A Survey and Theoretical Analysis of Gaussian Process Latent Variable Models

Oliver Liu
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
zhuoranl@andrew.cmu.edu

Darby Losey
Program in Neural Computation
Department of Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
dlosey@andrew.cmu.edu

1 Introduction

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It is a simplistic, yet fundamental dimensionality reduction technique ubiquitously used for data visualization and reasoning. Efforts have been made to integrate PCA with other more fine-grained unsupervised methods under a coherent framework. One line of such endeavors is to interpret PCA as a special case of Gaussian process latent variable models (GPLVMs). This specific interpretation highlights the flexible nature of GPLVMs. Specifically, GPLVMs connect the observed data to a underlying latent variable space with the assumption that this transformation function has a Gaussian process prior (Lawrence, 2005). This modeling framework has the advantage that it is flexible, nonlinear and the transformation between latent and observation space is straightforward (Gao *et al.* , 2011a).

The additional usage of different kernel based methods have lead to various improvements to the initial algorithm. Changing posterior distributions has also lead to a large diversity of method implementations. However, the rich family of GPLVMs lack organization, and the more refined algorithms present significant computational challenges. This report is an attempt to coalesce this diverse field. In doing so, we highlight the formulation and key theoretical properties for the various GPLVM classes, as well as present practical algorithms to improve computational efficiency for large datasets. Additionally, we provide context for GPLVM by contrasting it to other forms of dimensionality reduction, such as PCA.

2 Notation

1. $\mathbf{I}_D$: the identity matrix in $\mathbb{R}^{D\times D}$.
2. $\mathbf{X}_I^{(J)}$: the features in set J over observations in the set I.
3. $\mathbf{K}_{I,J}$: the kernel matrix with rows at set I and columns at set J
4. $\mathcal{N}(X|\mu,\sigma)$: equivalent to $X \sim \mathcal{N}(\mu,\sigma)$

3 Assumptions, and Problem Formulation

We assume all observation $\mathbf{X}_i$'s are i.i.d., unless otherwise noted. We briefly introduce two major components of GPLVMs, latent variable models and Gaussian processes, as well as necessary assumptions in constructing our models.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

3.1 Latent Variable Models

A latent variable model is a statistical model that relates a set of observable variables $\mathbf{X} \in \mathbb{R}^{n\times D}$ to a set of latent variables $\mathbf{Z} \in \mathbb{R}^{n\times q}$ through a set of parameters (Loehlin, 1998). The relationship between the latent variable and the data point is linear with noise added,

$$\mathbf{x} = \mathbf{f}(\mathbf{z}) + \epsilon,$$

where $\mathbf{W} \in \mathbb{R}^{D\times q}$ and we assume $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$. The likelihood for a data point can then be written as

$$p(\mathbf{x}|\mathbf{z}, \mathbf{W}, \sigma) \sim \mathcal{N}(\mathbf{W}\mathbf{z}, \sigma^2 \mathbf{I}_D).$$

Then, subject to our desired interpretations, the marginal likelihoods can be formulated as

$$p(\mathbf{x}|\mathbf{W}, \sigma) = \int_{\mathcal{Z}} p(\mathbf{x}|\mathbf{z}, \mathbf{W}, \sigma) p(\mathbf{z}) d\mathbf{z} \tag{1a}$$

$$p(\mathbf{x}|\mathbf{z}, \sigma) = \int_{\mathcal{W}} p(\mathbf{x}|\mathbf{z}, \mathbf{W}, \sigma) p(\mathbf{W}) d\mathbf{W} \tag{1b}$$

Their log-likelihoods over the full data set are in turn thus given by

$$\log(p(\mathbf{X}|\mathbf{W}, \sigma)) = \sum_{i=1}^{N} \log(p(\mathbf{x}_i|\mathbf{W}, \sigma)) \tag{2a}$$

$$\log(p(\mathbf{X}|\mathbf{Z}, \sigma)) = \sum_{j=1}^{D} \log(p(\mathbf{x}^{(j)}|\mathbf{Z}, \sigma)) \tag{2b}$$

We observe that 1a and 1b require distribution assumptions for $\mathbf{z}$ and $\mathbf{W}$ respectively.

