, and remain fixed as streaming observations become available.

An additional challenge GPLVM faces is that of kernel selection, as this critically affects the resultant embeddings. We will address this issue in an efficient fashion, using the notion of ensemble learning that involves multiple experts, each relying on a different kernel. As experts operate independently (except for the inexpensive fusion step) the computational burden of kernel exploration is fully parallelizable. As more data become available, the probabilities associated with each expert are updated accordingly. This allows for sequential refinement of the kernel choice.

Several variants of the GPLVM are available, including the back-constrained [11], variational GPLVM [3], and a few online schemes [25, 24, 16]. Regarding random features, this is to the best of our knowledge the first time they are being used in the context of GPLVMs. Random features have, however, been used in the realm of online kernel PCA; see e.g. [6]. With respect to ensemble learning, a GPLVM scheme which can be broadly categorized in this area is [24], where different from the proposed approach the goal is to track the latent

This work was supported in part by NSF grant 1901134.

state of a dynamical system. Ensemble methods are, nonetheless, more commonplace in the context of probabilistic PCA; see [22] for the seminal work and [1] for an online variant. In these approaches, however, each member of the ensemble only performs *linear* dimensionality reduction. The present paper is inspired by our recent work [14] that highlighted the benefits of combining random features (RFs) with ensembles of experts in the context of *supervised* learning; see also [21] for (supervised) learning of kernels.

Notation Subscripts in matrices indicate the (temporal) index of the last entry included, whereas $[\mathbf{x}]_d$ denotes the d-th entry of vector $\mathbf{x}$. Superscripts of the form $^{(s)}$ specify quantities related to expert s. Finally, $\mathcal{N}(x;\mu,\sigma^2)$ indicates the value of the probability density function of a Gaussian random variable with mean μ and variance σ^2, when evaluated at x.

2. RANDOM FEATURE BASED GPLVM

Consider the problem of dimensionality reduction. Let $\mathbf{y}_t \in \mathbb{R}^D$ denote the t-th observation and $\mathbf{x}_t \in \mathbb{R}^q$ be the corresponding (latent) low-dimensional representation ($q < D$). For the associated matrices we have $\mathbf{Y} := [\mathbf{y}_1 \dots \mathbf{y}_T]^\top \equiv [\mathbf{y}_{:1} \dots \mathbf{y}_{:D}]$, and $\mathbf{X} := [\mathbf{x}_1 \dots \mathbf{x}_T]^\top$, respectively.

The mapping from the latent representation $\mathbf{x}_t$ the to d-th entry (dimension) of the observation $\mathbf{y}_t$ is modeled as

$$y_{td} = f_d(\mathbf{x}_t) + \epsilon_{td} \tag{1}$$

where f_d is a nonlinear function, and $\{\epsilon_{td}\}$ are assumed to be drawn i.i.d. from $\mathcal{N}(0,\sigma_\epsilon^2)$. In the original GPLVM [9], a GP prior is placed on f_d, with a covariance structure determined by a kernel, let κ. Here we will consider a random feature based approximation for the kernel [17, 12]. In short, let $\bar{\kappa}(\tilde{\mathbf{x}}) = \bar{\kappa}(\mathbf{x}-\mathbf{x}')$ denote an appropriately normalized version of a stationary kernel $\kappa(\mathbf{x},\mathbf{x}') = \kappa(\mathbf{x}-\mathbf{x}')$. It can be shown that by drawing m i.i.d. random vectors $\mathbf{v}_i \sim \pi_\kappa(\mathbf{v}) = \mathcal{F}(\bar{\kappa}(\tilde{\mathbf{x}}))$, where $\mathcal{F}$ denotes the Fourier transform, and letting

$$\phi(\mathbf{x}) = \frac{1}{\sqrt{m}}[\cos(\mathbf{v}_1^\top\mathbf{x})\ \sin(\mathbf{v}_1^\top\mathbf{x}) \dots \cos(\mathbf{v}_m^\top\mathbf{x})\ \sin(\mathbf{v}_m^\top\mathbf{x})]^\top$$

