Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 7: Dimensionality Reduction

The curse of (attribute) dimensionality

- Data may have large number of attributes
  - Data matrix with many columns d

$$\mathbf{D} = \begin{array}{c|cccc} & X_1 & X_2 & \cdots & X_d \\ \hline \mathbf{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \mathbf{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{array}$$

  - Challenges
    - Models have large number of parameter
      - d(d-1)/2 variances / covariances in a Gaussian model
      - Cumbersome, more difficult to estimate well
  - Dimension reduction
    - Identify small number of attributes (or combinations of attributes) that account for most of the data variation
    - Other dimensions of the data are viewed as noise,

Dimensionality Reduction

The goal of dimensionality reduction is to find a lower dimensional representation of the data matrix $\boldsymbol{D}$ to avoid the curse of dimensionality.

Given $n \times d$ data matrix, each point $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ is a vector in the ambient d-dimensional vector space spanned by the d standard basis vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$.

Given any other set of d orthonormal vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d$ we can re-express each point $\boldsymbol{x}$ as

$$\boldsymbol{x} = a_1\boldsymbol{u}_1 + a_2\boldsymbol{u}_2 + \cdots + a_d\boldsymbol{u}_d$$

where $\boldsymbol{a} = (a_1, a_2, \ldots, a_d)^T$ represents the coordinates of $\boldsymbol{x}$ in the new basis. More compactly:

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{a}$$

where $\boldsymbol{U}$ is the $d \times d$ orthogonal matrix, whose ith column comprises the ith basis vector $\boldsymbol{u}_i$. Thus $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$, and we have

$$\boldsymbol{a} = \boldsymbol{U}^T\boldsymbol{x}$$

Optimal Basis: Projection in Lower Dimensional Space

There are potentially infinite choices for the orthonormal basis vectors. Our goal is to choose an *optimal* basis that preserves essential information about $\boldsymbol{D}$.

We are interested in finding the optimal r-dimensional representation of $\boldsymbol{D}$, with $r \ll d$. Projection of $\boldsymbol{x}$ onto the first r basis vectors is given as

$$\boldsymbol{x}' = a_1\boldsymbol{u}_1 + a_2\boldsymbol{u}_2 + \cdots + a_r\boldsymbol{u}_r = \sum_{i=1}^{r} a_i\boldsymbol{u}_i = \boldsymbol{U}_r\boldsymbol{a}_r$$

where $\boldsymbol{U}_r$ and $\boldsymbol{a}_r$ comprises the r basis vectors and coordinates, respv. Also, restricting $\boldsymbol{a} = \boldsymbol{U}^T\boldsymbol{x}$ to r terms, we have

$$\boldsymbol{a}_r = \boldsymbol{U}_r^T\boldsymbol{x}$$

The r-dimensional projection of $\boldsymbol{x}$ is thus given as:

$$\boldsymbol{x}' = \boldsymbol{U}_r\boldsymbol{U}_r^T\boldsymbol{x} = \boldsymbol{P}_r\boldsymbol{x}$$

where $\boldsymbol{P}_r = \boldsymbol{U}_r\boldsymbol{U}_r^T = \sum_{i=1}^{r} \boldsymbol{u}_i\boldsymbol{u}_i^T$ is the *orthogonal projection matrix* for the subspace spanned by the first r basis vectors.

Optimal Basis: Error Vector

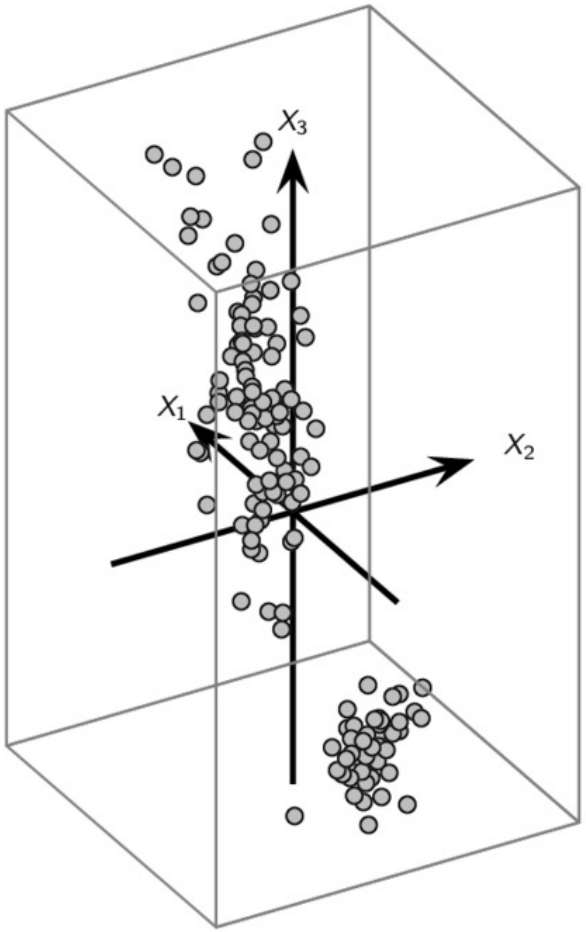
Given the projected vector $\boldsymbol{x}' = \boldsymbol{P}_r \boldsymbol{x}$, the corresponding *error vector*, is the projection onto the remaining $d - r$ basis vectors

$$\boldsymbol{\epsilon} = \sum_{i=r+1}^{d} a_i \boldsymbol{u}_i = \boldsymbol{x} - \boldsymbol{x}'$$

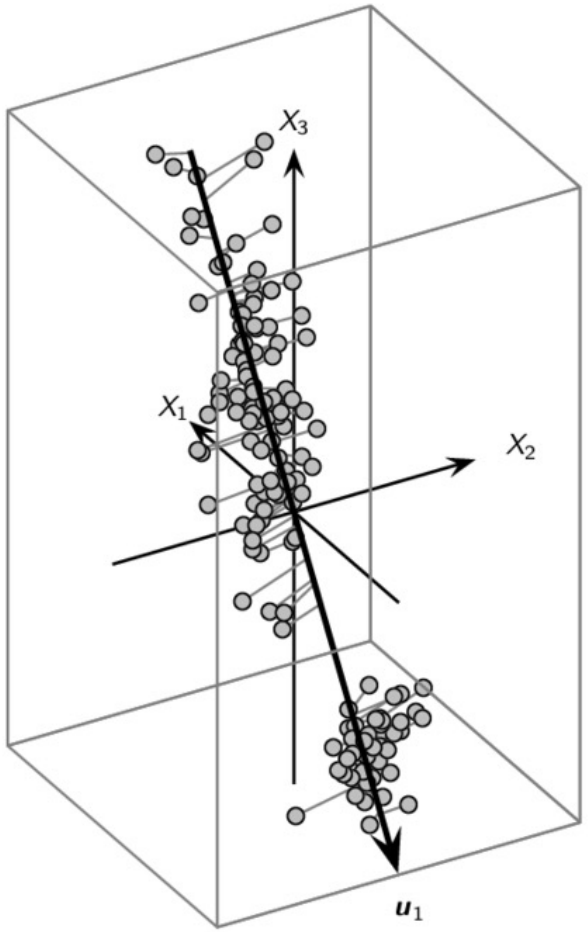
The error vector $\boldsymbol{\epsilon}$ is orthogonal to $\boldsymbol{x}'$.

The goal of dimensionality reduction is to seek an r-dimensional basis that gives the best possible approximation $\boldsymbol{x}'_i$ over all the points $\boldsymbol{x}_i \in \boldsymbol{D}$. Alternatively, we seek to minimize the error $\boldsymbol{\epsilon}_i = \boldsymbol{x}_i - \boldsymbol{x}'_i$ over all the points.

