Topics in PCA

Asymptotic, Non-linear, Functional and Probabilistic PCA

Elvis Cui

MS in Biostatistics, UCLA

November 4, 2019

Overview

Hotelling's PCA

In 1933, Hotelling developed Principal Component Analysis (a.k.a. Karhunen-Loeve Transform).

Definition 1 (First Principal Component)

Given a random vector $\mathbf{X} \in \mathbb{R}^p$, the first principal component of $\mathbf{X}$ is given by the following optimization problem:

$$\max_{\mathbf{v}_1} Var(\mathbf{v}_1^T \mathbf{X}) = \mathbf{v}_1^T \mathbf{\Sigma}_\mathbf{X} \mathbf{v}_1$$
$$s.t. \ \|\mathbf{v}_1\| = 1$$

Equivalently,

$$\mathbf{v}_1 = \arg\max_{\mathbf{v}_1} \frac{\mathbf{v}_1^T \mathbf{\Sigma}_X \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1}$$

The maximum is called the **operator norm** of $\mathbf{\Sigma}_X$.

Hotelling's PCA cont'd

Definition 2 (i^{th} Principal Component)

Given the first $(i-1)$ principal component $(i \leq n)$, he i^{th} principal component of $\mathbf{X}$ is defined to be

$$\max_{\mathbf{v}_1} Var(\mathbf{v}_i^T \mathbf{X}) = \mathbf{v}_i^T \mathbf{\Sigma}_{\mathbf{X}} \mathbf{v}_1$$

$$s.t.\ \|\mathbf{v}_i\| = 1, \langle \mathbf{v}_t, \mathbf{v}_i \rangle = 0\ \forall t < i$$

Hotelling's PCA: Sequential Method

We could derive all n principal components sequentially:

- Introducing Lagrangian multiplier λ_1, define

$$f(\mathbf{v}_1) = \mathbf{v}_1^T \mathbf{\Sigma}_X \mathbf{v}_1 + \lambda_1(1 - \mathbf{v}_1^T \mathbf{v}_1)$$

- Taking derivative: $\frac{\partial f(\mathbf{v}_1)}{\partial \mathbf{v}_1} = 2(\mathbf{\Sigma}_X - \lambda_1 \mathbf{I}_p)\mathbf{v}_1 = \mathbf{0}$. The maximum of variance is attained when λ_1 is the largest eigenvalue of $\mathbf{\Sigma}_X$.
- Introducing Lagrangian multiplier λ_2 and μ, define

$$f(\mathbf{v}_2) = \mathbf{v}_2^T \mathbf{\Sigma}_X \mathbf{v}_2 + \lambda_2(1 - \mathbf{v}_2^T \mathbf{v}_2) + \mu \mathbf{v}_2^T \mathbf{v}_2$$

- Taking derivative w.r.t. $\mathbf{v}_2$:

$$\frac{\partial f(\mathbf{v}_2)}{\partial \mathbf{v}_2} = 2(\mathbf{\Sigma}_X - \lambda_2 \mathbf{I}_p)\mathbf{v}_2 + \mu \mathbf{v}_1 = \mathbf{0}$$

Hotelling's PCA: Sequential Method

Cont'd from last slide:

- Pre-multiplying $\mathbf{v}_1$ on both sides

$$2\mathbf{v}_1^T \mathbf{\Sigma}_X \mathbf{v}_2 + \mu = \mathbf{0}$$

- Pre-multiplying the equation $((\mathbf{\Sigma}_X - \lambda_1 \mathbf{I}_p)\mathbf{v}_1 = \mathbf{0})$ by $\mathbf{v}_2$ yields $\mathbf{v}_2^T \mathbf{\Sigma}_X \mathbf{v}_1 = \mathbf{0}$. Thus, $\mu = 0$
- $\mu = 0 \Rightarrow \lambda_2$ is the second largest eigenvalue of $\mathbf{\Sigma}_X$.
- Repeating the above procedure to get all n principal components.

Reduced-rank regression

PCA can also be derived from a classic multivariate analysis framework known as **Reduced-rank Regression**.

Definition 3 (Reduced-rank Regression)

Suppose $\mathbf{X} \in \mathbb{R}^p$ with mean $\mathbb{E}(\mathbf{X}) = \mu_X$, $\mathbf{B} \in \mathbb{R}^{t \times p}$ with $t < p$ and $\mathbf{A} \in \mathbb{R}^{p \times t}$. The goal is to choose $\mathbf{A}, \mathbf{B}$ and μ to minimize a least-square criterion:

$$\min_{\mathbf{A},\mathbf{B},\mu} \mathbb{E}\{\|\mathbf{X} - \mu - \mathbf{ABX}\|_2^2\}$$

Reduced-rank Regression cont'd

Theorem 4 (Solution of Reduced-rank Regression)

The solution of the reduced-rank regression problem is given by the eigenvectors and eigenvalues of $Var(\mathbf{X}) = \mathbf{\Sigma}_X$*:*

$$\mathbf{A} = (\mathbf{v}_1, \cdots, \mathbf{v}_t) = \mathbf{B}^T$$

$$\mu = (\mathbf{I}_p - \mathbf{AB})\mu_X$$

Where $\mathbf{v}_j$ *is the eigenvector associated with the* j^{th} *largest eigenvalue,* λ_j*, of* $\mathbf{\Sigma}_X$*.*

Proof.

e.g., Chapter 7 section 2 of modern multivariate statistical techniques (Izemann, 2013). □

Polynomial PCA

- Adding quadratic, cubic terms (Gnanadesikan and Wilk, 1969)
- **Problem**: Too many terms when dimension of $\mathbf{x}$ is large.

