Probabilistic PCA and Extensions

Michael Montemurri [1] **Ahmed Mhedhbi** [2]

Abstract

abstract goes here

```
dvips -Ppdf -tletter -G0 -o paper.ps paper.dvi
ps2pdf paper.ps
```

1. Introduction

Principal Component Analysis (PCA) has long served as a cornerstone in data analysis and dimensionality reduction, with applications spanning image processing, bioinformatics, finance, and natural language processing. By projecting high-dimensional data onto a lower-dimensional subspace, PCA efficiently captures the most significant features of the data while filtering out noise. However, for over 90 years, classical PCA lacked a formal probabilistic interpretation.

This limitation was addressed in 1997 when **Tipping and Bishop** introduced **Probabilistic PCA (PPCA)** (), framing PCA within a probabilistic model. PPCA models observed data as a linear transformation of lower-dimensional latent variables, with Gaussian noise accounting for variations not captured by the latent structure. This approach not only quantifies uncertainty in the data but also derives posterior distributions over the latent variables, enabling their estimation given observed data. Moreover, PPCA's probabilistic foundation facilitates parameter estimation through Bayesian techniques, such as the Expectation-Maximization (EM) algorithm.

While PPCA does not inherently improve PCA's performance in dimensionality reduction, its probabilistic nature enables powerful model extensions. **Tipping and Bishop** extended PPCA to **Mixture of PPCA Models**, allowing for data generated from multiple sources or clusters. **Zhang et al.** introduced nonlinearity into PPCA through kernel methods, resulting in **Probabilistic Kernel PCA** (PKPCA). Further advancing this, **Alvarez et al.** incorporated time-dependent kernel functions, enabling PPCA to model temporal dependencies in time-series data by leveraging concepts from **Hidden Markov Models (HMMs)**.

In this report, we explore the theoretical foundations and practical applications of PPCA and its extensions. We begin with a review of classical PCA and PPCA before investigating key extensions, including Mixture Models and Probabilistic Kernel PCA. We introduce original contributions by employing Bayesian optimization to identify optimal kernel parameters and functions for PKPCA. Experimental results are presented to compare analytical and EM-based PPCA, evaluate Mixture Models, and assess the performance of PKPCA in dynamic settings. Through this work, we aim to highlight PPCA's versatility in tackling complex data challenges while offering novel insights to the field.

2. PCA and the Probabilistic PCA model

2.1. PCA and Its Limitations

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction and feature extraction. It identifies a lower-dimensional subspace by projecting data onto orthonormal axes, called principal components, such that the retained variance under projection is maximized. Formally, let $\mathbf{X} \in \mathbb{R}^{N \times d}$ represent a dataset of N observations, where each observation $\mathbf{t}_n \in \mathbb{R}^d$ is a d-dimensional vector. The sample covariance matrix is defined as

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{t}_n - \bar{\mathbf{t}})(\mathbf{t}_n - \bar{\mathbf{t}})^\top,$$

where $\bar{\mathbf{t}}$ is the sample mean. The principal components are given by the dominant eigenvectors $\mathbf{w}_j$ (those associated with the largest eigenvalues λ_j) of $\mathbf{S}$, satisfying $\mathbf{S}\mathbf{w}_j = \lambda_j \mathbf{w}_j$. By projecting the data onto the subspace spanned by the first q principal components, PCA provides a q-dimensional representation $\mathbf{x}_n = \mathbf{W}^\top(\mathbf{t}_n - \bar{\mathbf{t}})$, where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_q]$.

where:

- $\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top$ is the sample covariance matrix,
- $\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$ is the mean of the dataset,

*Equal contribution [1] Department of Mathematics and Statistics, McGill University, Montréal, Canada [2] Departement d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada. Correspondence to: Michael Montemurri <michael.montemurri@mail.mcgill.ca>, Ahmed Mhedhbi <>.

