# Probabilistic PCA and Extensions

Michael Montemurri[1] Ahmed Mhedhbi[2]

[1]McGill University
[2]Université de Montréal

December 2024
Group 6

- Tipping, Michael E., and Christopher M. Bishop. *"Mixtures of probabilistic principal component analyzers."* Neural Computation, 11.2 (1999): 443-482.
- Tipping, Michael E., and Christopher M. Bishop. *"Probabilistic principal component analysis."* Journal of the Royal Statistical Society Series B: Statistical Methodology, 61.3 (1999): 611-622.
- Zhang, Zhihua, et al. *"Probabilistic kernel principal component analysis."* Department of Computer Science, The Hong Kong University of Science and Technology, Technical Report (2004).

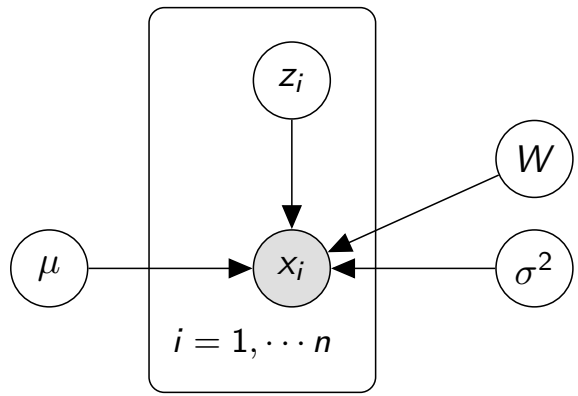
# Fundamental Results of PPCA

**Goal of PPCA:** Providing a probabilistic framework for density modeling and dimensionality reduction.

**Generative Model:**

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

Where: $x_n \in \mathbb{R}^d, z_n \in \mathbb{R}^q$ and $\mathbf{W} \in \mathbb{R}^{d\times q}$ is a matrix mapping latent space to data space.




**Maximum Likelihood Estimation:**

$$\mu_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n} x_n, \quad \mathbf{W}_{\text{ML}} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R}, \quad \sigma^2_{\text{ML}} = \frac{1}{d-q}\sum_{j=q+1}^{d} \lambda_j$$

Where $\mathbf{U}_q, \boldsymbol{\Lambda}_q$ are the matrices of the top $q$ eigenvectors and corresponding eigenvalues of the sample covariance matrix $\mathbf{S}$.

# Mixture of PPCA Models (MPPCA)

**Goal of MPPCA:** Model multimodal data by capturing local linear subspaces instead of a global linear approximation.

**Generative Model:**

$$p(x_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, C_k), \quad C_k = \mathbf{W}_k\mathbf{W}_k^T + \sigma_k^2\mathbf{I}.$$

**Posterior Responsibilities:**

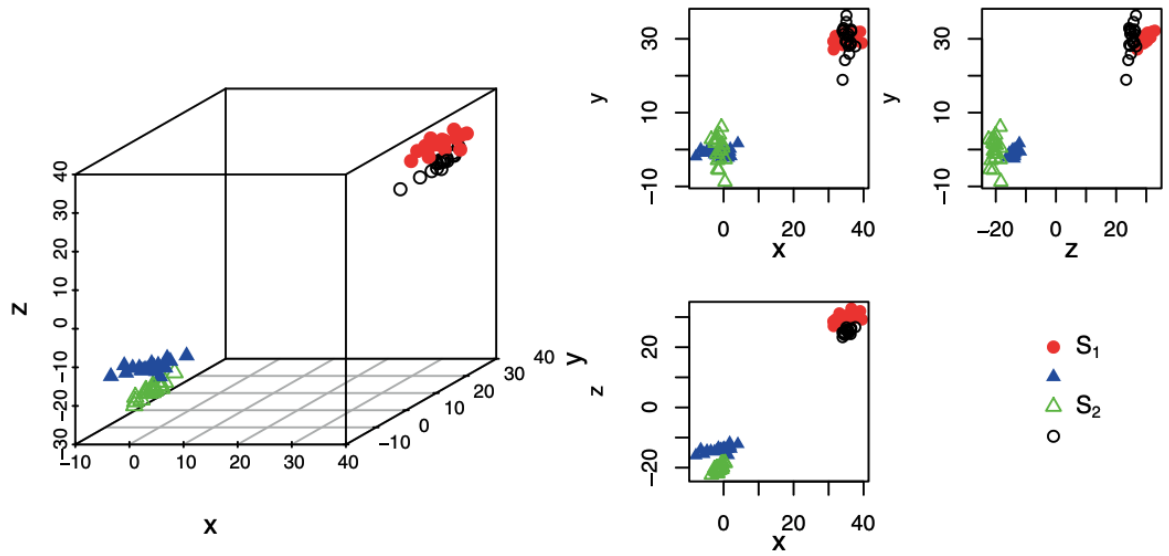
$$R_{nk} = p(z_n = k|\mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{C}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \mathbf{C}_j)}.$$

**M-Step Updates:**

$$\pi_i = \frac{1}{N}\sum_{n=1}^{N} R_{ni}, \boldsymbol{\mu}_i = \frac{\sum_{n=1}^{N} R_{ni}\mathbf{x}_n}{\sum_{n=1}^{N} R_{ni}}, \mathbf{S}_i = \frac{1}{\pi_i N}\sum_{n=1}^{N} R_{ni}(\mathbf{x}_n - \boldsymbol{\mu}_i)(\mathbf{x}_n - \boldsymbol{\mu}_i)^T.$$

The eigen-decomposition of $\mathbf{S}_i$ yields $W_i$ and $\sigma_i^2$. For large $d$, its more efficient to iteratively update them via the EM schedule.