Example 5 (Quadratic PCA)

- $\mathbf{x} \in \mathbb{R}^p \Rightarrow$ Extended $\mathbf{x} \in \mathbb{R}^{2p+p(p-1)/2}$ (p quadratic powers and $p(p-1)/2$ cross terms).
- λ_{min} of the covariance matrix of the extended vector will be 0 if observations follow an exact quadratic curve.

Kernel PCA: Intuition

Kernel PCA can be considered as a two-step process:

- Non-linearly transform the data point $\mathbf{x} \in \mathbb{R}^r$ into $\mathbf{\Phi}_i$ in a high dimensional **feature space** $\mathcal{H}$ (say p dimension), where

$$\mathbf{\Phi}_i = (\phi_1(\mathbf{x}_i), \cdots, \phi_p(\mathbf{x}_i))$$

  The map $\mathbf{\Phi}$ is called a **feature map**.
- Given $\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_n \in \mathcal{H}$ with $\sum_{i=1}^n \mathbf{\Phi}_1 = \mathbf{0}$, solve a linear PCA problem in feature space $\mathcal{H}$.
- Note that the dimension of $\mathcal{H}$ can be ∞, and in the second step, the data **MUST** be centered in $\mathcal{H}$ (can be done via a matrix trick).

Kernel PCA: Kernel trick

Suppose $\mathbf{\Phi}_i's$ are centered. Define the empirical covariance matrix:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{\Phi}_i \mathbf{\Phi}_i^T$$

The eigen-value and eigen-vector of $\mathbf{C}$ is given by $\mathbf{Cv} = \lambda \mathbf{v}$

Lemma 6 (Equivalent form of eigen-vector)

The eigen-vector and eigen-values can also be derived from solving n linear equations:

$$\langle \mathbf{\Phi}_i, \mathbf{Cv} \rangle = \lambda \langle \mathbf{\Phi}_i, \mathbf{v} \rangle, i = 1, \cdots, n$$

Or more compactly, define $\mathbf{\Phi}^T = (\mathbf{\Phi}_i, \cdots, \mathbf{\Phi}_n)$ *(thus* $\mathbf{\Phi} \in \mathbb{R}^{n \times p}$*)*

$$\langle \mathbf{\Phi}^T, \mathbf{Cv} \rangle = \lambda \langle \mathbf{\Phi}^T, \mathbf{v} \rangle$$

Kernel PCA: Kernel trick cont'd

Proof.

Note that

$$\mathbf{C}\mathbf{v} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{\Phi}_i \mathbf{\Phi}_i^T \mathbf{v} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{\Phi}_i \langle \mathbf{\Phi}_i, \mathbf{v} \rangle = \lambda \mathbf{v}$$

Thus each eigenvector is a linear combination of $\mathbf{\Phi}_i, i = 1, \cdots, n$. So there exists coefficients $\alpha_i, i = 1, \cdots, n$ s.t.

$$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{\Phi}_i$$

Kernel PCA: Kernel trick cont'd

Substituting $\mathbf{C}$ and the representation $\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{\Phi}_i$ in

$$\langle \mathbf{\Phi}^T, \mathbf{C}\mathbf{v} \rangle = \lambda \langle \mathbf{\Phi}^T, \mathbf{v} \rangle$$

we get that (vectorized version, you may want to verify it):

$$\langle \mathbf{\Phi}^T, \frac{1}{n} \mathbf{\Phi}^T \mathbf{\Phi} \langle \mathbf{\Phi}, \boldsymbol{\alpha} \rangle \rangle = \lambda \langle \mathbf{\Phi}^T, \langle \mathbf{\Phi}, \boldsymbol{\alpha} \rangle \rangle$$

where $\boldsymbol{\alpha}^T = (\alpha_1, \cdots, \alpha_n)$. Moreover,

$$LHS = \frac{1}{n} \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Phi}^T \boldsymbol{\alpha} = \lambda \mathbf{\Phi}\mathbf{\Phi}^T \boldsymbol{\alpha} = RHS$$

Kernel PCA: Kernel trick cont'd

Now let's define $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$.
Then $LHS = \mathbf{K}^2\boldsymbol{\alpha} = n\lambda\mathbf{K}\boldsymbol{\alpha} = RHS$, or as (why ?)

$$\mathbf{K}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}$$

Here $\mathbf{K}$ is the famous **Kernel Matrix** with elements denoted as $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Therefore, we don't need to define the mapping $\mathbf{\Phi}$ explicitly but just he inner product (kernel) $k(\mathbf{x}_i, \mathbf{x}_j)$.

- The principal component (or direction) in the feature space $\mathcal{H}$ is the eigenvector $\boldsymbol{\alpha}_{\text{max}}$ corresponding to the largest eigenvalue λ_{max} of the kernel matrix $\mathbf{K}$.