3.2 Gaussian Processes

Gaussian processes are a class of stochastic process such that every finite collection of its belonging random variables has a multivariate normal distribution. The mean and variance of such a Gaussian process must be functions of the space on which the process operates. Typically, the mean function is taken to be zero, while the covariance function is necessarily constrained to produce positive semi-definite matrices.

3.3 Gaussian process latent variable models

The GPLVMs describe a new class of models which consist of Gaussian process mappings from a latent space, $\mathbf{Z}$, to an observed data-space, $\mathbf{X}$. Starting with GPLVMs' original motivation as an interpretation of probabilistic principal component analysis (PPCA), we survey their potent as a general framework in admitting non-linear dimensionality reduction techniques, as well as other supervised and unsupervised learning methods.

4 Key Results

4.1 Interpreting PPCA as a GPLVM

We first Assume the relationship between the latent variable and output variable to be linear:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \epsilon,$$

We also assume the latent variable has a priori distribution

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_q)$$

The marginal likelihood for each observation can be found analytically by 1a as

$$p(\mathbf{x}|\mathbf{W}, \sigma) \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}_D)$$

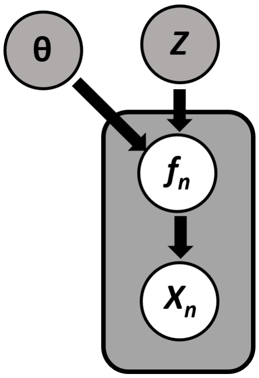

Figure 1: Latent variable $\mathbf{Z}$ dictates observed variable $\mathbf{X}_n$ by transformation function $\mathbf{f}_n$ where $\mathbf{f}_n$ has a Gaussian process prior parameterized by θ.

Then, the standard approach is to maximize the data likelihood 2a with respect to parameters $\mathbf{W}, \sigma$. However, under appropriate conjugate prior over $\mathbf{W}$ such as the spherical Gaussian distribution:

$$p(\mathbf{W}) = \prod_{j=1}^{D} \mathcal{N}(\mathbf{0}, \mathbf{I}_q)$$

we derive the marginal likelihood for j-th feature analytically as in $2b$:

$$p(\mathbf{x}^{(j)}|\mathbf{Z}, \sigma) \sim \mathcal{N}(\mathbf{0}, \mathbf{Z}\mathbf{Z}^T + \sigma^2 \mathbf{I}_D)$$

and we instead optimize 2b with respect to $\mathbf{Z}$, the locale of the points in latent space. The problem is hence given as

$$L = -\frac{DN}{2} \log 2\pi - \frac{D}{2} \log |\mathbf{K}| - \frac{1}{2} \operatorname{tr}(\mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T) \tag{3}$$

where

$$\mathbf{K} = \mathbf{Z}\mathbf{Z}^T + \sigma^2 \mathbf{I}$$

Theorem 1. *Optimizing 3 with respect to $\mathbf{Z}$ is equivalent to the eigenvalue problem solved in PCA.*

Consider a simple Gaussian process prior over the space of functions that are fundamentally linear, but are corrupted by Gaussian noise of variance $\sigma^2 I$. The resulted covariance function, or kernel, for such a prior is given by

$$\mathbf{K} = \mathbf{Z}\mathbf{Z}^T + \sigma^2 \mathbf{I} \tag{4}$$

We observe that 4 is identical to the covariance associated with each factor of the marginal likelihood for PPCA. This marginal likelihood is therefore a product of D independent Gaussian processes. In PCA we are optimizing the parameters and input positions of a Gaussian process prior distribution where the (linear) covariance function for each dimension is given by $\mathbf{K}$.

4.2 Relation to other Forms of Dimensionality Reduction

In the previous section we interpreted PPCA as the special case of a GPLVM where the output dimensions are a priori assumed to be linear. In fact, the linearity assumption can be broken by replacing the inner product kernel with a non-linear covariance function. From this point of view, we can treat GPLVM as an interpretation for non-linear probabilistic version of PCA.

In this section, we introduce the connection of GPLVM with proximity data based methods such as probabilistic kernel PCA (PKPCA) and classical MDS. These connections are through a unifying objective function which embraces all three models.