a kernel approximation can be obtained as $\breve{\kappa}(\mathbf{x},\mathbf{x}') = \phi^\top(\mathbf{x})\phi(\mathbf{x}')$ [12]. Finally, $\phi(\mathbf{x})$ is referred to as the random or spectral feature vector corresponding to $\mathbf{x}$. Consequently, a parametric form for f_d is available. In particular, we have $\breve{f}_d(\mathbf{x}) = \mathbf{w}_d^\top\phi(\mathbf{x})$, where $\mathbf{w}_d$ is a vector of regression coefficients[1]. The corresponding likelihood is given by

$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{t=1}^{T}\prod_{d=1}^{D}\mathcal{N}(y_{td};\mathbf{w}_d^\top\phi(\mathbf{x}_t),\sigma_\epsilon^2) \tag{2}$$

[1]The choice of m impacts the approximation accuracy for the kernel matrix as well as computational complexity; see also [17].

where $\mathbf{W} := [\mathbf{w}_1 \dots \mathbf{w}_D]^\top \in \mathbb{R}^{D\times 2m}$. Similar to the dual form of probabilistic PCA [9, 23], we impose a prior on the regression weights

$$p(\mathbf{W}) = \prod_{d=1}^{D}\mathcal{N}(\mathbf{w}_d;\mathbf{0},\mathbf{I})$$

where $\mathbf{I}$ denotes the identity matrix. Marginalizing $p(\mathbf{Y},\mathbf{W}|\mathbf{X})$ over $\mathbf{W}$ yields

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D}\mathcal{N}(\mathbf{y}_{:d};\mathbf{0},\mathbf{\Phi}^\top\mathbf{\Phi}+\sigma_\epsilon^2\mathbf{I}) \tag{3}$$

where $\mathbf{\Phi} := [\phi(\mathbf{x}_1)\dots\phi(\mathbf{x}_T)]$.

Remark. Through the RF approximation, per dimension evaluations of the likelihood and gradients thereof can be performed in $\mathcal{O}(Tm^2)$ operations, compared to $\mathcal{O}(T^3)$ when the exact kernel matrix is used. This can be seen by using the matrix inversion lemma in (3). Leveraging RFs reflects a departure from the inducing-points based approximations used in GPLVM for reducing computational complexity [9, 10].

Having established the form of the conditional likelihood of the observations given their latent representations, we can now turn our attention to attaining the latter. In the GPLVM paradigm $\mathbf{X}$ is obtained by minimizing the negative log-likelihood of the observations $\mathbf{Y}$ [9]. Further imposing a prior on $\mathbf{X}$, we have

$$\mathbf{X} = \arg\min_{\boldsymbol{\mathcal{X}}} -\log p(\mathbf{Y}|\boldsymbol{\mathcal{X}}) - \log p(\boldsymbol{\mathcal{X}}). \tag{4}$$

Notice that this choice corresponds to seeking the maximum-a-posteriori (MAP) estimate of $\mathbf{X}$. With regards to the prior, we will assume hereafter that $p(\mathbf{X}) = \prod_{t=1}^{T}\mathcal{N}(\mathbf{x}_t;\mathbf{0},\sigma_x^2\mathbf{I})$. Following a standard choice in the literature, (4) is solved using the (scaled) conjugate gradient method [15]. Alternatives include e.g. stochastic gradient descent with momentum [25]. The latent representations are initialized through a probabilistic PCA (PPCA) embedding. Finally, note that generally the cost function in (4) is jointly optimized over latent representations and hyperparameters, such as kernel parameters and variance σ_ϵ^2; we have omitted the latter for exposition purposes.