Kernel PCA: Normalization

- Recall that $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$, if we require $\|\mathbf{v}_i\|_2^2 = 1, i = 1, \cdots, n$, then, using the representation of $\mathbf{v}$ by $\boldsymbol{\alpha}$, we have

$$\begin{aligned} 1 &= \sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_{ij}\alpha_{ik}\langle \boldsymbol{\Phi}_j, \boldsymbol{\Phi}_k \rangle \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_{ij}\alpha_{ik}K_{jk} \\ &= \langle \boldsymbol{\alpha}_i, \mathbf{K}\boldsymbol{\alpha}_i \rangle \\ &= n\lambda\langle \boldsymbol{\alpha}_i, \boldsymbol{\alpha}_i \rangle \end{aligned}$$

  which determines the normalization for the vectors $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n$.

Kernel PCA: Centralization

If $\mathbf{K}$ is not centralized, then replace it with (Scholkopf, Smola and Muller, 1998)

$$\widetilde{\mathbf{K}} = \mathbf{HKH}$$

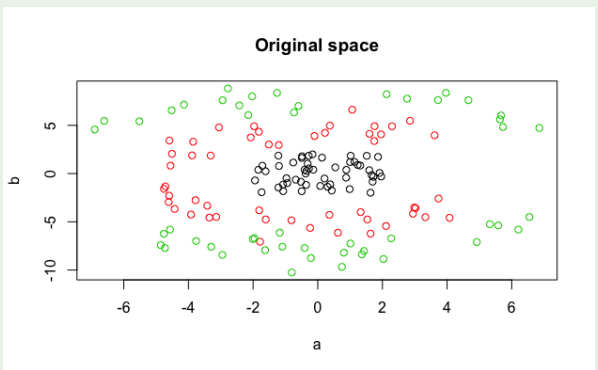
where $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{J}_n$ is called the centering matrix, $\mathbf{J}_n = \mathbf{1}_n\mathbf{1}_n^T$ is an $(n \times n)-$matrix of all ones.

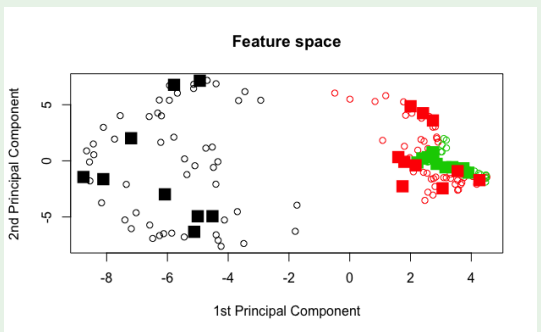
- Using this trick, the mapping $\mathbf{\Phi}_i$ in feature space $\mathcal{H}$ is centered.

Kernel PCA: Example with toy data

Left: original data.
Right: data in first two principal component space.
Square points: predicted new feature with new inputs.

Example 7 (Three clusters (Gaussian kernel))





Principal Curves: Intuition

Another way of linear reduction is called **Principal Curves**. First, I give intuitive definitions and then we run into formal ones.

- **Principle curves and surfaces** were proposed by Hastie, 1984; Hastie and Stuetzle, 1989.
- Suppose the data observed on **X** lie close to a smooth nonlinear manifold of low dimension.
- **Principle Curve** is a smooth one-dimensional **parameterized curve f** that passes through the "middle" of the data, regardless of whether the "middle" is a straight line or a nonlinear curve.

Principle Surface: A generalization of principle curve to a high dimensional curve.

Principle Curves: A Review of Geometry

[geometry] To give formal definition of principal curves, we need to review some concepts from geometry:

- **p dimension curve/surface in** $\mathbb{R}^k$ $(p < k)$:

$$\mathbf{f}(\lambda) = (f_1(\lambda), f_2(\lambda), \cdots, f_k(\lambda))^T$$

Where $\lambda = (\lambda_1, \cdots, \lambda_p)$ is the parameterization.

Example 8

The unit sphere in $\mathbb{R}^3$: $\{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 = 1\}$ is a two dimensional surface:

$$\mathbf{f}(\lambda) = (\cos\lambda_1 \sin\lambda_2, \sin\lambda_1 \sin\lambda_2, \cos\lambda_2)$$

Usually, it is convenient for us to assume $\mathbf{f} \in \mathcal{C}^\infty$.

Principle Curves: A Review of Geometry

For simplicity, let's consider the case $p = 1$ (or $\lambda \in \mathbb{R}$).

- $\mathbf{f}$ is **closed** of **periodic** if $\mathbf{f}(\lambda + \alpha) = \mathbf{f}(\lambda) \forall \lambda$.
- Velocity at λ: $v = \mathbf{f}'(\lambda)$
- Speed at λ: $\|v\| = \|\mathbf{f}'(\lambda)\|$
- Acceleration: $a = \mathbf{f}''(\lambda)$
- Curvature: $K(\lambda) = \|\mathbf{f}''(\lambda)\|$
- Circle of curvature (at λ): unit-speed circle with radius $r(\lambda) = \frac{1}{\|\mathbf{f}''(\lambda)\|}$
- Arc-length (from λ_0 to λ_1):

$$L(\mathbf{f}) = \int_{\lambda_0}^{\lambda_1} \|\mathbf{f}'(\lambda)\| d\lambda$$

When the speed of $\mathbf{f}(\lambda)$ is identically one, the arc-length is $\lambda_1 - \lambda_0$.