Classical MDS and PKPCA rely on proximity data, such as similarity matrices, denoted by $\mathbf{S}$. If $\mathbf{S}$ is assumed to be positive semi-definite, we can treat $\mathbf{S}$ as a covariance matrix. Then, the cross-entropy loss between this Gaussian and the Gaussian process whose marginal log-likelihood is given in 2b is given as

$$-\int \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{S}) \log \mathbf{N}(\mathbf{x}|\mathbf{0}, \mathbf{K}) d\mathbf{x} = \frac{N}{2} \log 2\pi + \frac{1}{2} \log |\mathbf{K}| + \frac{1}{2} \operatorname{tr}\left(\mathbf{K}^{-1}\mathbf{S}\right) \tag{5}$$

If we substitute $S = D^{-1}\mathbf{X}\mathbf{X}^T$ we observe, up to a scaling of $-D$, that 5 is identical to 3. Therefore by taking $\mathbf{K}$ to be 4 and minimize 5 with respect to $\mathbf{Z}$ we recover the GPLVM formulation of PPCA. Observing that the entropy of $\mathcal{N}(\mathbf{x}|0, \mathcal{S})$ is constant in $\mathbf{Z}$, we may subtract it from 5 and attain the KL divergence between the two Gaussians,

$$\begin{aligned} \mathrm{KL}(\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{S})||\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{K})) &= -\int \mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{S}) \log \frac{\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{K})}{\mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{S})} \\ &= \frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\log|\mathbf{S}| + \frac{1}{2}\mathrm{tr}(\mathbf{S}\mathbf{K}^{-1}) - \frac{N}{2} \end{aligned} \tag{6}$$

With appropriate choice of $\mathbf{S}$ and $\mathbf{K}$, 6 is a valid objective function for PPCA, PKPCA, classical MDS, and the GPLVM. In particular, for PKPCA $\mathbf{S}$ is a non-linear kernel and $\mathbf{K}$ is the linear kernel. For the GPLVM $\mathbf{S}$ is the linear kernel whereas $\mathbf{K}$ is a non-linear kernel.

4.3 Optimization of the Non-linear Model

In previous sections we observed how PCA can be interpreted as a Gaussian process that maps latent space points to points in observation-space. The positions of the points in the latent space can be determined by maximizing the process likelihood with respect to $\mathbf{Z}$. For the linear kernel, a closed form solution could be obtained up to an arbitrary rotation matrix. However, for non-linear kernels, there will be no such closed form solution and there are likely to be multiple local optima. To use a particular kernel in the GPLVM we first note that the gradient of 3 with respect to the latent points can be found through first taking the gradient with respect to the kernel,

$$\frac{\partial L}{\partial \mathbf{K}} = \mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1} - D\mathbf{K}^{-1} \tag{7}$$

and then combining it with $\frac{\partial \mathbf{K}}{\partial \mathbf{Z}}$ via the chain rule. Note that computation of 7 is independent of the kernel choice, and thus we only require that the gradient of the kernel with respect to the latent points can be computed. The log-likelihood is then optimized via iterative gradient methods.

4.3.1 Sparsification

A caveat worthy of notice is that, while the gradient update with respect to $\frac{\partial L}{\partial \mathbf{K}}\frac{\partial \mathbf{K}}{\partial \mathbf{Z}}$ is straightforward, each gradient step requires an inverse of the kernel matrix (see equation 7), an $O(N^3)$ operation, which renders the algorithm impractical for large datasets. A practical algorithm is presented in Lawrence (2005) through sparsification in kernel methods.

In particular, kernel methods may be sped up through representing the data set by an *active set*, I, which is selected sequentially according to the reduction in the posterior's entropy that they induce via the informative vector machine (IVM) Herbrich *et al.* (2003). A consequence of this enforced sparsification is that optimization of the points in the active set (with $|I| = M < N$) proceeds much quicker than the optimization of the full set of latent variables. The likelihood of the active set is given by

$$p(\mathbf{X}_I) = \frac{1}{(2\pi)^{D/2}|\mathbf{K}_{I,I}|^{1/2}} \exp\big(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_{I,I}^{-1}\mathbf{X}_I\mathbf{X}_I^T)\big) \tag{8}$$

which can be optimized with respect to the kernel's parameters and $\mathbf{Z}_I$ with gradient evaluations of cost $O(M^3)$ rather than the prohibitive $O(N^3)$ in the full model. The dominant cost becomes that of the active set selection which is $O(M^2N)$.