IFT 6269 project report - Université de Montréal

- $\mathbf{U} \in \mathbb{R}^{d\times q}$ is a matrix whose columns are the top q eigenvectors of $\mathbf{S}$, corresponding to the largest q eigenvalues.

While PCA provides an optimal linear representation in terms of variance maximization, it suffers from several limitations:

1. **Linear Assumption**: PCA assumes that the data lies in or near a linear subspace, making it unsuitable for datasets with complex nonlinear structures.
2. **No Probabilistic Interpretation**: Classical PCA lacks a probabilistic framework, preventing it from quantifying uncertainty or modeling latent variable distributions.
3. **Global Representation**: PCA provides a single global linear subspace, which may not adequately capture multimodal or heterogeneous data distributions.

These limitations motivated the development of Probabilistic PCA (PPCA), which introduces a probabilistic interpretation to address some of these challenges. In the following sections, we explore the PPCA model and its extensions, which build on the foundational principles of PCA to overcome these limitations.

NOTATION USED THROUGHOUT THE PAPER

- $\mathbf{X} \in \mathbb{R}^{N\times d}$: Observed dataset of N samples, each with d dimensions.
- $\mathbf{x}_n \in \mathbb{R}^d$: n-th data point.
- $\bar{\mathbf{x}} \in \mathbb{R}^d$: Mean of the dataset.
- $\mathbf{S} \in \mathbb{R}^{d\times d}$: Sample covariance matrix.
- $\mathbf{U} \in \mathbb{R}^{d\times q}$: Matrix of principal components (top q eigenvectors of $\mathbf{S}$).
- $\mathbf{z}_n \in \mathbb{R}^q$: Lower-dimensional representation of $\mathbf{x}_n$ in the principal component subspace.
- q: Target dimensionality ($q < d$).

This notation will serve as the basis for the probabilistic reformulation and extensions of PCA discussed in subsequent sections.

2.2. The PPCA Model

Probabilistic Principal Component Analysis (PPCA) extends classical PCA by introducing a probabilistic framework, which provides a generative interpretation of the observed data.

In PPCA, the observed d-dimensional data, $\mathbf{x}_n \in \mathbb{R}^d$, is modeled as a linear transformation of q-dimensional latent variables $\mathbf{z}_n \in \mathbb{R}^q$, with additive Gaussian noise. The generative model is given by:

$$\mathbf{x}_n = \mathbf{W}\mathbf{z}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}),$$

where:

- $\mathbf{W} \in \mathbb{R}^{d\times q}$ is the weight matrix that maps the latent variables to the observed space,
- $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector of the observed data,
- $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ represents isotropic Gaussian noise with variance σ^2.

The latent variables $\mathbf{z}_n$ are assigned a standard normal prior:

$$p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\mathbf{I}$ is the identity matrix.

The marginal likelihood of the observed data $\mathbf{x}_n$ can then be derived by integrating over the latent variables:

$$p(\mathbf{x}_n) = \int p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)\, d\mathbf{z}_n.$$

By substituting the Gaussian form of $p(\mathbf{x}_n|\mathbf{z}_n)$ and $p(\mathbf{z}_n)$, the observed data $\mathbf{x}_n$ follows a multivariate Gaussian distribution:

$$p(\mathbf{x}_n) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I},$$

where $\mathbf{C}$ is the covariance matrix capturing the contributions of both the latent variables and noise.

The posterior distribution of the latent variables $\mathbf{z}_n$ given the observed data $\mathbf{x}_n$ is also Gaussian:

$$p(\mathbf{z}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^\top(\mathbf{x}_n - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}),$$

where:

$$\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}.$$

The posterior provides a probabilistic estimate of the latent variables, incorporating uncertainty into the low-dimensional representation of the data.

Finally, the log-likelihood of the entire dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is expressed as:

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = -\frac{N}{2}\left[d\ln(2\pi) + \ln|\mathbf{C}| + \mathrm{Tr}(\mathbf{C}^{-1}\mathbf{S})\right],$$

where $\mathbf{S}$ is the sample covariance matrix:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top,$$

and $\bar{\mathbf{x}}$ is the mean of the observed dataset.