Principal Curves: Definition

Again, let's assume $\mathbf{f}(\lambda)$ is a 1-d parameterized curve in $\mathbb{R}^k$.

Definition 9 (Projection of $\mathbf{x}$ on $\mathbf{f}(\lambda)$)

Suppose the minimum (or infimum) is attained, the projection is defined to be

$$\mathbf{x_f} = \arg\min_{\mathbf{f}(\mu)} \|\mathbf{x} - \mathbf{f}(\mu)\|$$

Definition 10 (Projection index)

$$\lambda_{\mathbf{f}}(\mathbf{x}) = \sup_{\lambda}\{\lambda : \|\mathbf{x} - \mathbf{f}(\lambda)\|\} = \inf_{\mu}\|\mathbf{x} - \mathbf{f}(\mu)\|\}$$

We are taking sup because sometimes there are multiple choices of μ s.t. the infimum is attained. Note that $\lambda_{\mathbf{f}}(\mathbf{x})$ can be discontinuous.

Principle Curves: Definition cont'd

Definition 11 (Reconstruction error)

The reconstruction error is the expected squared distance between $\mathbf{X}$ (or its associated pdf) and $\mathbf{f}$:

$$D^2(\mathbf{X}, \mathbf{f}) = \mathbb{E}\{\|\mathbf{X} - \mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{X}))\|\}$$

Definition 12 (Principle curve)

The 1-d curve $\mathbf{f}(\lambda)$ is called a **principle curve** if the following holds:

$$\mathbf{f}(\lambda) = \mathbb{E}\{\mathbf{X}|\lambda_{\mathbf{f}}(\mathbf{X}) = \lambda\} \text{ for a.e. } \lambda$$

In this case, $\mathbf{f}(\lambda)$ is also said to be **self-consistent**. In fact, $\mathbf{f}(\lambda)$ minimizes the reconstruction error defined above (Hastie and Stuetzle, 1989).

Note that principle curves are measurable.

Principal Curves: Generalization to surfaces

Principle curve has a natural extension to **principal surfaces** in high dimensions (w.r.t. λ) (Hastie, 1984; LeBlanc and Tibshirani, 1994).

Definition 13 (Principal surface)

A principal surface satisfies the self-consistency property,

$$\mathbf{f}(\boldsymbol{\lambda}) = \mathbb{E}\{\mathbf{X}|\boldsymbol{\lambda}_{\mathbf{f}}(\mathbf{X}) = \boldsymbol{\lambda}\} \text{ for a.e. } \boldsymbol{\lambda}$$

Principal Curves: Projection-Expectation Algorithm

- Goal: estimate $\mathbf{f}$ from n observations $\{\mathbf{x}_i\}_1^n$.
- Idea: Empirical risk minimization (a.k.a. ERM principle in statistical learning theory)

$$D^2(\{\mathbf{x}_i\}, \hat{\mathbf{f}}) = \min_{\mathbf{f}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{x}_i))\|^2$$

- Solution: An algorithm iterating between **projection** (estimating λ given $\mathbf{f}$) and **expectation** (estimating $\mathbf{f}$ assuming a fixed λ).

Principal Curves: PE Algorithm cont'd

Definition 14 (Projection-Expectation Algorithm)

- Start with the initial curve (or line) $\mathbf{f}^{(0)}$. Then $\{\mathbf{x}_i\}$ are each projected onto this line, yielding the n points $\lambda_{\mathbf{f}^{(0)}}(\mathbf{x}_i) = \lambda_i^{(1)},\ i = 1, 2, \cdots, n.$
- The updated curve $\mathbf{f}^{(1)}$ is derived via the self-consistency property: $\mathbf{f}^{(1)}(\lambda_i^{(1)}) = \mathbb{E}\{\mathbf{X}|\lambda_{\mathbf{f}^{(0)}}(\mathbf{X}_i) = \lambda_i^{(1)}\}, i = 1, \cdots, n$
- Given the current principal curve $\mathbf{f}^{(k-1)}$, the k^{th} iteration consists of 2 steps :
  1. **Projection step**:

$$\lambda_{\mathbf{f}^{(k-1)}}(\mathbf{x}_i) = \lambda_i^{(k)},\ i = 1, 2, \cdots, n$$

  2. **Expectation step**:

$$\mathbf{f}^{(k)}(\lambda_i^{(k)}) = \mathbb{E}\{\mathbf{X}|\lambda_{\mathbf{f}^{(k-1)}}(\mathbf{X}_i) = \lambda_i^{(k)}\}, i = 1, \cdots, n$$

Principal Curves: PE Algorithm cont'd

- Problem: How to derive the expectation $\mathbf{f}^{(k)}(\lambda_i^{(k)}) = \mathbb{E}\{\mathbf{X}|\lambda_{\mathbf{f}^{(k-1)}}(\mathbf{X}_i) = \lambda_i^{(k)}\}$?
- Solution: estimate the expectation via **local averaging procedure** !
- More specifically, estimation of each coordinate of $\mathbf{f}^{(k)}(\lambda_i^{(k)})$ is accomplished using a **scatterplot smoother** (e.g., kernel, cubic spline, or locally weighted running-line smoother).
- Given the "final" estimation $(\hat{\lambda}_i, \hat{\mathbf{f}}_i), i = 1, \cdots, n$, the principal curve $\hat{\mathbf{f}}(\lambda)$ is then the polygon produced by joining up these n tuples (see example in next slide).
- **IMPORTANT**: Convergence of PE algorithm has **NOT** been proved. It may lead to poor "local" solution.

