Kernel Principal Component Analysis

Alex Beeny (abeeny@siue.edu)

Spring 2024

Contents

Abstract

Principal component analysis (PCA) and kernel methods are tools often used in data science. The underlying theory of these tools depend on the properties of a special type of Hilbert space called a reproducing kernel Hilbert space (RKHS). This paper explores the essence of RKHSs using data science examples, in particular, PCA and kernel PCA. When kernel methods are applied to PCA, we can analyze nonlinear data in a high-dimensional feature space with some nice properties.

1 Introduction

In linear regression, the equation of a line $y = a_0 + a_1x$ is used to model observations based on training data. Here, the input variable x is used to predict the response variable y. The parameters a_0 and a_1 are chosen such that

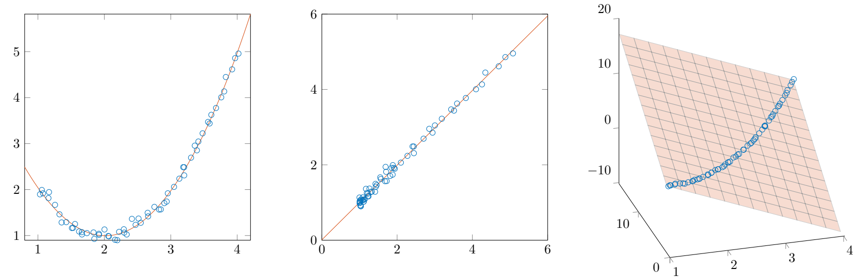

Figure 1: Plotting y against x (left) and the derived feature $z = \phi(x)$ (middle). The 3D plot (right) graphs the output y against x and x^2

the residual error[1] is minimized. It seems natural to model data using polyomial equations in a similar way, that is, determine $a_0, a_1, \ldots, a_n$ such that

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \tag{1}$$

minimizes the residual error.

In multiple linear regression, the equation of a hyperplane

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \tag{2}$$

is used to predict y using inputs $x_1, x_2, \ldots x_n$. It follows that these observations are points in $n + 1$ dimensions.

Example 1.1. Suppose we have a set of points $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{k}$ in $\mathbb{R}^2$. Using polynomial regression, we fit the model $y \sim x + x^2$. We can derive a new variable from x to get

$$z = \phi(x) = a_0 + a_1 x + a_2 x^2.$$

By transforming our original data, the quadradic relationship between x and y can be viewed as a linear relationship between z and y. This shows that polynomial regression is just a special case of multiple regression where each power of x is treated as a separate dimension. See Figure 1

Here, we make the distinction between different kinds of input variables. We say that x is an **attribute** and that z is a **feature**.

2 Principal Component Analysis

- describe the goal of PCA
- PCA as a rotation and rescaling
- show PCA minimizes projection residuals

[1] The residual for a given observation $\left(x^{(i)}, y^{(i)}\right)$ is $\left|y^{(i)} - \left(a_0 + a_1 x^{(i)}\right)\right|$.

- PCA algorithm
- PCA example (old)

Example 2.1. Consider the following matrix

$$A = \begin{bmatrix} 5 & 3 & 6 & 7 & 6 \\ 4 & 5 & 7 & 1 & 3 \\ 5 & 7 & 6 & 1 & 0 \\ 6 & 10 & 12 & 12 & 11 \\ 9 & 10 & 12 & 13 & 9 \end{bmatrix}.$$

The column means are $\mu = [5.8, 7, 8.6, 6.8, 5.8]$. Then the mean-centered data becomes

$$X = A - \mu = \frac{1}{5}\begin{bmatrix} -4 & -20 & -13 & 1 & 1 \\ -9 & -10 & -8 & -29 & -14 \\ -4 & 0 & -13 & -29 & -29 \\ 1 & 15 & 17 & 26 & 26 \\ 16 & 15 & 17 & 31 & 16 \end{bmatrix}.$$

The covariance matrix is

$$C = X^T X = \frac{1}{5}\begin{bmatrix} 74 & 85 & 93 & 179 & 104 \\ 85 & 190 & 170 & 225 & 150 \\ 93 & 170 & 196 & 313 & 238 \\ 179 & 225 & 313 & 664 & 484 \\ 104 & 150 & 238 & 484 & 394 \end{bmatrix}.$$