2.3. Maximum Likelihood Estimation of PPCA Parameters

The maximum likelihood estimate of the mean vector is simply the empirical mean of the observed data:

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n,$$

where N is the number of observations.

For the weight matrix $\mathbf{W}$, the log-likelihood is maximized when its columns are aligned with the eigenvectors of the sample covariance matrix $\mathbf{S}$ that correspond to the q largest eigenvalues. The sample covariance matrix $\mathbf{S}$ is defined as:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^\top.$$

Differentiating the log-likelihood with respect to $\mathbf{W}$ gives us the following:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = N(\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}\mathbf{W} - \mathbf{C}^{-1}\mathbf{W}) = 0,$$

now using $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}$.

We get the maximum likelihood estimate of the weight matrix as:

$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R},$$

where:

- $\mathbf{U}_q \in \mathbb{R}^{d\times q}$ is the matrix of eigenvectors of $\mathbf{S}$ corresponding to the q largest eigenvalues,
- $\boldsymbol{\Lambda}_q \in \mathbb{R}^{q\times q}$ is the diagonal matrix of the q largest eigenvalues,
- $\mathbf{R} \in \mathbb{R}^{q\times q}$ is an arbitrary orthogonal rotation matrix.

The log-likelihood achieves its global maximum when the columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{S}$ corresponding to the q largest eigenvalues.

The maximum likelihood estimate of the noise variance σ^2 is given by:

$$\sigma^2_{\mathrm{ML}} = \frac{1}{d-q}\sum_{j=q+1}^{d} \lambda_j,$$

where $\lambda_{q+1}, \ldots, \lambda_d$ are the smallest $d-q$ eigenvalues of the sample covariance matrix $\mathbf{S}$. This ensures that the variance not explained by the first q principal components is captured by the noise term.

These MLE estimates form the basis for the PPCA model, allowing it to capture both the principal structure of the data and the uncertainty inherent in the observations.

2.4. EM Algorithm for PPCA

Although the maximum likelihood estimates for PPCA parameters can be computed directly using the eigendecomposition of the sample covariance matrix, the probabilistic framework of PPCA enables the use of the Expectation-Maximization (EM) algorithm. The EM algorithm provides a natural and iterative approach for estimating the parameters $\mathbf{W}$, $\boldsymbol{\mu}$, and σ^2 in the PPCA model. This method can offer computational advantages, especially for high-dimensional data (d) or when the dataset is incomplete, as it avoids explicitly forming and diagonalizing the sample covariance matrix.

2.4.1. E-Step

In the E-step, we compute the expected value of the latent variables $\mathbf{z}_n$ and their second moments, conditioned on the observed data:

$$\mathbb{E}[\mathbf{z}_n|\mathbf{x}_n] = \mathbf{M}^{-1}\mathbf{W}^\top(\mathbf{x}_n - \boldsymbol{\mu}),$$

$$\mathbb{E}[\mathbf{z}_n\mathbf{z}_n^\top|\mathbf{x}_n] = \sigma^2\mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}_n|\mathbf{x}_n]\mathbb{E}[\mathbf{z}_n|\mathbf{x}_n]^\top,$$

where:

$$\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}.$$

These expectations reflect the posterior mean and covariance of the latent variables under the current parameter estimates.

2.4.2. M-Step

In the M-step, we update the parameters $\mathbf{W}$, $\boldsymbol{\mu}$, and σ^2 by maximizing the expected complete-data log-likelihood.

The weight matrix is updated as:

$$\mathbf{W}_{\text{new}} = \mathbf{S}\mathbf{W}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^\top\mathbf{S}\mathbf{W})^{-1},$$

The noise variance is updated as:

$$\sigma^2_{\text{new}} = \frac{1}{d}\left[\mathrm{Tr}(\mathbf{S}) - \mathrm{Tr}(\mathbf{S}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}_{\text{new}}^\top)\right].$$

The EM algorithm avoids directly diagonalizing the sample covariance matrix and provides a robust framework for high-dimensional and incomplete data.