Principal Curves: A 2-D Example

Example 15 (Principal Curve in 2-D)

$\mathbf{X} \sim \mathcal{U}(-5, 5)$, $\mathbf{Y} = \sin \mathbf{X} + \frac{2}{\mathbf{X}}\mathcal{N}(0, 1)$, 100 data points in total.

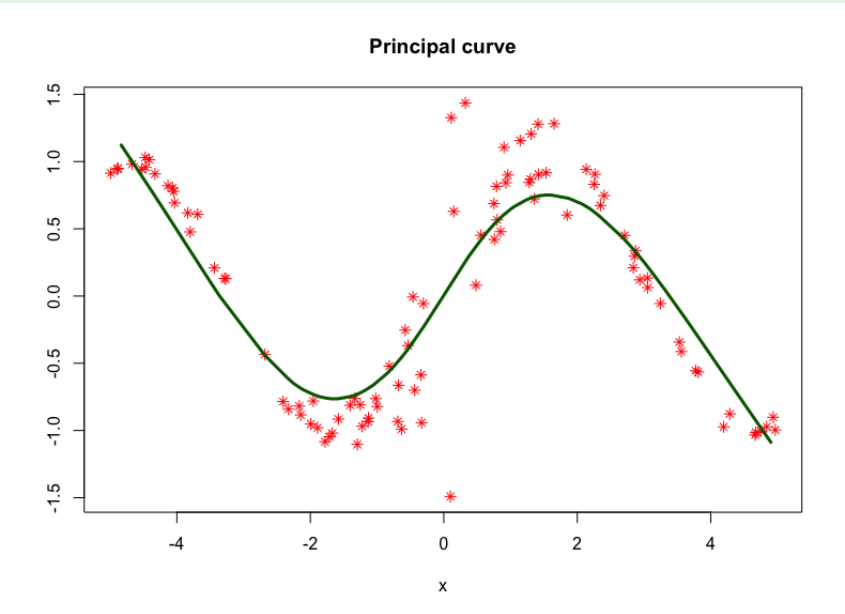


Figure: Principal Curve

Autoencoder in Deep Learning

- Peter Bickel: Imaging compression is a modern version of deriving sufficient statistic (Efron, 2016).

Example 16 (Autoencoder for MNIST)

1. **Autoencoder**: Each image of MNIST dataset has 784 elements.
2. Reduce each image to 10 dimensions.
3. Reconstruct each image via its unique 10-d vector.

- Generate images from 10-d vector directly (without reduction step): This is known as **GAN** (Goodfellow, 2014).
- See `http://cs231n.stanford.edu` for further information.

Functional PCA: Intuition

- **Intuition**: Data consisting of functions or curves.
- **Data**: Independent individuals may be recorded at different sets of time points.
- **Goal**: Characterize the main features of those curves.
- **Solution**: A functional version of PCA (e.g. Ramsay and Silverman, 1997)
- The vector-valued observations $\mathbf{X}_1, \cdots, \mathbf{X}_n$ are replaced by

$$X_1(t), \cdots, X_n(t), t \in [0, T]$$

- Each sample curve is considered to be an independent realization of a univariate stochastic process $X(t)$ with smooth mean function $\mathbb{E}\{X(t)\} = \mu(t)$ and covariance function

$$cov\{X(s), X(t)\} = \sigma(s, t)$$

Functional PCA: Karhunen-Loeve Expansion

By a spectral decomposition of the covariance function, we express σ as an orthogonal expansion in the L_2 sense in terms of its eigenvalues $\{\lambda_j\}$ and associated eigenfunctions $\{V_j(t)\}$, so that

$$\sigma(s,t) = \sum_{j=1}^{\infty} \lambda_j V_j(s) V_j(t)$$

where the eigenvalues quickly tend to zero and the first few eigenfunctions are slowly varying. A **random curve** can then be expressed as

$$X(t) = \mu(t) + \sum_{j=1}^{\infty} c_j V_j(t), \;\; c_j = \int [X(t) - \mu(t)] V_j(t) dt$$

This is the well-known **Karhunen-Loeve expansion** in stochastic processes. The coefficient c_j is called **the j^{th} functional PC score** with $\mathbb{E}(c_j) = 0$, $Var(c_j) = \lambda_j$, $\sum_j \lambda_j < \infty$ and $Cov(c_j, c_k) = 0$.

Probabilistic PCA: Model Set-up

- The formulation of PCA as a probabilistic model was proposed independently by Tipping and Bishop in 1997 and by Roweis in 1998.
- Probabilistic PCA can be expressed as a directed graph in which each observation $\mathbf{x}_i$ is associated with a value $\mathbf{z}_i$ of the latent variable.
- This is a special case of **Gaussian Linear Model**.