3. ONLINE RF-BASED GPLVM

Our discussion so far involved obtaining an RF-based counterpart to the GPLVM. As a first step towards an online algorithm consider an out-of-sample observation $\mathbf{y}_*$, and the corresponding latent representation $\mathbf{x}_*$. Using identities for Gaussian conditionals, the associated likelihood turns out to be given by $p(\mathbf{y}_*|\mathbf{Y},\mathbf{X},\mathbf{x}_*) = \mathcal{N}(\mathbf{y}_*;\boldsymbol{\mu}_*,\sigma_*^2\mathbf{I})$ where

$$[\boldsymbol{\mu}_*]_d = \phi^\top(\mathbf{x}_*)\mathbf{A}^{-1}\mathbf{\Phi}\mathbf{y}_{:d} \tag{5a}$$

$$\sigma_*^2 = \sigma_\epsilon^2 + \sigma_\epsilon^2\phi^\top(\mathbf{x}^*)\mathbf{A}^{-1}\phi(\mathbf{x}^*) \tag{5b}$$

Algorithm 1 Online RF-based GPLVM

Draw vectors $\{\mathbf{v}_i\}_{i=1}^m \sim \pi_\kappa(\mathbf{v})$

Embed $\mathbf{Y}_{t_0} \to \mathbf{X}_{t_0}$ and obtain hyperparameters (cf. (4))

$\mathbf{\Phi}_{t_0} = [\phi(\mathbf{x}_1) \dots \phi(\mathbf{x}_{t_0})]$

$\mathbf{B}_{t_0} = \mathbf{\Phi}_{t_0}\mathbf{Y}_{t_0}$

$\mathbf{R}_{t_0} = \texttt{CholeskyFactor}(\mathbf{\Phi}_{t_0}\mathbf{\Phi}_{t_0}^\top + \sigma_\epsilon^2\mathbf{I})$

for $t = t_0+1, t_0+2, \dots$ **do**

Receive datum $\mathbf{y}_t$

▷ Embed $\mathbf{y}_t \to \mathbf{x}_t$

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \; -\log p(\mathbf{y}_t|\mathbf{Y}_{t-1}, \mathbf{X}_{t-1}, \mathbf{x}) - \log p(\mathbf{x})$$

$$\boldsymbol{\phi}_t = \frac{1}{\sqrt{m}}[\cos(\mathbf{v}_1^\top\mathbf{x}_t)\;\; \sin(\mathbf{v}_1^\top\mathbf{x}_t) \dots \cos(\mathbf{v}_m^\top\mathbf{x}_t)\;\; \sin(\mathbf{v}_m^\top\mathbf{x}_t)]^\top$$

▷ Update $\mathbf{B},\ \mathbf{R}$

$$\mathbf{B}_t = \mathbf{B}_{t-1} + \boldsymbol{\phi}_t\mathbf{y}_t^\top$$

$$\mathbf{R}_t = \texttt{CholeskyUpdate}(\mathbf{R}_{t-1}, \boldsymbol{\phi}_t)$$

end for

with $\mathbf{A} := \mathbf{\Phi}\mathbf{\Phi}^\top + \sigma_\epsilon^2\mathbf{I}$. The initially unknown embedding $\mathbf{x}_*$ is obtained as the MAP estimate, by solving

$$\mathbf{x}_* = \arg\min_{\mathbf{x}} \; -\log p(\mathbf{y}_*|\mathbf{Y}, \mathbf{X}, \mathbf{x}) - \log p(\mathbf{x}). \quad (6)$$

In practice, (6) is initialized at the embedding corresponding to the point in $\mathbf{Y}$ that is the nearest neighbor of $\mathbf{y}_*$. Observe now that the key quantities in (5) can be updated in a recursive fashion. In particular, letting $\mathbf{B}_T = \mathbf{\Phi}_T\mathbf{Y}_T$ we have

$$\mathbf{B}_{T+1} = \mathbf{B}_T + \phi(\mathbf{x}_{T+1})\mathbf{y}_{T+1}^\top.$$

Furthermore, $\mathbf{A}$ also admits rank-one updates as $\mathbf{A}_{T+1} = \mathbf{A}_T + \phi(\mathbf{x}_{T+1})\phi^\top(\mathbf{x}_{T+1})$. Since computations involve $\mathbf{A}^{-1}$, however (cf. (5)), it is more convenient to perform the updates on the corresponding Cholesky factor (CF), let $\mathbf{R}$, where $\mathbf{A} = \mathbf{R}^\top\mathbf{R}$ [8]; see also [7].