Iris Data: Optimal One-dimensional Basis

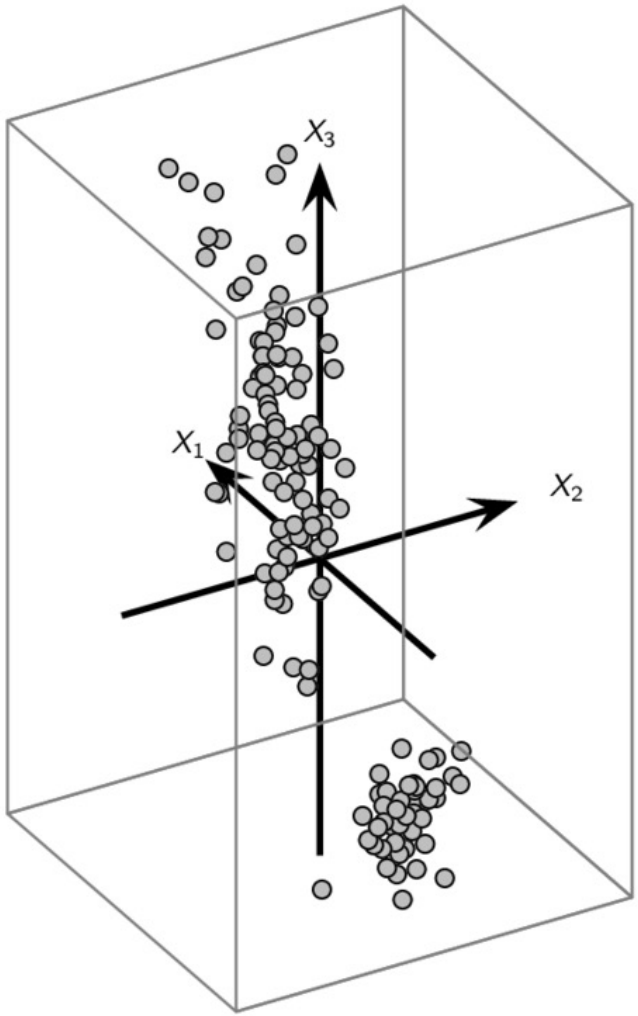


Iris Data: 3D

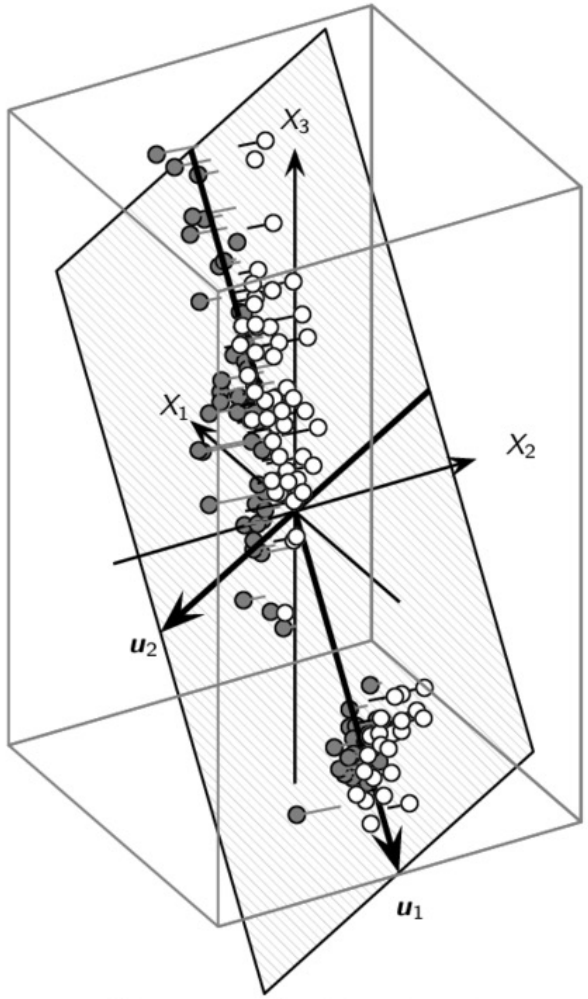


Optimal 1D Basis

Iris Data: Optimal 2D Basis



Iris Data (3D)



Optimal 2D Basis

Principal Component Analysis

Principal Component Analysis (PCA) is a technique that seeks a r-dimensional basis that best captures the variance in the data.

The direction with the largest projected variance is called the first principal component.

The orthogonal direction that captures the second largest projected variance is called the second principal component, and so on.

The direction that maximizes the variance is also the one that minimizes the mean squared error.

Principal Component: Direction of Most Variance

We seek to find the unit vector $\boldsymbol{u}$ that maximizes the projected variance of the points. Let $\boldsymbol{D}$ be centered, and let Σ be its covariance matrix.

The projection of $\boldsymbol{x}_i$ on $\boldsymbol{u}$ is given as

$$\boldsymbol{x}_i' = \left(\frac{\boldsymbol{u}^T\boldsymbol{x}_i}{\boldsymbol{u}^T\boldsymbol{u}}\right)\boldsymbol{u} = (\boldsymbol{u}^T\boldsymbol{x}_i)\boldsymbol{u} = a_i\boldsymbol{u}$$

$a_i = u^T x_i$

Across all the points, the projected variance along $\boldsymbol{u}$ is

projected mean is 0 because **D** is centered

$$\sigma_{\boldsymbol{u}}^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu_{\boldsymbol{u}})^2 = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{u}^T\left(\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u} = \boldsymbol{u}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{u} = \boldsymbol{u}^T\Sigma\boldsymbol{u}$$

We have to find the optimal basis vector $\boldsymbol{u}$ that maximizes the projected variance $\sigma_{\boldsymbol{u}}^2 = \boldsymbol{u}^T\Sigma\boldsymbol{u}$, subject to the constraint that $\boldsymbol{u}^T\boldsymbol{u} = 1$. The maximization objective is given as

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^T\Sigma\boldsymbol{u} - \alpha(\boldsymbol{u}^T\boldsymbol{u} - 1)$$

α is Lagrange Multiplier used in constrained optimization.
See https://en.wikipedia.org/wiki/Lagrange_multipliers

Principal Component: Direction of Most Variance

Given the objective $\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^T\Sigma\boldsymbol{u} - \alpha(\boldsymbol{u}^T\boldsymbol{u} - 1)$, we solve it by setting the derivative of $J(\boldsymbol{u})$ with respect to $\boldsymbol{u}$ to the zero vector, to obtain

$$\frac{\partial}{\partial \boldsymbol{u}}\left(\boldsymbol{u}^T\Sigma\boldsymbol{u} - \alpha(\boldsymbol{u}^T\boldsymbol{u} - 1)\right) = 0$$

$$\text{that is, } 2\Sigma\boldsymbol{u} - 2\alpha\boldsymbol{u} = 0$$

$$\text{which implies } \boxed{\Sigma\boldsymbol{u} = \alpha\boldsymbol{u}}$$

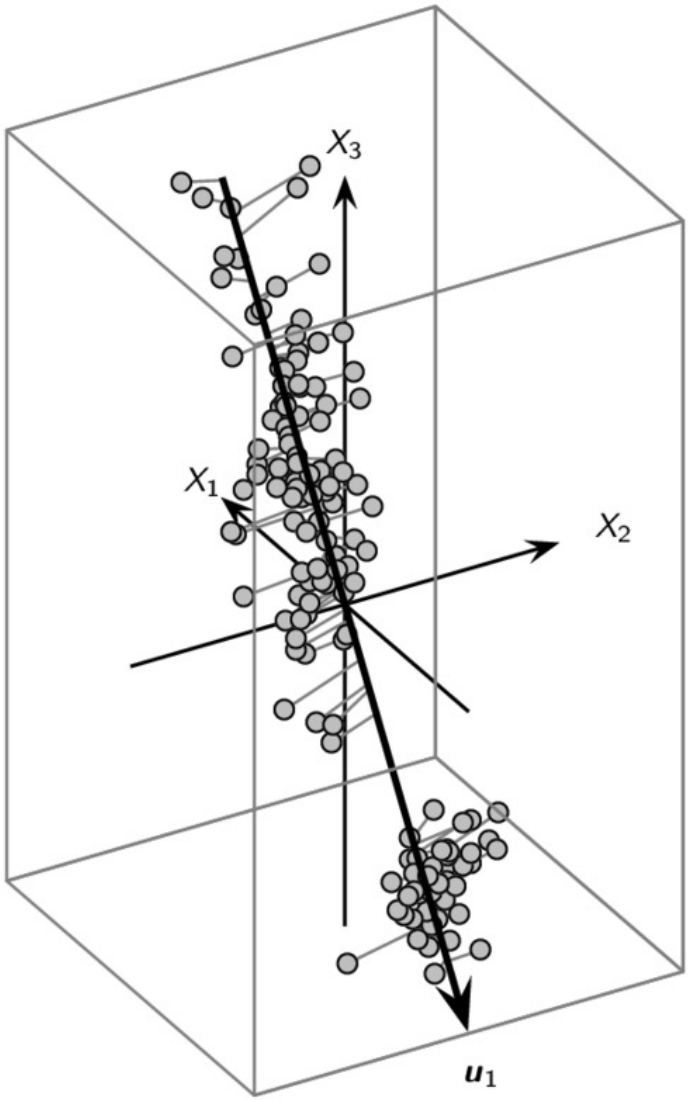
Thus α is an eigenvalue of the covariance matrix Σ, with the associated eigenvector $\boldsymbol{u}$.

Taking the dot product with $\boldsymbol{u}$ on both sides, we have

$$\sigma_{\boldsymbol{u}}^2 = \boldsymbol{u}^T\Sigma\boldsymbol{u}\boldsymbol{u}^T\alpha\boldsymbol{u} = \alpha\boldsymbol{u}^T\boldsymbol{u} = \alpha$$

To maximize the projected variance $\sigma_{\boldsymbol{u}}^2$, we thus choose the largest eigenvalue λ_1 of Σ, and the dominant eigenvector $\boldsymbol{u}_1$ specifies the direction of most variance, also called the *first principal component*.