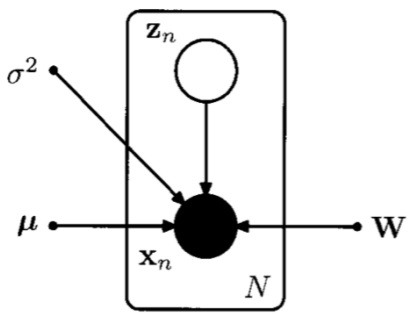


Figure: Bishop, 2006, pp574

Probabilistic PCA: Model Set-up cont'd

More formally, we write our model as

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{W}\mathbf{z} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\mu} \in \mathbb{R}^p, \mathbf{W} \in \mathbb{R}^{p \times q}$ are fixed and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I_q})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I_p})$, $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are independent.

- $q << p$, thus the dimension of $\mathbf{z}$ is much less than $\mathbf{y}$.
- $\mathbf{z}$ is known as the latent variable and $\boldsymbol{\epsilon}$ is the noise vector.
- **Goal**: estimate $\mathbf{W}$ and $\boldsymbol{\mu}$ so that we could recover $\mathbf{z}$ (principal components) from observation $\mathbf{y}$ (we assume σ) is known for simplicity.
- **Solution I**: Maximum likelihood estimation
- **Solution II**: EM Algorithm.

Probabilistic PCA: A Review of SVD

Before we go into MLE framework, let's take a quick review of **singular value decomposition**:

Definition 17 (Singular Value)

For any matrix $\mathbf{W} \in \mathbb{R}^{p \times q}$ (p¿q), its singular value is defined to be

$$\sigma(\mathbf{W}) = \lambda(\mathbf{W}^T\mathbf{W})$$

where $\lambda(\cdot)$ denotes the eigen-value and $\sigma(\cdot)$ denotes the singular value.

Probabilistic PCA: SVD cont'd

Theorem 18 (Singular Value Decomposition)

For any matrix $\mathbf{W} \in \mathbb{R}^{p \times q}$*, it admits the following decomposition:*

$$\mathbf{W} = \widetilde{\mathbf{U}}\widetilde{\mathbf{L}}\mathbf{V}^{\mathsf{T}} = \mathbf{U}\mathbf{L}\mathbf{V}^{\mathsf{T}}$$

where $\widetilde{\mathbf{U}} \in \mathbb{R}^{p \times p}, \mathbf{U} \in \mathbb{R}^{p \times q}, \mathbf{V} \in \mathbb{R}^{q \times q}$ *are all* **orthogonal matrices** *and* $\widetilde{\mathbf{L}} \in \mathbb{R}^{p \times q}, \mathbf{L} \in \mathbb{R}^{q \times q}$ *are* **diagonal matrices** *whose elements are known as* **singular values**. *Note that* $\widetilde{\mathbf{L}}$ *has* q *non-zero elements lying on the diagonal.*

We need both representations later in deriving MLE for $\mathbf{W}$ and σ^2.

MLE for Probabilistic PCA

- Knowing that $\mathbf{y}$ has normal distribution, derive its mean and variance (left as an exercise)

$$\mathbb{E}(\mathbf{Y}) = \boldsymbol{\mu}$$

$$Var(\mathbf{Y}) = \sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^T$$

- Write out log-likelihood function:

$$\log \mathbb{P}(\mathbf{y}_1, \cdots, \mathbf{y_n}|\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \sum_{i=1}^{n} \log \mathbb{P}(\mathbf{y}_i|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$$

$$= -\frac{N}{2}\left(p\log(2\pi) + \log\det \mathbf{C}_y + Tr(\mathbf{C}_y^{-1}\mathbf{S})\right)$$

where $\mathbf{C}_y = \sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^T$ and $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^T$.

MLE for Probabilistic PCA cont'd: 2

- A useful handbook for matrix derivatives: `https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`.
- Take derivative w.r.t. $\boldsymbol{\mu}$

$$-N\mathbf{C}_y^{-1}\boldsymbol{\mu} = \sum_{i=1}^{n} \mathbf{C}_y^{-1}\mathbf{y}_i = 0$$

$$\Rightarrow \hat{\boldsymbol{\mu}}_{MLE} = \frac{1}{N}\sum_{i=1}^{n} \mathbf{y}_i$$

- MLE of $\mathbf{W}$ is in general difficult to get but there is a closed form solution (Tipping and Bishop, 1999). However, in their original paper, I didn't figure out how they derive equation (A.13).
- On the other hand, I derived another equation similar to (A.13) which has the same solution. It is on the next slide.