Summarizing, for each incoming sample, an instance of (6) is solved yielding the associated latent representation, followed up by rank-one updates on $\mathbf{R}$ and $\mathbf{B}$. The above procedure is summarized in Alg. 1, where t_0 denotes the number of samples used in the batch initialization phase, `CholeskyFactor` computes the CF of its argument and `Cholesky Update` performs a rank-one CF update. Finally, note that upon receiving a datum, Alg. 1 yields the corresponding low-dimensional representation; additionally, one could update the embeddings of e.g. all of the k-nearest neighbors (kNNs) of the aforementioned datum.

4. ENSEMBLE ONLINE RF-BASED GPLVM

Obtaining a high quality embedding through Alg. 1 heavily relies upon the kernel κ being appropriately selected. Due to the online nature of the problem, however, it is hard to make a suitable choice a priori. Intuitively, a more realistic route is to start from a highly expressive model class, that is one employing several kernels, and refine the model, in terms of kernel selection, as more data become available.

Towards this end, we will now consider an extension of Alg. 1 inspired by the ensemble learning paradigm. As a starting point, consider an ensemble (set) of S experts, where each member, let s, is associated with a kernel κ_s and the corresponding RF mapping $\phi^{(s)}(\cdot)$. More specifically, each expert runs an instance of Alg. 1, albeit with a different kernel, and our short-term goal is to choose the "best" embedding across experts, for each incoming sample.

Formally, let $\mathbf{x}_t^{(s)}$ denote the embedding of sample $\mathbf{y}_t$ according to expert s. With the corresponding likelihoods $\{p(\mathbf{y}_t|\mathbf{Y}_{t-1}, s, \mathbf{X}_{t-1}^{(s)}, \mathbf{x}_t^{(s)})\}_{s=1}^S$ at hand (cf. (5)), the MAP solution[2] for the "best" embedding across experts, let $\mathbf{x}_t^{(s^*)}$, is given by

$$(\mathbf{x}_t^{(s^*)}, s^*) := \arg\max_{\mathbf{x},s} p(\mathbf{y}_t|\mathbf{Y}_{t-1}, s, \mathbf{X}_{t-1}^{(s)}, \mathbf{x})\; p(s|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)\; p(\mathbf{x})$$

where $p(s|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)$ denotes the posterior probability of the model associated with expert s, given the observations $\mathbf{y}_1, \dots, \mathbf{y}_{t-1}$. Notice that the main computational burden in the above optimization problem (i.e., solving subproblems of the form (6)) is decoupled across experts and hence fully parallelizable. The posterior probabilities, which can also be viewed as expert weights, are in turn updated in a recursive fashion. Dropping common terms, we have

$$p(s|\mathbf{Y}_t, \{\mathbf{X}_t^{(\sigma)}\}_{\sigma=1}^S) \propto p(s|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)\; p(\mathbf{y}_t|\mathbf{Y}_{t-1}, s, \mathbf{X}_{t-1}^{(s)}, \mathbf{x}_t^{(s)})$$

for $s = 1, \dots, S$. Intuitively, in the long run, we expect the probability mass to concentrate at the experts whose embeddings best describe the observed data. In other words, our scheme is effectively performing online kernel selection, adapting to the data as they become available.