4.3.2 Latent Variable Optimization

While selecting an active set can certainly speed up calculations, for tasks such as data visualization we still need to optimize with respect to inactive points. A standard result for Gaussian processes (*e.g.* Goldberg *et al.* (1998)) is that a point j from the inactive set, denoted by J, can be shown to project into the data-space as a Gaussian distribution

$$p(\mathbf{X}_j|\mathbf{Z}_j) \sim \mathcal{N}(\mathbf{X}^T\mathbf{K}_{I,I}^{-1}\mathbf{k}_{I,j}, k(\mathbf{Z}_j,\mathbf{Z}_j) - \mathbf{k}_{I,j}^T\mathbf{K}_{I,I}^{-1}\mathbf{k}_{I,j}\mathbf{I}_D) \tag{9}$$

where I is the active set. We observe that gradients with respect to $\mathbf{Z}_j$ do not depend on other data in the inactive set J. Therefore, we can independently optimize the likelihood of each $\mathbf{X}_j$ with respect to corresponding $\mathbf{Z}_j$. We then reselect the active set and iterate until the convergence criterion is met. An algorithm for practical GPLVM optimization is organized as follows. Note that in each iteration we perform two active set selections because the choice of active set is dependent on both the kernel parameters and the latent point positions.

Algorithm 1 Efficient GPLVM optimization

```
Require: A size for the active set, $M$. A number of iterations, $T$.
  Initialize $\mathbf{Z}$ through PCA
  for $T$ iterations do
      Select a new active set using the IVM algorithm
      Optimize 8 with respect to the parameters of $\mathbf{K}$ and the latent positions $\mathbf{Z}_I$
      using iterative gradient methods
      Select a new active set
      for each point $j$ not in active set do
          Optimize 9 with respect to $\mathbf{Z}_j$ using iterative gradient methods
      end for
  end for
```

4.4 Constraint Based GPLVMs

Thus far we have shown the potential of GPLVMs in generalizing unsupervised learning techniques. However, conventional GPLVM needs not make any assumptions on the prior of latent variables and thus, is prone to overfitting. One of the effective approaches to tackle this problem is to introduce constraints into the prior for posterior estimation. The variability of constraints, in turn, has enabled GPLVM in adapting to different tasks, such as supervised learning and manifold learning.

4.4.1 Discriminative GPLVM

GPLVM has been shown to discover low dimensional manifolds given only a small number of examples Tipping & Bishop (1997). Under this motivation, discriminative GPLVM (D-GPLVM) is developed and widely used for Gaussian process classification. using an information prior for latent positions $\mathbf{Z}$ Urtasun & Darrell (2007):

$$p(\mathbf{Z}) = \frac{1}{Z_d}\exp\{-\frac{1}{\sigma_d^2}J^{-1}\} \tag{10}$$

where Z_d and σ_d are global normalization constants, $J(\mathbf{Z}) = \mathrm{tr}(S_w^{-1}S_b)$ measures the trade-off between maximizing class separability (S_b) and minimizing within-class separability (S_w), respectively defined as

$$S_b = \sum_{j=1}^{L}\frac{n_j}{N}\big[\frac{1}{n_j}\sum_{i=1}^{n_j}(\mathbf{Z}_i^{(j)} - \mu^{(j)})(\mathbf{Z}_i^{(j)} - \mu^{(j)})^T\big]$$

$$S_w = \sum_{j=1}^{L}\frac{n_j}{N}(\mu^{(j)} - \mu)(\mu^{(j)} - \mu)^T,$$

where L is the number of distinct output labels, n_j the number of labels in class j, and $\mu^{(j)}$ the average of latent positions in class j. 10 encourages latent positions of the same class to be close and those of different classes to be far. Finally, we impose a kernel that is a sum of RBF, a bias term, and a noise term,