MLE for Probabilistic PCA cont'd: 3

- Take derivative w.r.t. $\mathbf{W}$ via the Chain rule

$$\mathbf{S}\mathbf{C}_y^{-1}\mathbf{W} = \mathbf{W}$$

- **Problem**: how to derive the solution for $\mathbf{W}$?
- **Solution**: Use singular value decomposition !
- **Singular value decomposition**: Write $\mathbf{W}$ as

$$\mathbf{W} = \widetilde{\mathbf{U}}\widetilde{\mathbf{L}}\mathbf{V}$$

Then $\mathbf{C}_y^{-1}$ becomes

$$(\sigma^2\mathbf{I}_p + \widetilde{\mathbf{U}}\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^{\mathbf{T}}\widetilde{\mathbf{U}}^{\mathbf{T}})^{-1} = \widetilde{\mathbf{U}}(\sigma^2\mathbf{I}_p + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^{\mathbf{T}})^{-1}\widetilde{\mathbf{U}}^{\mathbf{T}}$$

MLE for Probabilistic PCA cont'd: 4

- Using the new representation of $\mathbf{C}_y^{-1}$:

$$\mathbf{S}\mathbf{C}_y^{-1}\mathbf{W} = \mathbf{S}\widetilde{\mathbf{U}}(\sigma^2\mathbf{I}_p + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^T)^{-1}\widetilde{\mathbf{L}}\mathbf{V}^T$$

- Now here is the other form of SVD come in:

$$\mathbf{W} = \widetilde{\mathbf{U}}\widetilde{\mathbf{L}}\mathbf{V}^{\mathbf{T}} = \mathbf{U}\mathbf{L}\mathbf{V}^{\mathbf{T}} \Rightarrow$$

$$\widetilde{\mathbf{U}}(\sigma^2\mathbf{I}_p + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^T)^{-1}\widetilde{\mathbf{L}} = \mathbf{U}(\sigma^2\mathbf{I}_q + \mathbf{L}\mathbf{L}^T)^{-1}\mathbf{L}$$

For details, see next slide.

MLE for Probabilistic PCA cont'd: 5

- Here I made use of the fact that the inverse of the following matrix

$$(\sigma \mathbf{I}_p) + \begin{bmatrix} l_1^2 & & \cdots & \\ & l_2^2 & & \\ \cdots & & & \\ & & l_q^2 & \\ & & & \mathbf{0} \end{bmatrix}$$

is nothing but

$$\begin{bmatrix} \frac{1}{l_1^2+\sigma^2} & & & & \\ & \cdots & & & \\ & & \frac{1}{l_q^2+\sigma^2} & & \\ & & & \frac{1}{\sigma^2} & \\ & & & & \cdots \end{bmatrix}$$

where l_i's are diagonals of $\mathbf{L}$ (singular values).

MLE for Probabilistic PCA cont'd: 6

- However, note that we could cut off all diagonals below the q^{th} row (or column) since all elements below q^{th} row of $\widetilde{\mathbf{L}}$ is 0.
- This is gives us the following matrix

$$\begin{bmatrix} \frac{1}{l_1^2+\sigma^2} & & \\ & \cdots & \\ & & \frac{1}{l_q^2+\sigma^2} \end{bmatrix}$$

But this is just $(\sigma^2 \mathbf{I}_q + \mathbf{L}\mathbf{L}^T)^{-1}$.

MLE for Probabilistic PCA cont'd: 7

Recall slide 37 and 38, the equation now becomes

$$\mathbf{SU}(\sigma^2\mathbf{I}_q + \mathbb{L}\mathbb{L}^T)^{-1}\mathbf{L}\mathbf{V}^T = \mathbf{U}\mathbf{L}\mathbf{V}^T$$

Since $\mathbf{V}$ and $\mathbf{L}$ is non-singular so they can be cancelled out on both sides. Therefore, the equation finally turns into

$$\mathbf{SU} = \mathbf{U}(\sigma^2\mathbf{I}_q + \mathbf{L}\mathbf{L}^T)$$

This is equivalent to solve q eigen-value problems:

$$(\sigma^2 + l_j^2)\mathbf{u}_j = \mathbf{S}\mathbf{u}_j = \lambda_j\mathbf{u}_j, j = 1, \cdots, q$$

where λ_j's are the eigenvalues of sample covariance matrix $\mathbf{S}$.

MLE for Probabilistic PCA cont'd: 8

The solution of previous equations is

$$l_j = (\lambda_j - \sigma^2)^{-\frac{1}{2}}, j = 1, \cdots, q$$

Thus,

$$\hat{\mathbf{W}}_{MLE} = \hat{\mathbf{U}}(\hat{\mathbf{L}}\hat{\mathbf{L}}^{\mathbf{T}} - \sigma \mathbf{I}_q^2)^{\frac{1}{2}}\mathbf{V}$$

Now since the distribution of $\mathbf{Y}$ does not depend on the rotation of $\mathbf{W}$ (*), $\mathbf{V}$ is chosen to be any orthogonal matrix.

- (*) That is, the mean and variance of $\mathbf{Y}$ stays the same if we replace $\mathbf{W}$ as $\mathbf{WR}$ for any rotation (orthogonal matrix) $\mathbf{R}$.