[2]

$$\arg\max_{\mathbf{x}_t,s} p(\mathbf{x}_t, s|\mathbf{y}_t, \mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)$$

$$\equiv \arg\max_{\mathbf{x}_t,s} p(\mathbf{y}_t, \mathbf{x}_t, s|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)$$

$$\equiv \arg\max_{\mathbf{x}_t,s} p(\mathbf{y}_t|\mathbf{Y}_{t-1}, s, \mathbf{X}_{t-1}^{(s)}, \mathbf{x}_t)\; p(s|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\sigma)}\}_{\sigma=1}^S)\; p(\mathbf{x}_t)$$

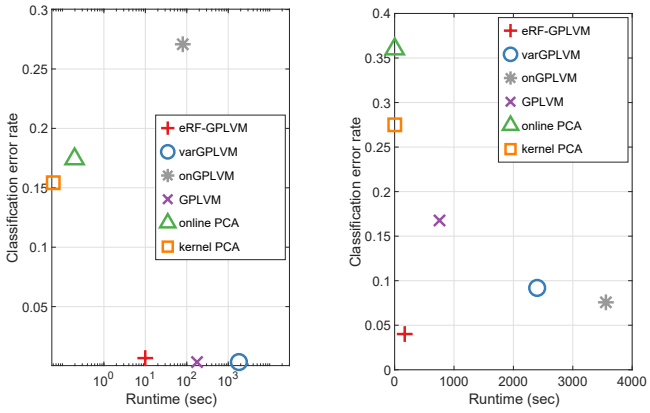

Fig. 1: Classification error versus runtime plots for the `oil` (a), and `USPS` (b) datasets.

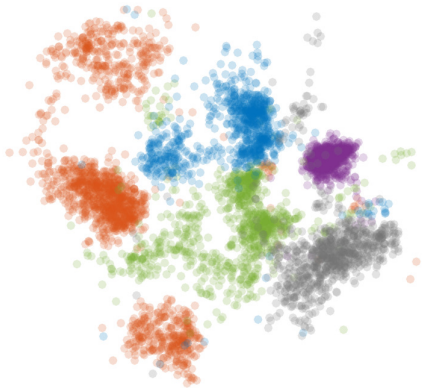

Fig. 2: Visualization of the embedding attained by the proposed approach on the `USPS` dataset. Colors represent different digits.

5. NUMERICAL TESTS

In order to assess the performance of the proposed approach, tests were performed on two benchmark datasets, namely `oil` flow data ($D = 12$) [2], and the `USPS` handwritten digits set ($D = 256$). The latter was limited to digits $0-4$ for ease in their two-dimensional visualization. These datasets were also used in the original GPLVM paper [9].

Several competing alternatives were considered. GPLVM based methods comprise the original GPLVM [9, 10], a variational inference based scheme (`varGPLVM`) [3], as well as an online GPLVM variant (`onGPLVM`); see Alg. 2 in [25]. PCA based alternatives encompass online PCA [19, 13], and (batch) kernel PCA [20].

The embedding dimensionality was set to $q = 2$, and the results presented correspond to the median across 11 trials. Regarding the proposed scheme, $2m = 100$ random features were used, each expert relied on a radial basis function (RBF) (also known as Gaussian) kernel with variance taken from the set $\{2^k\}_{k=-3}^{3}$, t_0 was set to 10% of the number of samples, and (4) was additionally optimized over σ_ϵ^2. For the GPLVM based methods, the RBF kernel was used, 100 inducing points were utilized, and the maximum number of iterations was set to $1,000$. Initializations were provided by means of PPCA embeddings. Finally, for kernel PCA (kPCA), a grid search was performed over RBF kernels with variances in $\{2^k\}_{k=-10}^{10}$ and the lowest error rate achieved is reported. Note that for kPCA the reported runtime does not include the time required for the grid search.

The error rate of the nearest neighbor classification rule was used as the performance metric, when applied to the resultant embeddings; see e.g. [9]. The results are summarized in Fig. 1, in the form of error rate versus runtime plots. In the `oil` dataset the proposed `eRF-GPLVM` scheme achieves similar error rate (less than 1%) to `GPLVM` and `varGPLVM`, while being more than one and two orders of magnitude faster, respectively. In the `USPS` set, our approach achieves the lowest overall error rate. The only schemes that achieve error rates in the same order of magnitude, namely `onGPLVM` and `varGPLVM` have runtimes that are 14 and 21 times higher, respectively. Finally, in both experiments, PCA based schemes, although computationally efficient, yield high error rates. Due to space limitations, a visualization of the embedding attained is only provided for the proposed approach on the `USPS` dataset (Fig. 2). We can observe that good separation between clusters of different digits is achieved, in line with the low classification error rate in Fig. 1.