Iris Data: First Principal Component



Minimum Squared Error Approach

The direction that maximizes the projected variance is also the one that minimizes the average squared error. The mean squared error (MSE) optimization condition is

$$MSE(\boldsymbol{u}) = \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{\epsilon}_i\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{x}_i'\|^2 = \sum_{i=1}^{n}\frac{\|\boldsymbol{x}_i\|^2}{n} - \boldsymbol{u}^T\Sigma\boldsymbol{u}$$

Exercise!

Since the first term is fixed for a dataset $\boldsymbol{D}$, we see that the direction $\boldsymbol{u}_1$ that maximizes the variance is also the one that minimizes the MSE. Further,

$$\sum_{i=1}^{n}\frac{\|\boldsymbol{x}_i\|^2}{n} - \boldsymbol{u}^T\Sigma\boldsymbol{u} = var(\boldsymbol{D}) = tr(\Sigma) = \sum_{i=1}^{d}\sigma_i^2$$

Thus, for the direction $\boldsymbol{u}_1$ that minimizes MSE, we have

$$MSE(\boldsymbol{u}_1) = var(\boldsymbol{D}) - \boldsymbol{u}_1^T\Sigma\boldsymbol{u}_1 = var(\boldsymbol{D}) - \lambda_1$$

Best 2-dimensional Approximation

The best 2D subspace that captures the most variance in $\boldsymbol{D}$ comprises the eigenvectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ corresponding to the largest and second largest eigenvalues λ_1 and λ_2, respv.

Let $\boldsymbol{U}_2 = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 \end{pmatrix}$ be the matrix whose columns correspond to the two principal components. Given the point $\boldsymbol{x}_i \in \boldsymbol{D}$ its projected coordinates are computed as follows:

$$\boldsymbol{a}_i = \boldsymbol{U}_2^T \boldsymbol{x}_i$$

Let $\boldsymbol{A}$ denote the projected 2D dataset. The total projected variance for $\boldsymbol{A}$ is given as

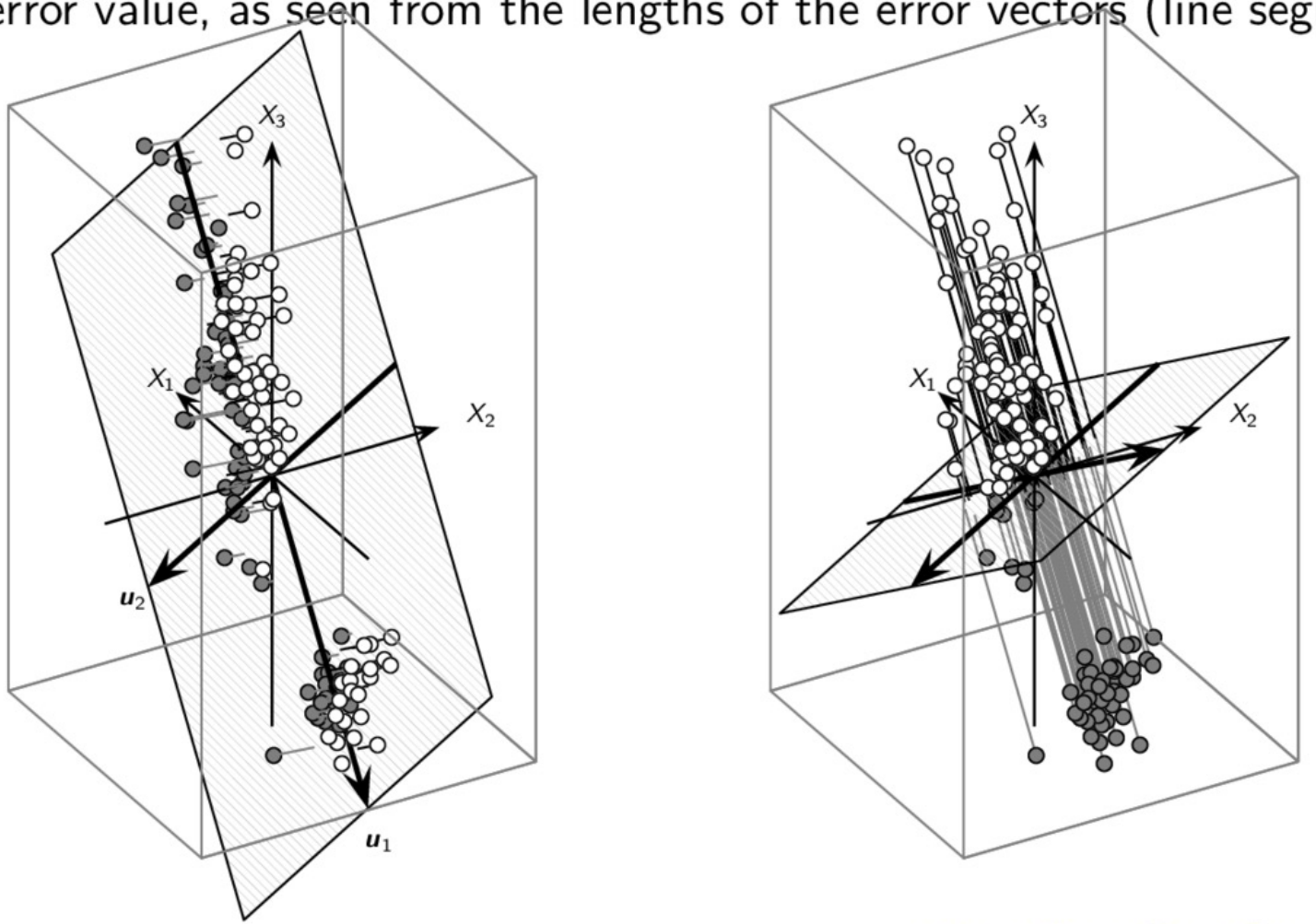
$$var(\boldsymbol{A}) = \boldsymbol{u}_1^T \Sigma \boldsymbol{u}_1 + \boldsymbol{u}_2^T \Sigma \boldsymbol{u}_2 = \boldsymbol{u}_1^T \lambda_1 \boldsymbol{u}_1 + \boldsymbol{u}_2^T \lambda_2 \boldsymbol{u}_2 = \lambda_1 + \lambda_2$$

The first two principal components also minimize the mean square error objective, since

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_i'\|^2 = var(\boldsymbol{D}) - \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{x}_i^T \boldsymbol{P}_2 \boldsymbol{x}_i \right) = var(\boldsymbol{D}) - var(\boldsymbol{A})$$

Optimal and Non-optimal 2D Approximations

The optimal subspace maximizes the variance, and minimizes the squared error, whereas the nonoptimal subspace captures less variance, and has a high mean squared error value, as seen from the lengths of the error vectors (line segments).



Best r-dimensional Approximation

To find the best r-dimensional approximation to $\boldsymbol{D}$, we compute the eigenvalues of Σ. Because Σ is positive semidefinite, its eigenvalues are non-negative

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r \geq \lambda_{r+1} \cdots \geq \lambda_d \geq 0$$

We select the r largest eigenvalues, and their corresponding eigenvectors to form the best r-dimensional approximation.

Total Projected Variance: Let $\boldsymbol{U}_r = \begin{pmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r \end{pmatrix}$ be the r-dimensional basis vector matrix, withe the projection matrix given as $\boldsymbol{P}_r = \boldsymbol{U}_r \boldsymbol{U}_r^T = \sum_{i=1}^{r} \boldsymbol{u}_i \boldsymbol{u}_i^T$.

Let $\boldsymbol{A}$ denote the dataset formed by the coordinates of the projected points in the r-dimensional subspace. The projected variance is given as

$$var(\boldsymbol{A}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i^T \boldsymbol{P}_r \boldsymbol{x}_i = \sum_{i=1}^{r} \boldsymbol{u}_i^T \Sigma \boldsymbol{u}_i = \sum_{i=1}^{r} \lambda_i$$

Mean Squared Error: The mean squared error in r dimensions is

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \boldsymbol{x}_i' \right\|^2 = var(\boldsymbol{D}) - \sum_{i=1}^{r} \lambda_i = \sum_{i=1}^{d} \lambda_i - \sum_{i=1}^{r} \lambda_i$$

