Mercer's Thm. & Kernel PCA

Recap

Principle Component Analysis identifies directions of maximal variance in data

$$x_1, x_2, \ldots, x_m \in \mathbb{R}^n$$

by diagonalizing the (sample) covariance

$$C = \frac{1}{m-1} \sum_{j=1}^{m} (x_j - \mu)(x_j - \mu)^T$$

where $\mu = \frac{1}{m} \sum_{j=1}^{m} x_j$ is the sample mean.

This makes PCA a useful tool for many foundational tasks in data-science:

- ⇒ Model identification & reduction
- ⇒ De-noising and filtering
- ⇒ Clustering & Classification
- ⇒ Pre/Post processing for data

Many extensions and adaptations of PCA to various application domains.

PCA: Nonlinear Effects

A fundamental limitation of PCA is that it only captures linear trends in data. It diagonalizes the covariance matrix, but is blind to higher-order statistical trends in data.

To incorporate higher-order correlations, one might consider "adding" new variables

For example, $x_1 = \begin{pmatrix} x_1^{(1)} \\ x_1^{(2)} \end{pmatrix}, \dots, x_m = \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix}$

add $\Rightarrow$ $x_k^{(3)} = (x_k^{(1)})^2$, $x_k^{(4)} = x_k^{(1)} x_k^{(2)}$, $x_k^{(5)} = (x_k^{(2)})^2$

(+) The new 5×5 covariance matrix now contains higher-order statistical moments of the data.

(−) However, the size of the covariance matrix grows. For high-dimensional data, PCA w/augmented data is intractable.

We can write down this idea in a slightly more general setting and then work out how to do computation efficiently.

Let $\phi: \mathbb{R}^n \to \mathbb{R}^d$ be a dictionary of features that "lift" the data into a higher-dimensional space ($d \gg n$).

$$\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_d(x)]^T$$

In the new space, the mean and covariance of the mapped data is

$$\mu = \sum_{j=1}^{m} \phi(x_j), \quad C = \frac{1}{m-1} \sum_{j=1}^{m} (\phi(x_j)-\mu)(\phi(x_j)-\mu)^T$$

We can run PCA in the new, higher dimensional feature space,

$$C = U \Lambda U^T \implies \psi(x_j) = U^T \phi(x_j)$$

Diagonalize dual covariance matrix for features

principle components of mapped data in feature space

Kernel PCA

To get around the "curse" of dimension $d \gg n$, Kernel PCA computes the leading Principle Components of the data in feature space without ever manipulating the d-dimensional features directly!

In particular, the $d \times d$ covariance matrix C is never formed explicitly.

$$C = \frac{1}{m-1} \sum_{j=1}^{m} (\phi(x_j) - \mu)(\phi(x_j) - \mu)^T$$

$$= \frac{1}{m-1} \underset{d \times m}{B} \underset{m \times d}{B^T} \qquad \text{rank } m \text{ matrix}$$

To compute nonzero eigenpairs of C:

$$\lambda \neq 0 \qquad Cu = \lambda u \iff \frac{1}{m-1} B^T B v = \lambda v$$

$$u = \frac{1}{\sqrt{m-1}} B v$$

We only need the $m \times m$ matrix $B^T B$.

To compute the principle components, we do not even need u explicitly!

$$u^T(\phi(x_j)-\mu) = \frac{1}{\sqrt{\lambda_{m}}} v^T B^T(\phi(x_j)-\mu)$$

$$= \frac{1}{\sqrt{\lambda_{m}}} v^T (B^T B)_{j^{th}\,col}$$

So we can recover principle component of data purely in terms of $B^T B$.

The Kernel Matrix

To avoid working in d-dimensional feature space, we can frame PCA entirely in terms of the Gram matrix

$$(B^T B)_{ij} = \phi^T(x_i)\,\phi(x_j)$$

$$= \sum_{k=1}^{d} \phi_k(x_j)\,\phi_k(x_i)$$

We can associate this with a kernel

$$K(x,y) = \sum_{k=1}^{d} \varphi_k(x)\varphi_k(y)$$

So to do PCA in high-dim feature space we only need to be able to compute entries of $m \times m$ Kernel matrix and work w/ m-dimensional vectors.

Mercer's Theorem

Mercer's theorem provides an implicit characterization of the dictionary/feature map by a continuous, self-adjoint, semi-definite matrix. Spectral decomp.

$$K(x,y) = \sum_{n=1}^{\infty} \lambda_n u_n(x)\overline{u_n(y)}$$

converges pointwise, absolutely & uniformly

So feature map for Mercer Kernel is

$$\varphi_n(x) = \sqrt{\lambda_n}\, u_n(x) \qquad n = 1, 2, 3, \ldots$$