3. Extensions of PPCA

3.1. Mixture of PPCA Models

The Mixture of PPCA models (MPPCA) extends the probabilistic framework of PPCA to capture multimodal data distributions. By combining multiple PPCA components, each of which models a local linear subspace, MPPCA is well-suited for datasets with heterogeneous or clustered structures.

3.1.1. Generative Model

MPPCA assumes that the observed data $\mathbf{x}_n \in \mathbb{R}^d$ is generated from a mixture of K local PPCA components. Each component corresponds to a Gaussian distribution parameterized by a local mean $\boldsymbol{\mu}_k$ and covariance matrix $\mathbf{C}_k$.

The generative process for the data can be described as follows:

1. A latent variable $z_n \in \{1, \ldots, K\}$ is drawn from a categorical prior distribution:

$$p(z_n = k) = \pi_k, \quad \text{where} \sum_{k=1}^{K} \pi_k = 1.$$

   Here, π_k represents the mixing coefficients (the prior probabilities of each component).

2. Conditioned on $z_n = k$, the observed data $\mathbf{x}_n$ is generated from a Gaussian distribution with parameters $(\boldsymbol{\mu}_k, \mathbf{C}_k)$:

$$p(\mathbf{x}_n | z_n = k) = \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k),$$

   where the covariance matrix $\mathbf{C}_k$ is defined as:

$$\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^\top + \sigma_k^2 \mathbf{I}.$$

   - $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the mean of the k-th PPCA component.
   - $\mathbf{W}_k \in \mathbb{R}^{d \times q}$ maps the q-dimensional latent subspace to the observed d-dimensional space.
   - $\sigma_k^2 \mathbf{I}$ accounts for the isotropic Gaussian noise.

The overall marginal distribution of the observed data $\mathbf{x}_n$ is obtained by summing over all K components (marginalizing out the latent variable z_n):

$$p(\mathbf{x}_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k).$$

This mixture model captures the multimodal nature of complex datasets, where each Gaussian component corresponds to a distinct region or subspace of the data distribution.

3.1.2. Posterior Distribution of the Latent Variables

Given the observed data $\mathbf{x}_n$, the posterior distribution of the latent variable z_n (i.e., the probability that $\mathbf{x}_n$ belongs to the k-th component) is obtained using Bayes' theorem:

$$p(z_n = k | \mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \mathbf{C}_j)}.$$

Here:

- The numerator $\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k)$ represents the joint probability of $\mathbf{x}_n$ and $z_n = k$.

- The denominator $\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \mathbf{C}_j)$ is the marginal likelihood (evidence) of the data.

The posterior $p(z_n = k | \mathbf{x}_n)$ reflects the *responsibility* that the k-th PPCA component has for generating the observation $\mathbf{x}_n$.

3.1.3. Log-Likelihood of the Model

The log-likelihood of the observed dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ under the MPPCA model is expressed as:

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k) \right),$$

where $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{W}_k, \sigma_k^2\}_{k=1}^{K}$ represents the model parameters:

- π_k: mixing coefficients such that $\sum_{k=1}^{K} \pi_k = 1$,

- $\boldsymbol{\mu}_k$: the mean vector for component k,

- $\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^\top + \sigma_k^2 \mathbf{I}$: the covariance matrix.

3.1.4. Responsibilities and Soft Assignment

To assign data points to mixture components, the **responsibility** r_{nk} is introduced. This is the posterior probability that observation $\mathbf{x}_n$ belongs to component k, defined as:

$$r_{nk} = p(z_n = k | \mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \mathbf{C}_j)}.$$

The posterior responsibilities r_{nk} facilitate **soft assignments** of data points to components, allowing overlap among mixture components.