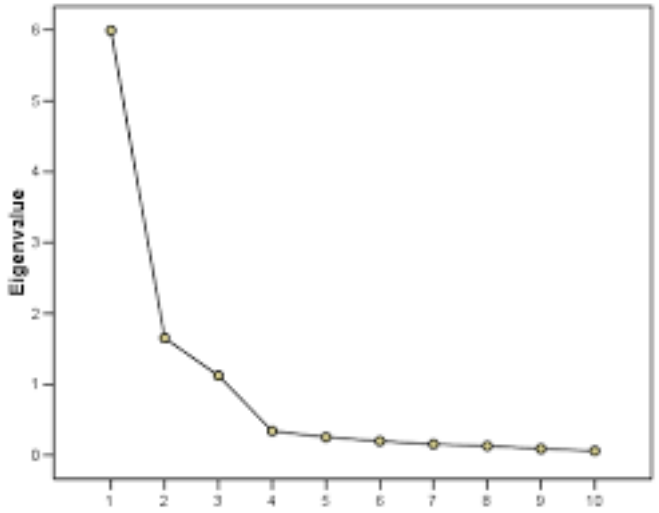
Choosing the Dimensionality

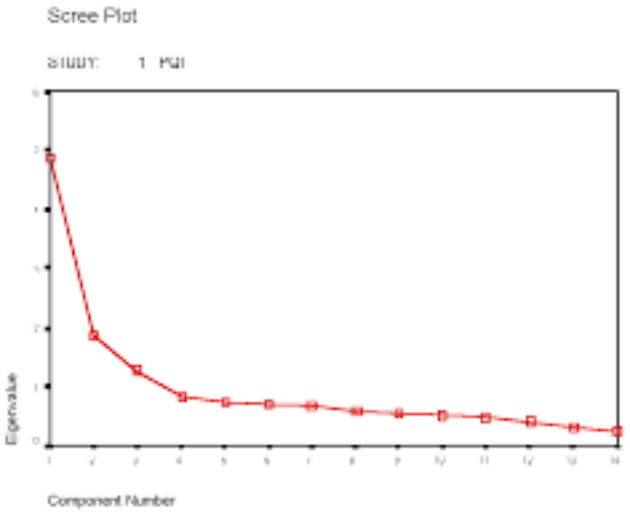
One criteria for choosing r is to compute the fraction of the total variance captured by the first r principal components, computed as

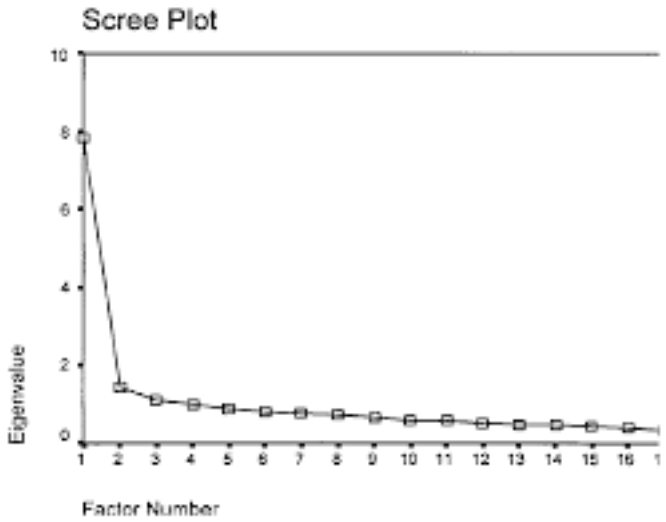
$$f(r) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_r}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{r} \lambda_i}{var(\boldsymbol{D})}$$

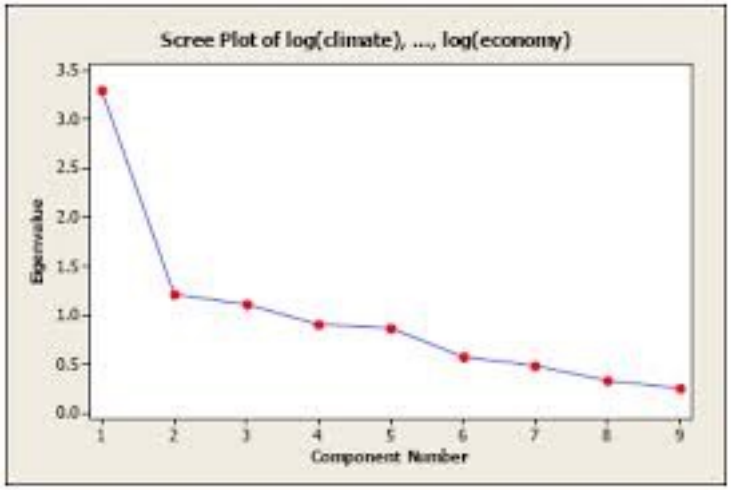
Given a certain desired variance threshold, say α, starting from the first principal component, we keep on adding additional components, and stop at the smallest value r, for which $f(r) \geq \alpha$. In other words, we select the fewest number of dimensions such that the subspace spanned by those r dimensions captures at least α fraction (say 0.9) of the total variance.

Scree Plots:
eigenvalues in decreasing order









Principal Component Analysis: Algorithm

PCA $(\boldsymbol{D}, \alpha)$:

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ // compute mean

2 $\boldsymbol{Z} = \boldsymbol{D} - 1 \cdot \boldsymbol{\mu}^T$ // center the data

3 $\Sigma = \frac{1}{n}\left(\boldsymbol{Z}^T\boldsymbol{Z}\right)$ // compute covariance matrix

4 $(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\Sigma)$ // compute eigenvalues

5 $\boldsymbol{U} = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_d \end{pmatrix} = \text{eigenvectors}(\Sigma)$ // compute eigenvectors

6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance

7 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

8 $\boldsymbol{U}_r = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_r \end{pmatrix}$ // reduced basis

9 $\boldsymbol{A} = \{\boldsymbol{a}_i \mid \boldsymbol{a}_i = \boldsymbol{U}_r^T\boldsymbol{x}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

Iris Principal Components

Covariance matrix:

$$\Sigma = \begin{pmatrix} 0.681 & -0.039 & 1.265 \\ -0.039 & 0.187 & -0.320 \\ 1.265 & -0.32 & 3.092 \end{pmatrix}$$

The eigenvalues and eigenvectors of Σ

$$\lambda_1 = 3.662 \qquad \lambda_2 = 0.239 \qquad \lambda_3 = 0.059$$

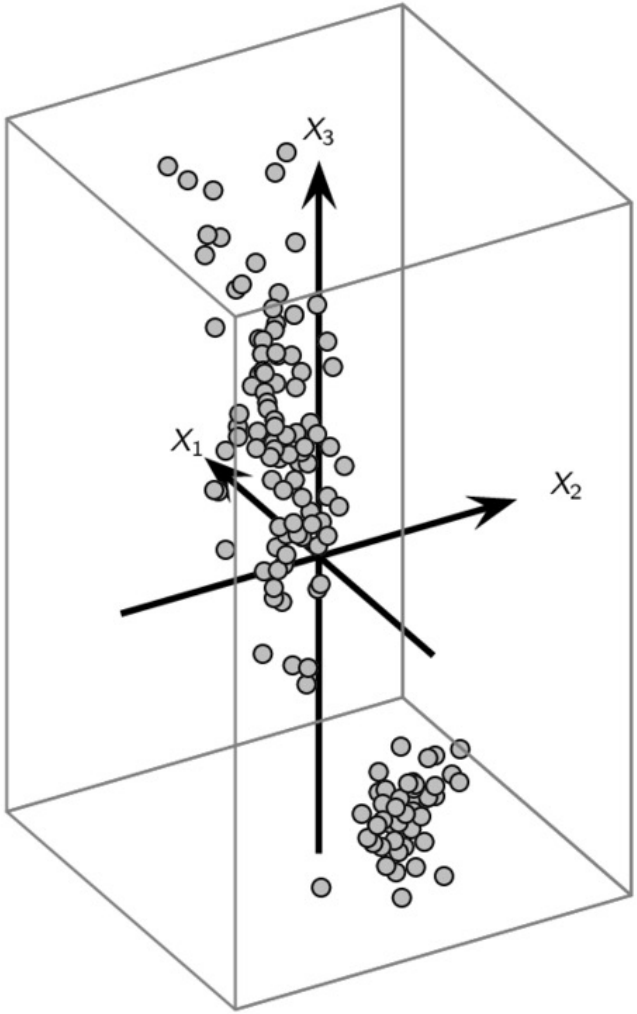
$$\boldsymbol{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix} \qquad \boldsymbol{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix} \qquad \boldsymbol{u}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

The total variance is therefore $\lambda_1 + \lambda_2 + \lambda_3 = 3.662 + 0.239 + 0.059 = 3.96$.
The fraction of total variance for different values of r is given as

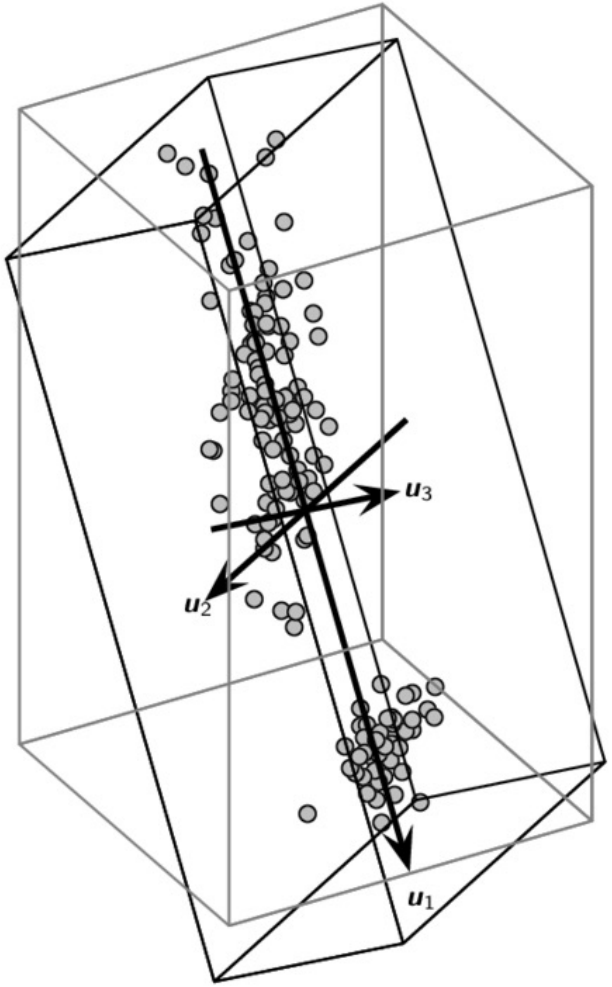
r	1	2	3
$f(r)$	0.925	0.985	1.0

This $r = 2$ PCs are need to capture $\alpha = 0.95$ fraction of variance.