Diagonalizing C gives

$$V = \begin{bmatrix} 0.1888 & -0.2020 & -0.6366 & 0.5495 & -0.4651 \\ 0.2755 & -0.7886 & 0.1472 & -0.4502 & -0.2791 \\ 0.3606 & -0.3464 & 0.3128 & 0.5836 & 0.5582 \\ 0.6979 & 0.2522 & -0.4422 & -0.3707 & 0.3411 \\ 0.5209 & 0.3922 & 0.5288 & 0.1316 & -0.5271 \end{bmatrix},$$

$$D = \begin{bmatrix} 264.8458 & 0 & 0 & 0 & 0 \\ 0 & 27.9766 & 0 & 0 & 0 \\ 0 & 0 & 9.3198 & 0 & 0 \\ 0 & 0 & 0 & 1.4579 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

If we keep all 5 principal component vectors, then $V_5 = V$ and the projection of X along V is

$$P = XV = \begin{bmatrix} -1.9469 & 4.3453 & -0.8756 & -0.2039 & 0 \\ -6.9742 & -0.0660 & 1.4352 & 0.7590 & 0 \\ -8.1577 & -2.6752 & -0.8063 & -0.5704 & 0 \\ 8.4282 & -0.2330 & 1.8282 & -0.4996 & 0 \\ 8.6507 & -1.3711 & -1.5815 & 0.5149 & 0 \end{bmatrix}.$$

Here, the last column of P is the zero vector because the last eigenvalue of C is zero[2]. To perfectly reconstruct A, we need $k = 4$ principal components and the row vector μ

$$A = PV^T + \mu = PV_4^T + \mu.$$

If we use $k = 3$ principal components, then the projection of X onto V_3 is

$$P = XV_3 = \begin{bmatrix} -1.9469 & 4.3453 & -0.8756 \\ -6.9742 & -0.0660 & 1.4352 \\ -8.1577 & -2.6752 & -0.8063 \\ 8.4282 & -0.2330 & 1.8282 \\ 8.6507 & -1.3711 & -1.5815 \end{bmatrix}$$

and A is approximately reconstructed by

$$A \approx PV_3^T + \mu = \begin{bmatrix} 5.1 & 2.9 & 6.1 & 6.9 & 6.0 \\ 3.6 & 5.3 & 6.6 & 1.3 & 2.9 \\ 5.3 & 6.7 & 6.3 & 0.8 & 0.1 \\ 6.3 & 9.8 & 12.3 & 11.8 & 11.1 \\ 8.7 & 10.2 & 11.7 & 13.2 & 8.9 \end{bmatrix}.$$

We can compute the reconstruction error using

$$E_k = \|A - (PV_k^T + \mu)\|_F,$$

where $\|\cdot\|_F$ is the Frobenius norm. By the SVD, we have $X = USV^T$, where $S = \sqrt{D}$. So, the projection of X onto V_k is

$$P = XV_k = US_k,$$

where S_k is the diagonal matrix of the first k singular values. Then the reconstruction error becomes

$$\begin{aligned} \|A - (PV_k^T + \mu)\|_F &= \|(A - \mu) - PV_k^T\|_F \\ &= \|X - PV_k^T\|_F \\ &= \|USV^T - US_kV^T\|_F \\ &= \|U(S - S_k)V^T\|_F \\ &= \|S - S_k\|_F \\ &= \sigma_k + \sigma_{k+1} + \cdots + \sigma_p. \end{aligned}$$

Hence,

$$E_3 = \sigma_3 + \sigma_4 = \sqrt{1.4579} + 0 = 1.2074.$$

[2] Since we subtracted the column means from a square matrix A, the dimension of the row space was reduced to 4.

3 Reproducing Kernel Hilbert Space

4 Kernel PCA

PCA works by computing vector projections using the dot product. This is the typical inner product for $\mathbb{R}^n$ and the resulting basis is orthogonal. The idea behind kernel PCA is to replace the dot product with a kernel function.

Definitions.

- inner product and inner product space
- induced norm, normed space, Banach space
-

5 Conclusion

A Linear Algebra

B Riesz Representation Theorem

C Mercer's Theorem

D Code

References

[1] Peter Bartlett. CS 281B / Stat 241B Statistical Learning Theory. Lecture notes, Spring 2008.

[2] Peter Bartlett. CS 281B / Stat 241B Statistical Learning Theory. Lecture notes, Spring 2014.

[3] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3), jun 2008.

[4] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[5] The MathWorks Inc. Statistics and machine learning toolbox, 2022.

[6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2012.

[7] Cynthia Rudin. Intuition for the Algorithms of Machine Learning. Lecture notes, 2023.

[8] Walter Rudin. *Real and Complex Analysis*. Higher Mathematics Series. McGraw-Hill Education, 1987.

[9] Cosma Rohilla Shalizi. Advanced Data Analysis from an Elementary Point of View. Draft textbook, 2021.

[10] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.