3.1.5. EM Algorithm for MPPCA

In the MPPCA Model, we maximize the parameters for all clusters simultaneously. The E-step, computes the posterior responsibilities r_{nk}, and the M-step updates the model parameters. The algorithm is outlined below:

3.1.6. Algorithm Summary

Algorithm 1 EM Algorithm for MPPCA

```
Input: Observed data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, number of components $K$.
Initialization: Initialize $\pi_k$, $\boldsymbol{\mu}_k$, $\mathbf{W}_k$, and $\sigma_k^2$ for all $k$.
repeat
    E-step: Compute the responsibilities $r_{nk}$ for each observation $n$ and component $k$:
        $\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^\top + \sigma_k^2 \mathbf{I}.$
        $r_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \mathbf{C}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \mathbf{C}_j)}$
    M-step: Update the parameters:
        • Update the mixing coefficients and means:
            $\pi_k = \frac{1}{N} \sum_{n=1}^N r_{nk}, \quad \boldsymbol{\mu}_k = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}}.$
        • Update the covariance matrix:
            $\mathbf{S}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top,$
        • Update Weight Matrix: Perform the eigendecomposition of $\mathbf{S}_k$:
            $\mathbf{S}_k = \mathbf{U}_k \boldsymbol{\Lambda}_k \mathbf{U}_k^\top.$
          Update the weight matrix $\mathbf{W}_k$ using the top $q$ eigenvalues:
            $\mathbf{W}_k = \mathbf{U}_k \left( \boldsymbol{\Lambda}_k - \sigma_k^2 \mathbf{I} \right)^{1/2}.$
        • Update the noise variance:
            $\sigma_k^2 = \frac{1}{d} \left[ \mathrm{Tr}(\mathbf{S}_k) - \mathrm{Tr}(\mathbf{W}_k^\top \mathbf{S}_k \mathbf{W}_k) \right].$
until The log-likelihood converges.
```

3.2. Probabilistic Kernel PCA

Classical Principal Component Analysis (PCA) assumes a linear relationship in the data, limiting its ability to model complex nonlinear structures. Kernel PCA (KPCA) overcomes this limitation by projecting the data into a high-dimensional feature space $\mathcal{F}$ using a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. By operating in this implicit feature space, KPCA enables the extraction of nonlinear relationships while still relying on linear algebraic techniques. Zhang et al. introduced Probabilistic Kernel PCA (PKPCA), combining the nonlinear mapping of KPCA with the framework of PPCA. PKPCA models the data in the kernel-induced feature space $\mathcal{F}$, providing both the flexibility to capture nonlinear structures and the ability to incorporate uncertainty into the model.

3.2.1. Generative Model

In PKPCA, the kernel-induced feature space $\mathcal{F}$ is assumed to consist of mutually independent Gaussian processes. The generative process for the feature vector $\mathbf{g}$ in $\mathcal{F}$ is defined as:

$$\mathbf{g} = \mathbf{B}\mathbf{w} + u\mathbf{1}_n + \boldsymbol{\epsilon},$$

where:

- $\mathbf{g} \in \mathbb{R}^n$ is the feature vector in the kernel-induced space,
- $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the weight matrix mapping the m-dimensional latent variables $\mathbf{w} \in \mathbb{R}^m$ to the n-dimensional feature space,
- u is a scalar allowing the feature space to have a non-zero mean,
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, n\sigma^2\mathbf{I})$ represents Gaussian noise with variance $n\sigma^2$.