Iris Data: Optimal 3D PC Basis

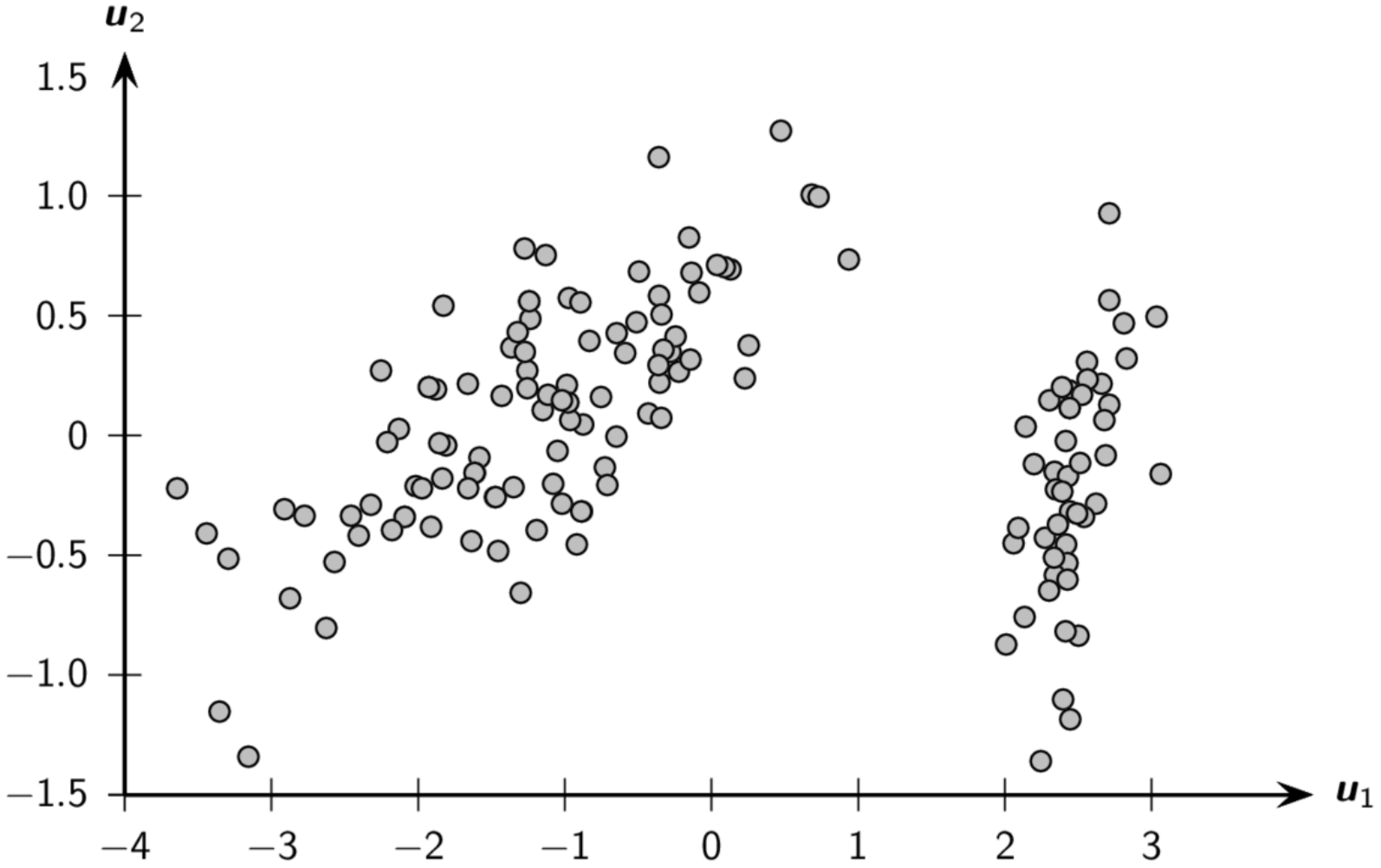


Iris Data (3D)



Optimal 3D Basis

Iris Principal Components: Projected Data (2D)



Geometry of PCA

Geometrically, when $r = d$, PCA corresponds to a orthogonal change of basis, so that the total variance is captured by the sum of the variances along each of the principal directions $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d$, and further, all covariances are zero.

Let $\boldsymbol{U}$ be the $d \times d$ orthogonal matrix $\boldsymbol{U} = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_d \end{pmatrix}$, with $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$. Let $\Lambda = diag(\lambda_1, \cdots, \lambda_d)$ be the diagonal matrix of eigenvalues. Each principal component $\boldsymbol{u}_i$ corresponds to an eigenvector of the covariance matrix Σ

$$\Sigma \boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \text{ for all } 1 \leq i \leq d$$

which can be written compactly in matrix notation:

$$\Sigma \boldsymbol{U} = \boldsymbol{U} \Lambda \text{ which implies } \Sigma = \boldsymbol{U} \Lambda \boldsymbol{U}^T$$

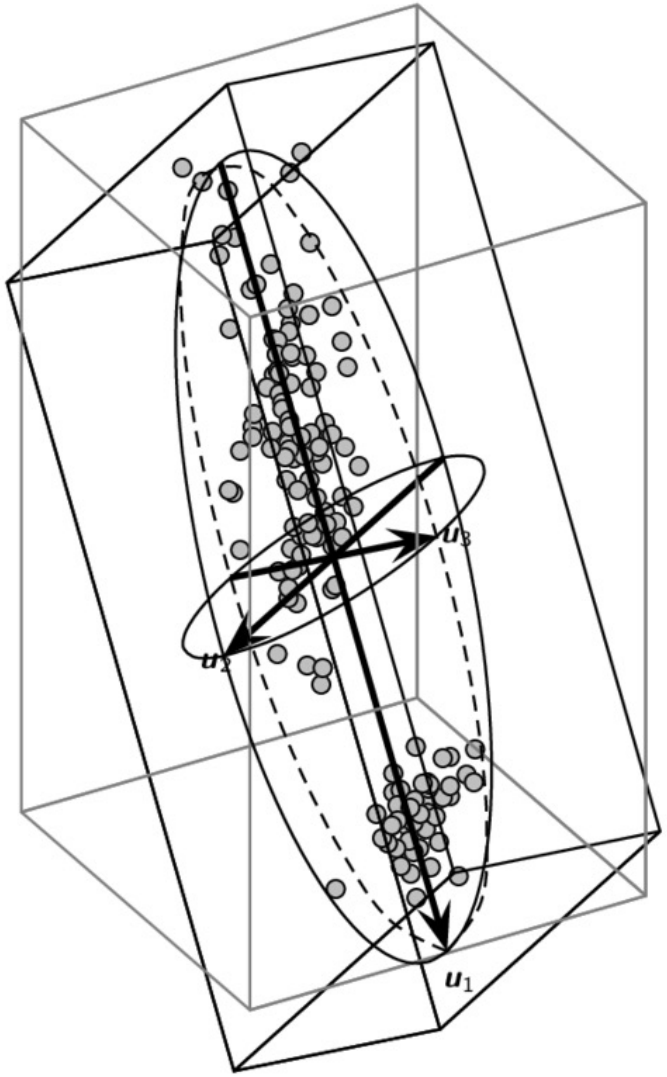
Thus, Λ represents the covariance matrix in the new PC basis.