$$k(\mathbf{Z}_i, \mathbf{Z}_j) = \gamma_1\exp(-\frac{\gamma_2}{2}||\mathbf{Z}_i - \mathbf{Z}_j||^2) + \gamma_3 + \frac{I_{(\mathbf{Z}_i = \mathbf{Z}_j)}}{\gamma_4},$$

where $\gamma = \{\gamma_1, \gamma_2, ...\}$ comprises the output parameters that govern the output variance. The posterior can then be written as

$$p(\mathbf{Z}, \hat{\theta}|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{Z}, \hat{\theta})p(\mathbf{Z})p(\gamma),$$

with log-likelihood maximized by

$$\operatorname*{argmax}_{\mathbf{Z},\gamma} \; l(\mathbf{X};\mathbf{Z},\theta) + \frac{1}{\sigma_d^2}\text{tr}(S_w^{-1}S_b) + \sum_i \gamma_i.$$

A new data point $(\mathbf{Z}', \mathbf{X}')$ is then predicted by maximizing $p(\mathbf{Z}', \mathbf{X}'|\mathbf{Z}, \mathbf{X}, \gamma)$.

4.4.2 Breaking I.I.D. Assumptions

Thus far we have implicitly assumed that our training and test data have been sampled independently and identically distributed (i.i.d.). Gao *et al.* (2011b) propose a modification on the latent variable model framework to extend to the case where i.i.d assumptions are not appropriate. Specifically, it assumes that the training set $\mathbf{X}$ and testing set $\mathbf{X}_t$ are drawn from two different distributions as well as corresponding latent variables $\mathbf{Z}$ and $\mathbf{Z}_t$ respectively. KL-divergence is then employed in order to modify model parameters. We define a new objective function:

$$\operatorname*{argmax}_{\mathbf{Z},\theta} \; l(\mathbf{X};\mathbf{Z},\theta) + \sum_{j=1}^{D} KL(p(\mathbf{X}^{(j)}, \mathbf{X}_t^{(j)}) \tag{11}$$

This new objective function places penalty on the divergence of the training and test datasets. To the best of our knowledge, there does not exist theoretical guarantees for a divergence penalty in this context. Instead only empirical results are presented (Gao *et al.* , 2011b).

4.5 Generation Process Based GPLVM

4.5.1 Capturing Common Structure Between Multiple Datasets

Often it is the case that multiple datasets can be related to the same underlying phenomenon. For example, neural recordings in two distinct brain areas may be associated in complex and often nonlinear ways (Semedo *et al.* , 2014). While an analysis of the individual neural populations may provide useful information, it does not consider potential correlations between the two datasets. For example, there could be a third brain region that drives the neuronal firing in the two observed regions. Shared GPLVM demonstrate an attempt to reformulate the GPLVM framework in order to address this particular problem (Ek *et al.* , 2007). Specifically we consider latent variable $\mathbf{Z} = [z_1, ..., z_n]$ and two sets of observed variables $\mathbf{X} = [y_1, ..., y_n]$, $\mathbf{X} = [x_1, ..., x_n]$, where $\mathbf{X}$ and $\mathbf{Y}$ are independent given latent $\mathbf{Z}$. $\mathbf{X}$ and $\mathbf{Y}$ are both of size n, with the ith sample related by the ith latent variable sample, however, $\mathbf{X_i}$ and $\mathbf{Y_i}$ are not constricted to be the same dimensionality. We make the assumption that both X and Y are generated in the following manner (Li & Chen, 2016):

$$\begin{aligned} \mathbf{X}_i &= f_X(z_i) + \mathcal{N}(0, \sigma_X^2\mathbf{I}); \;\; f_X \sim GP(\mu_X, K_X) \\ \mathbf{Y}_i &= f_Y(z_i) + \mathcal{N}(0, \sigma_Y^2\mathbf{I}); \;\; f_Y \sim GP(\mu_Y, K_Y) \end{aligned} \tag{12}$$

Using θ to denote the hyper-parameters of our model, we can now exploit our conditional independence assumption and integrate out the f_Z and f_Y terms to write:

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}|f_X, f_Y, X, \theta_X, \theta_Y) &= \prod_{i=1}^{n} P(X_i|f_Z, Z_i, \theta_Z)P(Y_i|f_Y, Z_i, \theta_Y) \\ &= \prod_{i=1}^{n} P(X_i|Z_i, \theta_Z)P(Y_i|Z_i, \theta_Y) \end{aligned} \tag{13}$$

5 Kernel Method Based GPLVM

5.1 Utilizing GPLVM to Compensate for Overrepresented Labels

We now consider the case of supervised data with imbalanced data labels. Without loss of generality, we consider the case of two classes. The more common classes can often dominate the objective function, resulting in poor model performance. In an attempt to circumvent this problem, the latent space (and hence the kernel function in our GPLVM) can be partitioned into "private" and "shared" components (Yousefi *et al.* , 2016). Namely, we can partition our latent variable

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_{\text{shared}} \\ \mathbf{Z}_{\text{private}} \end{bmatrix}$$

Now we can define our Kernel function as a sum between the shared and private components. Specifically, we define:

$$K(Z_{\text{C1}}, Z_{\text{C2}}) = K_{\text{shared}}(Z_{\text{shared, C1}}, Z_{\text{shared, C2}}) + K_{\text{private}}(Z_{\text{private, C1}}, Z_{\text{private, C2}})$$

where C_Y is the category for X. The shared Kernel can be constructed in an manner. However, the shared kernel must take the form:

$$K_{\text{private}}(Z_{\text{private, C1}}, Z_{\text{private, C2}}) = I(C_1 = C_2)K_{\text{covar}}(Z_{\text{private, C1}}, Z_{\text{private, C2}})$$

where I is the indicator function and K_{covar} calculates the covariance (Yousefi *et al.* , 2016). This partitioning allows for the public and private latent variable components to operate independently of each other, thus helping to ensure that our overrepresented category does not dominate the model.

6 Proof outlines for the results

Proof of Theorem 1. The gradients of 3 with respect to $\mathbf{Z}$ may be found as

$$\frac{\partial L}{\partial \mathbf{Z}} = \mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Z} - D\mathbf{K}^{-1}\mathbf{Z}$$

with a critical point given by

$$\mathbf{Z} = \frac{1}{D}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Z}.$$

By deriving the dual objective of 3, we deduce the critical point is equivalently attained at

$$\mathbf{Z} = \mathbf{U}\mathbf{L}\mathbf{V}^T$$

where $\mathbf{U}$ is an $N \times q$ matrix whose columns are the first q eigenvectors of $\mathbf{X}\mathbf{X}^T$, $\mathbf{L}$ is a $q \times q$ diagonal matrix whose j-th element is $(\lambda_j - \sigma^2)^{-1/2}$ where λ_j is the eigenvalue associated with the j-th eigenvector of $D^{-1}\mathbf{X}\mathbf{X}^T$ and $\mathbf{V}$ is an arbitrary $q \times q$ rotation matrix Lawrence (2005). Without loss of generality we assume that these eigenvalues are ordered according to magnitude in decreasing order. Then we observe that the eigenvalue problem in the dual solution is of the form

$$\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}\mathbf{\Lambda}$$

Multiplying both sides by $\mathbf{X}^T$ yields

$$\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{X}^T\mathbf{U}\mathbf{\Lambda} \tag{14}$$

Imposing eigenvectors to be orthonormal, we deduce $\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{\Lambda}$, and hence the matrix $\mathbf{U}' = \mathbf{X}^T\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}$ is orthonormal. Multiplying both sides of 14 by $\mathbf{\Lambda}^{-\frac{1}{2}}$, we conclude that

$$\mathbf{X}^T\mathbf{X}\mathbf{U}' = \mathbf{U}'\mathbf{\Lambda},$$

which is the eigenvalue problem associated with PPCA. □

7 Conclusion

GPLVMs are a flexible latent-variable model framework that can elucidate underlying structure in data. While originally developed as a data visualization tool, the model framework has been exploited to tackle a wide variety of problems in the machine learning field ((Lawrence, 2005)). We presented many augmentations of the original model and their relationship to the original framework and demonstrated that PCA can be thought of as a special case of GPLVM. However, while these models seem promising, our survey of current literature revolving GPLVMs revealed a number of gaps in the theoretical foundations of these models.