# Implementation of MPPCA




**Relationships and Comparisons of PPCA and MPPCA** :

- Classical PCA corresponds to a special case of probabilistic PCA where the noise covariance approaches zero, $\sigma^2 \to 0$
- PPCA is covariant under rotation
- The columns of the maximum-likelihood estimator $W_{\mathrm{ML}}$ are shown to span the principal subspace of the data without the assumption that $C = S$, contrary to factor analysis.
- Both rely on generative models, but MPPCA extends PPCA to handle heterogeneous datasets.

# Kernel PCA and a Probabilistic Extension (PKPCA)

**Kernel PCA:** Extends PCA to capture nonlinear patterns by mapping data $\mathbf{x}_i$ into a high-dimensional feature space $\mathcal{F}$ via a mapping $\phi(\mathbf{x}_i)$. Instead of explicitly computing in $\mathcal{F}$, the **kernel trick** calculates inner products:

$$K_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle,$$

using a kernel function $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$, to indirectly define the data in $\mathcal{F}$.

**Data Representation:**

$F$ is the matrix of feature vectors $\mathbf{f}_i = \phi(\mathbf{x}_i)$ in $\mathcal{F}$ ( rows of $F$).

$\mathbf{g}_j$ represents the feature dimensions (columns of $F$)

**Key Insight:** Zhang et al. showed that if $\mathbf{K}$ (the kernel matrix) is Wishart-distributed $W_n(r, \Sigma)$, meaning it is a random positive semi-definite matrix modeling the covariance structure of multivariate Gaussian-distributed data, then the columns of $F$ (the $\mathbf{g}_i$'s) are mutually independent $n$-dimensional vectors following:

$$\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}).$$

# Summary of PKPCA and Comparison to PPCA

**Generative Model in $\mathcal{F}$**: Zhang et al. propose that each column $\mathbf{g} \in \mathbb{R}^n$ of the feature matrix $\mathbf{F}$ is generated by

$$\mathbf{g} = \mathbf{B}\mathbf{w} + u\mathbf{1}_n + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{V}),$$

where:

- $\mathbf{g} \in \mathbb{R}^n$: represents feature dimensions (columns of $F$).
- $\mathbf{B} \in \mathbb{R}^{n \times m}$: weight matrix mapping the latent variables $\mathbf{w} \in \mathbb{R}^m$ to the feature space.
- $u\mathbf{1}_n$: scalar bias term for the mean.
- $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{V})$: noise term.

Here, $\mathbf{V} = n\sigma^2\mathbf{I}_n/r$, and $\mathbf{w} \sim \mathcal{N}(0, n\mathbf{I}_m/r)$, where $r$, the dimensionality of the feature space, is unknown. The kernel trick is used to estimate $\mathbf{B}$ and $\sigma^2$.

PKPCA models feature dimensions $\mathbf{g}$ in the kernel-induced feature space $\mathcal{F}$, while PPCA models the observed data $\mathbf{x}$.

# ML Estimates for PKPCA

**ML Estimates:**

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{f}_i, \quad \sigma^2_{\mathrm{ML}} = \frac{1}{n-m}\sum_{j=m+1}^{n}\lambda_j,$$

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

Where $\mathbf{U}_m$, $\boldsymbol{\Lambda}_m$ are matrices of the top $m$ eigenvectors and corresponding eigenvalues of $\frac{1}{n}\mathbf{HKH}$, $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is a centering matrix, and $\lambda_j$ are the eigenvalues of $\mathbf{HKH}$ for $j > m$.

- ML estimates depend only on the kernel matrix $\mathbf{K}$, making explicit computation in the feature space unnecessary.
- Similar to PPCA, an EM algorithm can be used to iteratively estimate parameters for computational efficiency when needed.
- PKPCA model is not a kernel equivalent of the PPCA, but rather in another model based on similar principles.

# Contributions, Future Directions, and Experimental Results

## Contributions

- Unified overview of PPCA, MPPCA, and PKPCA with their relationships and distinctions.
- Exploring Bayesian optimization Framework for Parameter Estimation for PPCA and MPPCA (priors on $W$, $\sigma^2$, mixture proportions $\pi_k$ and number of clusters/regions).

## Experimental Results

- Implemented all algorithms from scratch to confirm theoretical computational advantages of EM for large $d$ on MNIST.
- Comparison of MPPCA with PPCA for image reconstruction
- Comparison of models on synthetic data to demonstrate advantages.

## Future Directions

- Extending MPPCA to dynamic and time-series data for temporal clustering (integrating Hidden Markov Models (HMM) to account for temporal dependencies).
- Applying PKPCA within local clusters in MPPCA (swiss roll)
- Investigating hybrid models combining PPCA/MPPCA with deep generative frameworks like VAEs for scalable applications.