In the new PC basis, the equation

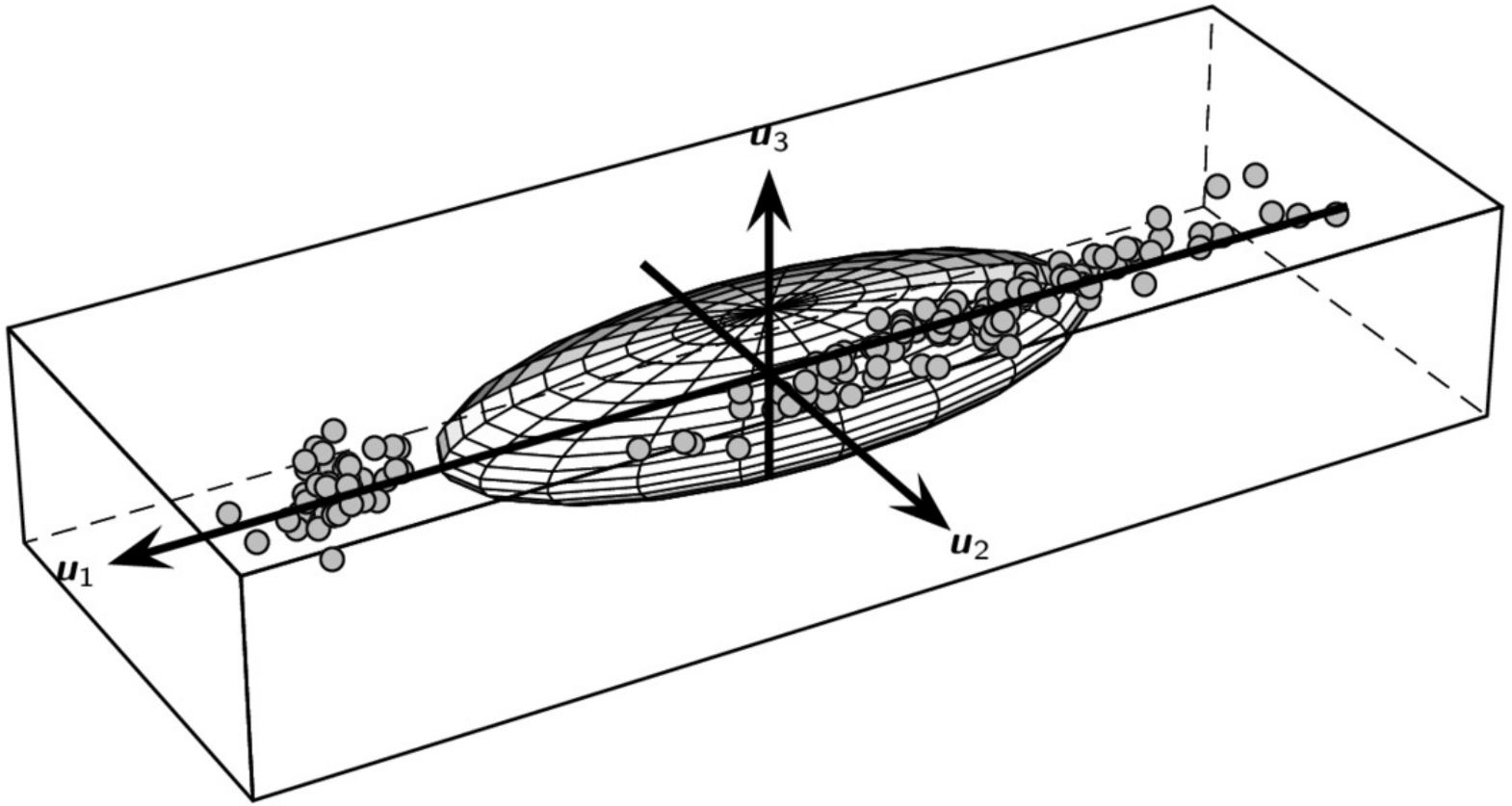
$$\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x} = 1$$

defines a d-dimensional ellipsoid (or hyper-ellipse). The eigenvectors $\boldsymbol{u}_i$ of Σ, that is, the principal components, are the directions for the principal axes of the ellipsoid. The square roots of the eigenvalues, that is, $\sqrt{\lambda_i}$, give the lengths of the semi-axes.

Iris: Elliptic Contours in Standard Basis



Iris: Axis-Parallel Ellipsoid in PC Basis



Stop here!

Kernel Principal Component Analysis

Principal component analysis can be extended to find nonlinear "directions" in the data using kernel methods. Kernel PCA finds the directions of most variance in the feature space instead of the input space. Using the *kernel trick*, all PCA operations can be carried out in terms of the kernel function in input space, without having to transform the data into feature space.

Let ϕ be a function that maps a point $\boldsymbol{x}$ in input space to its image $\phi(\boldsymbol{x}_i)$ in feature space. Let the points in feature space be centered and let Σ_ϕ be the covariance matrix. The first PC in feature space correspond to the dominant eigenvector

$$\Sigma_\phi \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1$$

where

$$\Sigma_\phi = \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T$$

Kernel Principal Component Analysis

It can be shown that $\boldsymbol{u}_1 = \sum_{i=1}^{n} c_i \phi(\boldsymbol{x}_i)$. That is, the PC direction in feature space is a linear combination of the transformed points.

The coefficients are captured in the weight vector

$$\boldsymbol{c} = \left(c_1, c_2, \cdots, c_n\right)^T$$

Substituting into the eigen-decomposition of Σ_ϕ and simplifying, we get:

$$\boldsymbol{K}\boldsymbol{c} = n\lambda_1 \boldsymbol{c} = \eta_1 \boldsymbol{c}$$

Thus, the weight vector $\boldsymbol{c}$ is the eigenvector corresponding to the largest eigenvalue η_1 of the kernel matrix $\boldsymbol{K}$.

Kernel Principal Component Analysis

The weight vector $\boldsymbol{c}$ can be used to then find $\boldsymbol{u}_1$ via $\boldsymbol{u}_1 = \sum_{i=1}^{n} c_i \phi(\boldsymbol{x}_i)$.

The only constraint we impose is that $\boldsymbol{u}_1$ should be normalized to be a unit vector, which implies $\|\boldsymbol{c}\|^2 = \frac{1}{\eta_1}$.

We cannot compute directly the principal direction, but we can project any point $\phi(\boldsymbol{x})$ onto the principal direction $\boldsymbol{u}_1$, as follows:

$$\boldsymbol{u}_1^T \phi(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) = \sum_{i=1}^{n} c_i K(\boldsymbol{x}_i, \boldsymbol{x})$$

which requires only kernel operations.

We can obtain the additional principal components by solving for the other eigenvalues and eigenvectors of

$$\boldsymbol{K}\boldsymbol{c}_j = n\lambda_j \boldsymbol{c}_j = \eta_j \boldsymbol{c}_j$$

If we sort the eigenvalues of $\boldsymbol{K}$ in decreasing order $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_n \geq 0$, we can obtain the jth principal component as the corresponding eigenvector $\boldsymbol{c}_j$. The variance along the jth principal component is given as $\lambda_j = \frac{\eta_j}{n}$.

Kernel PCA Algorithm

KernelPCA $(\boldsymbol{D}, K, \alpha)$**:**

1 $\boldsymbol{K} = \left\{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1,\ldots,n}$ // compute $n \times n$ kernel matrix

2 $\boldsymbol{K} = (\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n\times n})\boldsymbol{K}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}_{n\times n})$ // center the kernel matrix

3 $(\eta_1, \eta_2, \ldots, \eta_d) = \text{eigenvalues}(\boldsymbol{K})$ // compute eigenvalues

4 $\begin{pmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n\end{pmatrix} = \text{eigenvectors}(\boldsymbol{K})$ // compute eigenvectors

5 $\lambda_i = \frac{\eta_i}{n}$ for all $i = 1, \ldots, n$ // compute variance for each component

6 $\boldsymbol{c}_i = \sqrt{\frac{1}{\eta_i}} \cdot \boldsymbol{c}_i$ for all $i = 1, \ldots, n$ // ensure that $\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$