For example, there is no theoretical framework dictating model performance in the presents of data outliers and noisy data. It is commonly knowledge that the vast majority of dimensionality reduction techniques, such as Isomap, Local Linear Embedding and factor analysis, are sensitive to outlying data. Many attempts have been made to better understand and rectify how these models behavior under these conditions (Pison *et al.* , 2003; Chen & Liu, 2011; Choi & Choi, 2007). However, it seems that the behavior of GPLVM models under these circumstances has not received the same amount of attention and no method exists that provides robustness to such conditions.

As with many kernel-based methods, the selection of a kernel provides a wide range of possibilities, and thus opportunities for improvement. The essential step towards applying GPLVMs to a variety of contexts is to choose appropriate prior and mercer kernel. Thus, it seems natural that the development of algorithm with theoretical guarantee that automatically select prior and kernel would arise. However, a specific method for such selection has alluded the GPLVM community.

In summary, we illustrated the flexible framework by highlighting key contributions in the field, such as the extensions of GPLVMs for supervised situations, utilization in dealing with label imbalance, operations in sparse and non-I.I.D settings and the optimization characteristics of the loss function. PCA, and various other dimensionality reduction techniques, can be thought of as a special case of GPLVM. The flexible nature of this model provides substantial room for augmentations and improvements beyond those highlighted here.

References

Chen, Jing, & Liu, Yang. 2011. Locally linear embedding: a survey. *Artificial Intelligence Review*, **36**(1), 29–48.

Choi, Heeyoul, & Choi, Seungjin. 2007. Robust kernel isomap. *Pattern Recognition*, **40**(3), 853–862.

Ek, Carl Henrik, Torr, Philip HS, & Lawrence, Neil D. 2007. Gaussian process latent variable models for human pose estimation. *Pages 132–143 of: International workshop on machine learning for multimodal interaction*. Springer.

Gao, Xinbo, Wang, Xiumei, Tao, Dacheng, & Li, Xuelong. 2011a. Supervised Gaussian process latent variable model for dimensionality reduction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **41**(2), 425–434.

Gao, Xinbo, Wang, Xiumei, Li, Xuelong, & Tao, Dacheng. 2011b. Transfer latent variable model based on divergence analysis. *Pattern Recognition*, **44**(10-11), 2358–2366.

Goldberg, Paul W, Williams, Christopher KI, & Bishop, Christopher M. 1998. Regression with input-dependent noise: A Gaussian process treatment. *Pages 493–499 of: Advances in neural information processing systems*.

Herbrich, Ralf, Lawrence, Neil D, & Seeger, Matthias. 2003. Fast sparse Gaussian process methods: The informative vector machine. *Pages 625–632 of: Advances in neural information processing systems*.

Lawrence, Neil. 2005. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of machine learning research*, **6**(Nov), 1783–1816.

Li, Ping, & Chen, Songcan. 2016. A review on gaussian process latent variable models. *CAAI Transactions on Intelligence Technology*, **1**(4), 366–376.

Loehlin, John C. 1998. *Latent variable models: An introduction to factor, path, and structural analysis*. Lawrence Erlbaum Associates Publishers.

Pison, Greet, Rousseeuw, Peter J, Filzmoser, Peter, & Croux, Christophe. 2003. Robust factor analysis. *Journal of multivariate analysis*, **84**(1), 145–172.

Semedo, Joao, Zandvakili, Amin, Kohn, Adam, Machens, Christian K, & Byron, M Yu. 2014. Extracting latent structure from multiple interacting neural populations. *Pages 2942–2950 of: Advances in neural information processing systems*.

Tipping, Michael E., & Bishop, Christopher M. 1997. *Probabilistic principal component analysis*. Tech. rept.

Urtasun, Raquel, & Darrell, Trevor. 2007. Discriminative Gaussian process latent variable model for classification. *Pages 927–934 of: Proceedings of the 24th international conference on Machine learning*. ACM.

Yousefi, Fariba, Dai, Zhenwen, Ek, Carl Henrik, & Lawrence, Neil. 2016. Unsupervised Learning with Imbalanced Data via Structure Consolidation Latent Variable Model. *arXiv preprint arXiv:1607.00067*.