6. CONCLUSIONS

The present work put forth a novel online approach to nonlinear dimensionality reduction. Unlike existing GPLVM schemes, scalability is accomplished through the use of random features, a choice that allows for circumventing the need for recurring optimization over inducing points. Furthermore, the ensemble learning scheme our algorithm is endowed with, allows for parallelized kernel exploration. The result is a highly effective and scalable GPLVM scheme that achieves runtimes that are typically one or more orders of magnitude lower than the competing GPLVM alternatives on benchmark datasets.

7. REFERENCES

[1] A. Bellas, C. Bouveyron, M. Cottrell, and J. Lacaille, "Model-based clustering of high-dimensional data streams with online mixture of probabilistic PCA," *Advances in Data Analysis and Classification*, vol. 7, no. 3, pp. 281–300, 2013.

[2] C. M. Bishop and G. D. James, "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*, vol. 327, no. 2-3, pp. 580–593, 1993.

[3] A. C. Damianou, M. K. Titsias, and N. D. Lawrence, "Variational inference for latent variables and uncertain inputs in Gaussian processes," *The Journal of Mchine Learning Research*, vol. 17, no. 1, pp. 1425–1486, 2016.

[4] G. Feng, J. G. Quirk, and P. M. Djurić, "Supervised and unsupervised learning of fetal heart rate tracings with deep Gaussian processes," in *14th Symposium on Neural Networks and Applications (NEUREL)*, 2018, pp. 1–6.

[5] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using Gaussian process latent variable models." in *IJCAI*, vol. 7, no. 1, 2007, pp. 2480–2485.

[6] M. Ghashami, D. J. Perry, and J. Phillips, "Streaming kernel principal component analysis," in *Proc. of Artificial Intelligence and Statistics*, 2016, pp. 1365–1374.

[7] A. Gijsberts and G. Metta, "Real-time model learning using incremental sparse spectrum Gaussian process regression," *Neural Networks*, vol. 41, pp. 59–69, 2013.

[8] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 2013.

[9] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Journal of machine learning research*, vol. 6, no. Nov, pp. 1783–1816, 2005.

[10] N. D. Lawrence, "Learning for larger datasets with the Gaussian process latent variable model," in *Artificial intelligence and statistics*, 2007, pp. 243–250.

[11] N. D. Lawrence and J. Quiñonero-Candela, "Local distance preservation in the gp-lvm through back constraints," in *Proc. of Intl. Conf. on Machine Learning*, 2006, pp. 513–520.

[12] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *The Journal of Machine Learning Research*, vol. 11, pp. 1865–1881, 2010.

[13] A. Levey and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371–1374, 2000.

[14] Q. Lu, G. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian processes with spectral features for online interactive learning with scalability," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1910–1920.

[15] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[16] X. Qin, P. Blomstedt, and S. Kaski, "Scalable Bayesian non-linear matrix completion," *arXiv preprint arXiv:1908.01009*, 2019.

[17] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, 2008, pp. 1177–1184.

[18] J. E. Reid and L. Wernisch, "Pseudotime estimation: deconfounding single cell time series," *Bioinformatics*, vol. 32, no. 19, pp. 2973–2980, 2016.

[19] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[20] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[21] A. Sinha and J. C. Duchi, "Learning kernels with random features." in *NIPS*, 2016, pp. 1298–1306.

[22] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[23] ——, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[24] Y. Wang, M. A. Brubaker, B. Chaib-draa, and R. Urtasun, "Bayesian filtering with online Gaussian process latent variable models." in *UAI*, 2014, pp. 849–857.

[25] A. Yao, J. Gall, L. V. Gool, and R. Urtasun, "Learning probabilistic non-linear latent variable models for tracking complex activities," in *Advances in Neural Information Processing Systems*, 2011, pp. 1359–1367.