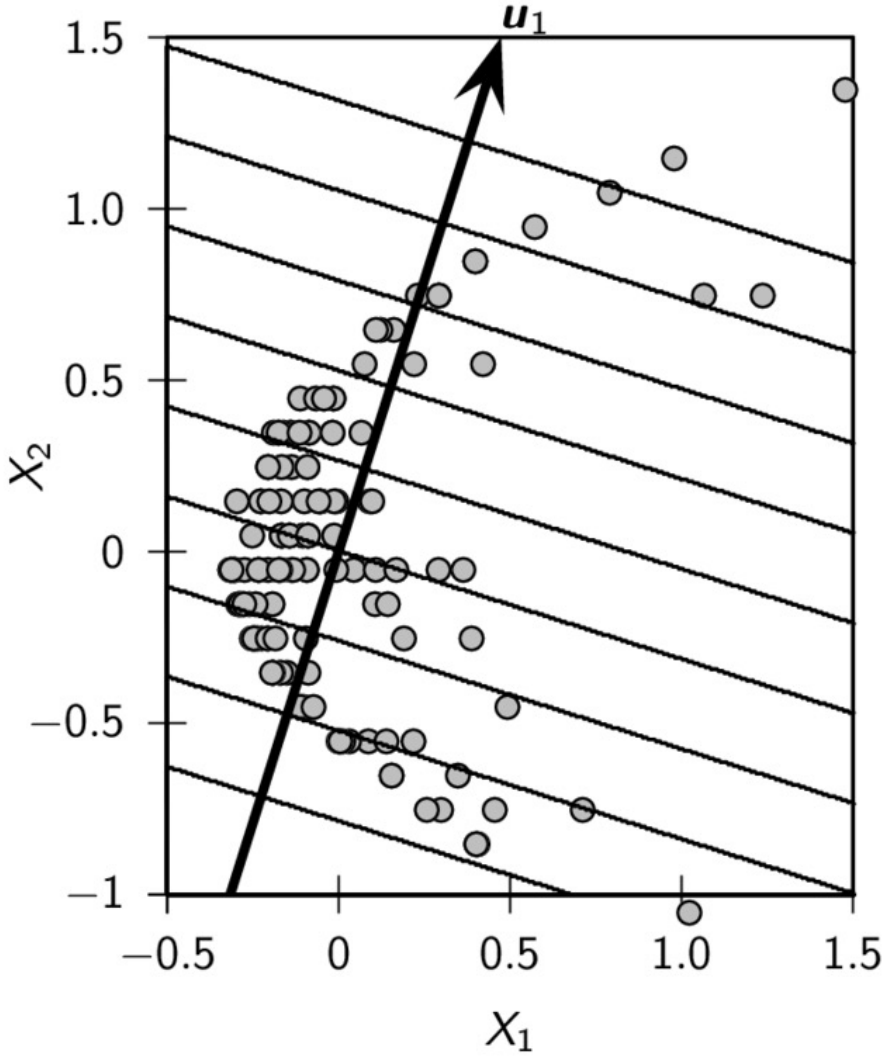
7 $f(r) = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance

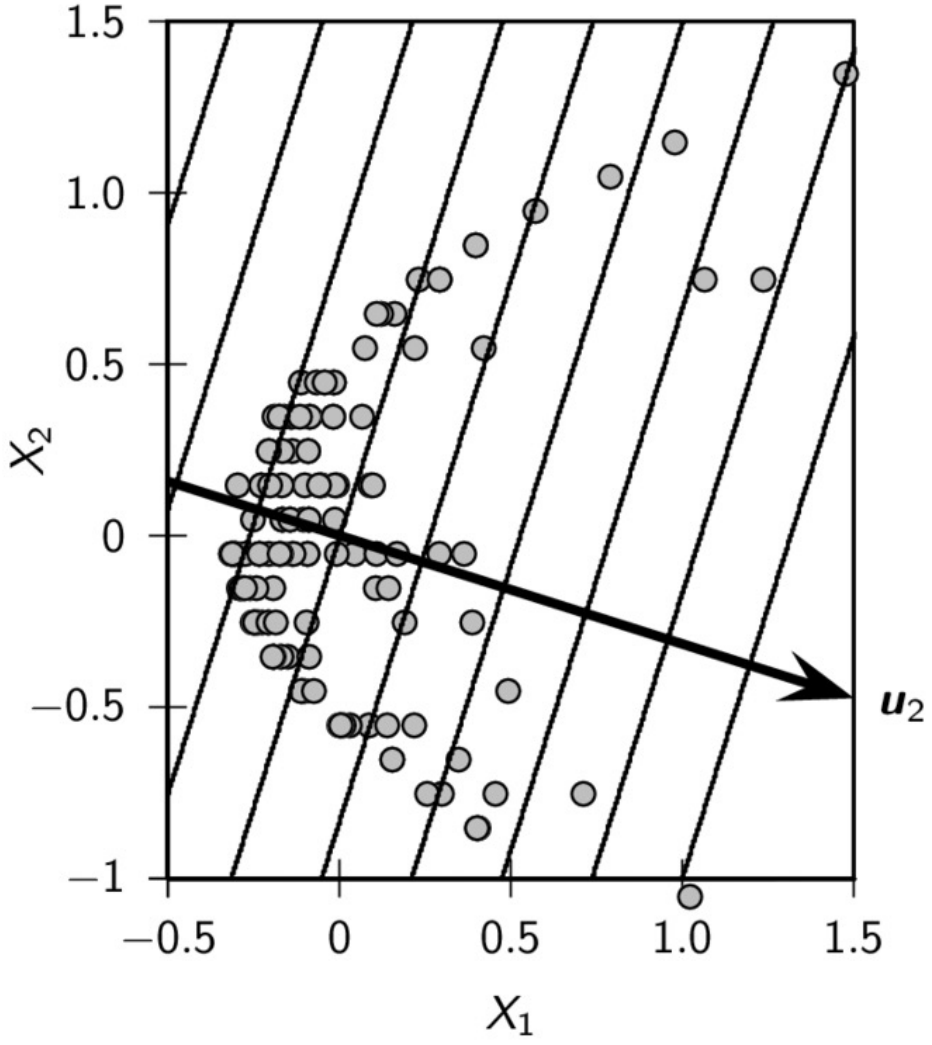
8 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

9 $\boldsymbol{C}_r = \begin{pmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_r\end{pmatrix}$ // reduced basis

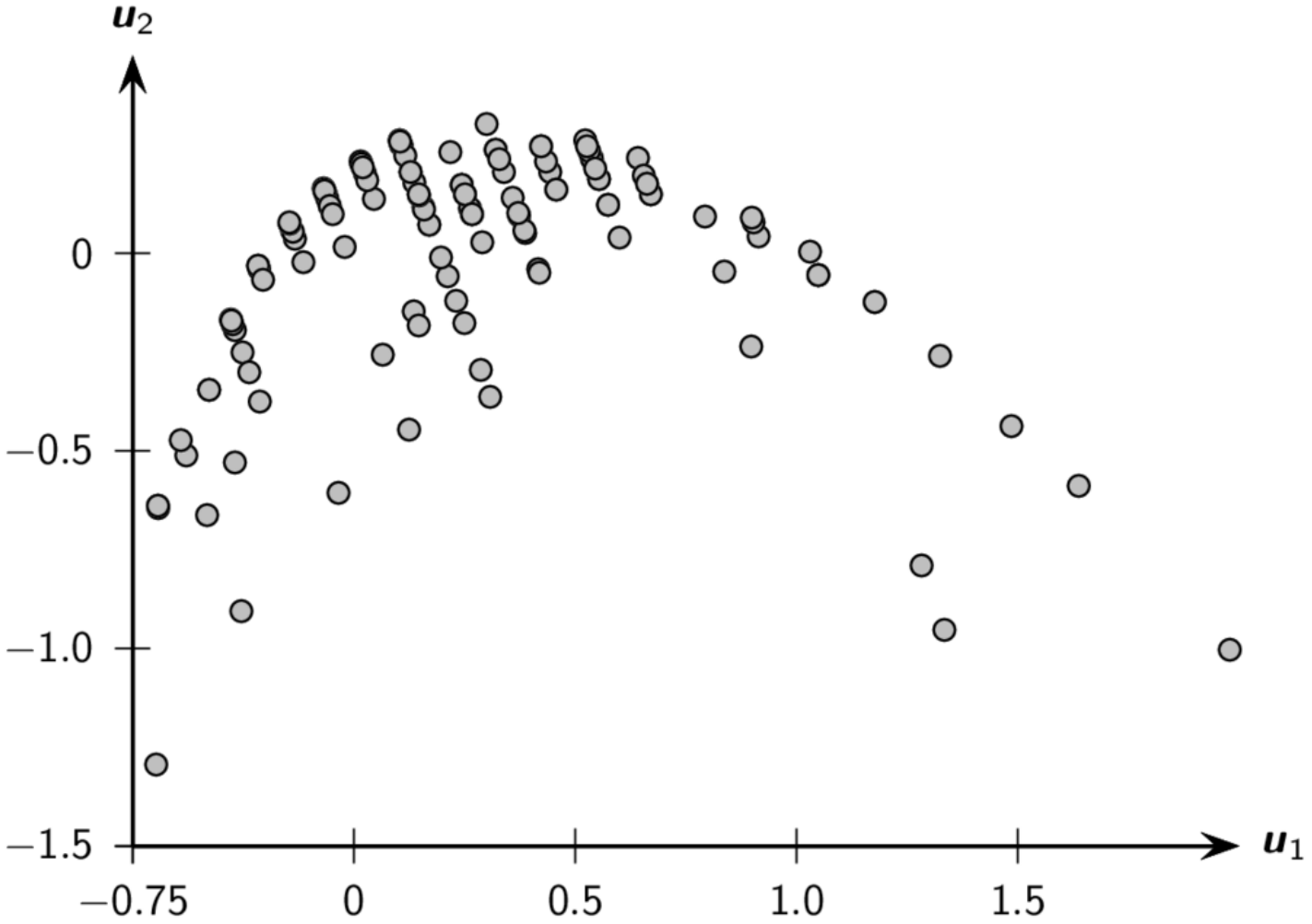
10 $\boldsymbol{A} = \{\boldsymbol{a}_i \mid \boldsymbol{a}_i = \boldsymbol{C}_r^T \boldsymbol{K}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