The latent variables $\mathbf{w}$ follow a standard Gaussian prior:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Thus, the marginal distribution of the observed data $\mathbf{x}$ in the feature space is:

$$p(\mathbf{g}) = \mathcal{N}(\mathbf{g} | u\mathbf{1}_n, \mathbf{B}\mathbf{B}^\top + n\sigma^2\mathbf{I}).$$

3.2.2. Posterior Distribution

Given the observed data $\mathbf{g}$, the posterior distribution of the latent variables $\mathbf{w}$ is Gaussian:

$$p(\mathbf{w}|\mathbf{g}) = \mathcal{N}(\mathbf{w} | \mathbf{M}^{-1}\mathbf{B}^\top(\mathbf{g} - u\mathbf{1}_n), \sigma^2\mathbf{M}^{-1}),$$

where:

$$\mathbf{M} = \mathbf{B}^\top\mathbf{B} + \sigma^2\mathbf{I}.$$

3.2.3. Kernel Trick and Maximum Likelihood Estimation

To avoid explicitly working in the feature space, PKPCA uses the kernel trick. The kernel matrix is defined as:

$$\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top,$$

where $\mathbf{\Phi}$ maps the data into the feature space. The covariance of the marginal likelihood becomes:

$$\mathbf{C} = \mathbf{K} + n\sigma^2\mathbf{I}.$$

The log-likelihood of the observed data under PKPCA is:

$$\mathcal{L} = -\frac{r}{2}\log|\mathbf{BB}^\top + \sigma^2\mathbf{I}_n| - \frac{r}{2n}\operatorname{Tr}\left((\mathbf{BB}^\top + \sigma^2\mathbf{I}_n)^{-1}\mathbf{HKH}\right),$$

where:

- $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix,
- $\mathbf{K}$ is the kernel matrix.

The model parameters $\mathbf{B}$ and σ^2 are optimized as:

$$\sigma^2_{\text{ML}} = \frac{1}{n-m}\sum_{j=m+1}^{n}\lambda_j,$$

$$\mathbf{B}_{\text{ML}} = \mathbf{U}_m(\mathbf{\Lambda}_m - \sigma^2\mathbf{I})^{1/2}\mathbf{R},$$

where:

- $\mathbf{U}_m$ contains the top m eigenvectors of $\frac{1}{n}\mathbf{HKH}$,
- $\mathbf{\Lambda}_m$ is the diagonal matrix of the top m eigenvalues,
- $\mathbf{R}$ is an arbitrary orthogonal matrix.

3.2.4. EM Algorithm for PKPCA

The parameters of the PKPCA model can also be estimated iteratively using the Expectation-Maximization (EM) algorithm. The algorithm alternates between computing the posterior distribution of the latent variables and updating the model parameters.

Algorithm 2 EM Algorithm for PKPCA

```
Input: Kernel matrix $\mathbf{K}$, dimensionality $m$, noise variance $\sigma^2$, tolerance tol
Initialize: Weight matrix $\mathbf{B}$, mean $u$, noise variance $\sigma^2$
repeat
    E-Step: Compute the posterior distribution of latent variables:
        $p(\mathbf{w}|\mathbf{g}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{B}^\top(\mathbf{g} - u\mathbf{1}_n), \sigma^2\mathbf{M}^{-1}),$
        where $\mathbf{M} = \mathbf{B}^\top\mathbf{B} + \sigma^2\mathbf{I}$.
    M-Step: Update the model parameters:
        • Update noise variance:
            $\sigma^2_{\text{new}} = \frac{1}{n-m}\sum_{j=m+1}^{n}\lambda_j.$
        • Update weight matrix:
            $\mathbf{B}_{\text{new}} = \mathbf{U}_m(\mathbf{\Lambda}_m - \sigma^2_{\text{new}}\mathbf{I})^{1/2}\mathbf{R}.$
until Change in log-likelihood $\mathcal{L}$ is below tol.
Output: Optimized parameters $\mathbf{B}$, $\sigma^2$, and posterior distribution.
```

3.3. Probabilistic Kernel PCA Through Time

4. Original Contributions

Armed with the Probabilistic Kernel PCA Model, we explore applying Bayesian optimization techniques for finding optimal kernel parameters and kernel functions.

5. Implementation and Experimental Results

5.1. Comparison of efficiency of Analytical PPCA versus EM PPCA

5.2. Mixture of PPCA models

5.3. A Real World Application

References