High dimensional PCA: Computational difficulty

Now suppose the dimension of each data point is high (larger the the sample size). The standard PCA will solve the following eigen-vector equation:

$$\frac{1}{n}\mathbf{X}^T\mathbf{X}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

Where here $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the (centered) data matrix. Since we know there are (at least) $p - n - 1$ eigen-values are 0, we could pre-multiply both sides by $\mathbf{X}$ to give

$$\frac{1}{n}\mathbf{X}^T\mathbf{X}\mathbf{X}\mathbf{v}_i = \lambda_i \mathbf{X}\mathbf{v}_i$$

High dimensional PCA: Computational difficulty cont'd

Define $\mathbf{u}_i = \mathbf{X}\mathbf{v}_i$, we obtain

$$\frac{1}{n}\mathbf{X}\mathbf{X}^T\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

Note that solving this eigen-vector problem only needs $O(n^3)$ instead of $O(p^3)$ for solving the standard one. Besides, there is a relationship between $\mathbf{v}_i$ and $\mathbf{u}_i$:

$$\mathbf{v}_i = \frac{1}{(n\lambda_i)^{1/2}}\mathbf{X}^T\mathbf{u}_i$$

In fact, the technique used above can be derived from a higher point of view known as **Woodbury identity** (see next slide). Such techniques are common in machine learning to reduce the computation of algorithms.

High dimensional PCA: Woodbury identity

Lemma 19 (Woodbury)

Provided all matrix multiplications are compatible, we have:

$$(\mathbf{A}+\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$$

Proof.

Left as an exercise.

Hint 1: ETS the inverse of LHS multiplies RHS is the identity.

Hint 2: $B(D+CA^{-1}B)^{-1}=BD^{-1}D(D+CA^{-1}B)^{-1}$ □

High dimensional PCA: Another useful matrix lemma

Lemma 20 (Another Useful Formula)

Provided all matrix multiplications are compatible, we have:

$$(\mathbf{P}^{-1} + \mathbf{B}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^{T}(\mathbf{B}\mathbf{P}\mathbf{B}^{\mathbf{T}} + \mathbf{R})^{-1}$$

It can be verified by right multiplying both sides by $(\mathbf{B}\mathbf{P}\mathbf{B}^{T} + \mathbf{R})$.

Example 21 (Dual Representation of Ridge Regression)

For ridge regression, we have

$$\widehat{\beta} = (\mathbf{X}^{T}\mathbf{X} + \lambda\mathbf{I}_n)^{-1}\mathbf{X}^{T}\mathbf{y}$$

By the above lemma, the dual representation is

$$\widehat{\beta} = \mathbf{X}^{T}(\mathbf{X}\mathbf{X}^{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{y}$$

Exact Distribution of the Eigenvalues

Before we dive into the high dimensional case where $p >> n$, let's recall a classic result in multivariate analysis.

Definition 22 (Wishart Distribution)

Suppose $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p),\ i = 1, \cdots, n$ and $n > p$, then

$$\mathbf{X}\mathbf{X}^T \sim \mathcal{W}_p(n, \mathbf{I}_p)$$

where $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_p)$ and $\mathcal{W}_p(n, \mathbf{I}_p)$ is called **the Wishart distribution** with parameter n and $\mathbf{I}_p$, and the pdf is

$$p_{\mathbf{X}}(\mathbf{X}) = \frac{(\det \mathbf{X})^{(n-p-1)/2} e^{-Tr(\mathbf{X})/2}}{2^{\frac{np}{2}} \Gamma_p(\frac{n}{2})}$$

Exact Distribution of the Eigenvalues cont'd

Theorem 23 (Dist. of the Eigenvalues of Wishart Dist.)

Now suppose $\mathbf{X}\mathbf{X}^T \sim \mathcal{W}_p(n, \mathbf{I}_p)$*, then the exact distribution of the eigenvalues of this random matrix has the form:*

$$p(\lambda_1, \cdots, \lambda_p) = c_{p,n} \prod_{j=1}^{p} [w(\lambda_j)]^{1/2} \prod_{j<k} (\lambda_j - \lambda_k)$$

where $w(x) = x^{n-r-1}e^{-x}$*,* $c_{p,n}$ *is the normalizing constant and* $\prod_{j<k}(\lambda_j - \lambda_k)$ *is known as the* **Vandermonde determinant**

Proof: Hsu, 1939.

e.g. Chapter 11 section 5 and chapter 13 section 3 of Anderson, 1984. □

Empirical Spectral Distribution

If p is larger than n, the previous theorem fails. However, random matrix theory gives us the asymptotic results:

Theorem 24 (Marcenko-Pastur Law)

Suppose $\mathbf{X}\mathbf{X}^T \sim \mathcal{W}_p(n, \mathbf{I}_p)$. *Define the* **empirical spectral distribution**

$$G_p(k) = \frac{1}{p}\#\{\hat{\lambda}_j \leq k\}$$

If $\frac{p}{n} \to \gamma \in (0, \infty)$, *then,* $G_r(k) \to G(k)$ *almost surely, where the limiting distribution* $G(k)$ *has density* $g(k) = G'(k)$*:*

$$g(k) = \frac{\sqrt{(b_+ - k)(k - b_-)}}{2\pi\gamma k}, b_\pm = (1 \pm \sqrt{\gamma})^2$$

This is the famous **Marcenko-Pastur Law** *in statistical physics.*

Sparse PCA

- Due to the previous theorem, we know that in high dimensional case, neither the eigen-values nor eigen-vectors of sample covariance matrix will be consistent. Therefore, we need to modify the standard PCA in such case. This is the so-called **Sparse PCA**.
- This part is quite technical (concentration inequalities) so we can omit it.

END !

Next time:
Radon-Nikodym Theorem,
Conditional Expectation
and Martingale !