Nonlinear Iris Data: PCA in Input Space



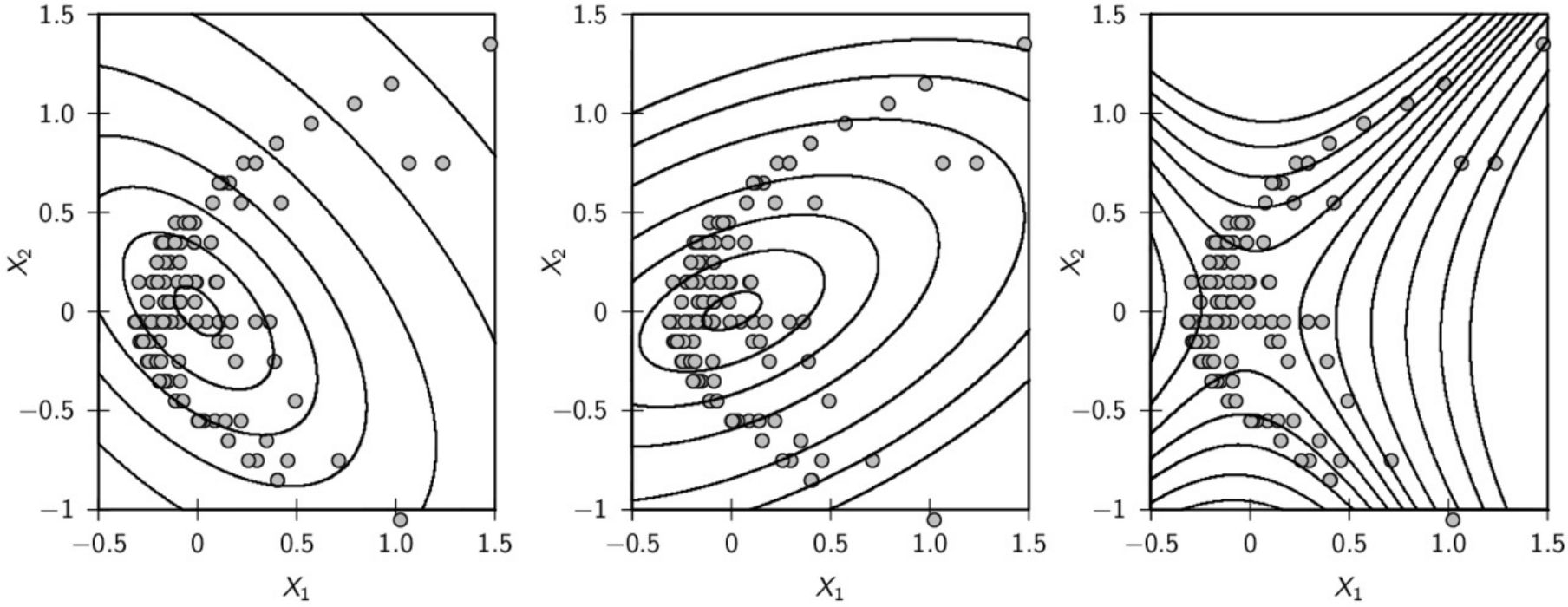


Nonlinear Iris Data: Projection onto PCs



Kernel PCA: 3 PCs (Contours of Constant Projection)

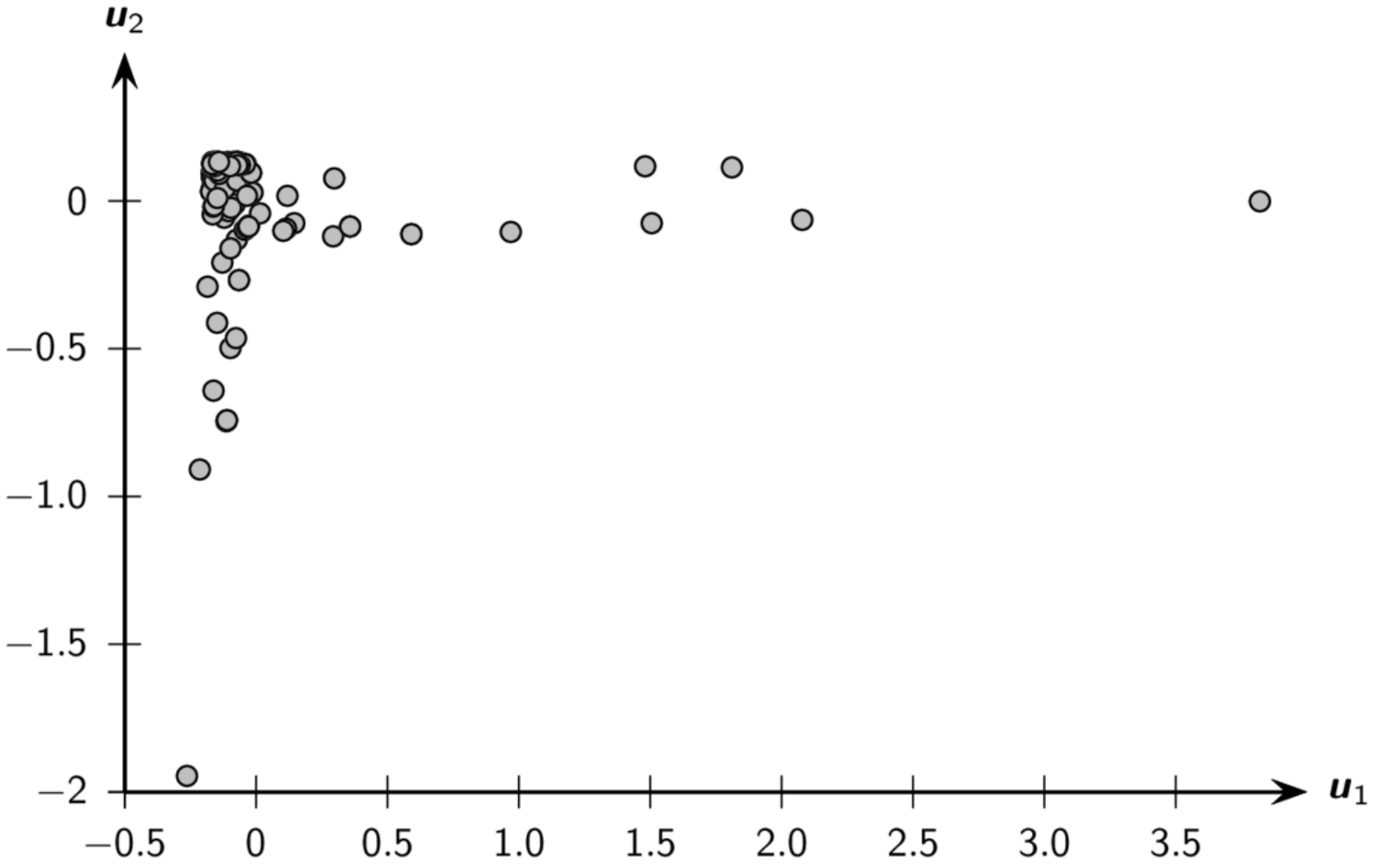
Homogeneous Quadratic Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2$



(a) $\lambda_1 = 0.2067$ (b) $\lambda_2 = 0.0596$ (c) $\lambda_3 = 0.0184$

Kernel PCA: Projected Points onto 2 PCs

Homogeneous Quadratic Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2$



Singular Value Decomposition

Principal components analysis is a special case of a more general matrix decomposition method called *Singular Value Decomposition (SVD)*. PCA yields the following decomposition of the covariance matrix:

$$\Sigma = \boldsymbol{U} \Lambda \boldsymbol{U}^T$$

where the covariance matrix has been factorized into the orthogonal matrix $\boldsymbol{U}$ containing its eigenvectors, and a diagonal matrix Λ containing its eigenvalues (sorted in decreasing order).

SVD generalizes the above factorization for any matrix. In particular for an $n \times d$ data matrix $\boldsymbol{D}$ with n points and d columns, SVD factorizes $\boldsymbol{D}$ as follows:

$$\boldsymbol{D} = \boldsymbol{L} \Delta \boldsymbol{R}^T$$

where $\boldsymbol{L}$ is a orthogonal $n \times n$ matrix, $\boldsymbol{R}$ is an orthogonal $d \times d$ matrix, and Δ is an $n \times d$ "diagonal" matrix, defined as $\Delta(i,i) = \delta_i$, and 0 otherwise. The columns of $\boldsymbol{L}$ are called the *left singular and the columns of* $\boldsymbol{R}$ *(or rows of* $\boldsymbol{R}^T$*) are called the* right singular vectors. The entries δ_i are called the *singular values* of $\boldsymbol{D}$, and they are all non-negative.

Reduced SVD

If the rank of $\boldsymbol{D}$ is $r \leq \min(n, d)$, then there are only r nonzero singular values, ordered as follows: $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_r > 0$.

We discard the left and right singular vectors that correspond to zero singular values, to obtain the *reduced SVD* as

$$\boldsymbol{D} = \boldsymbol{L}_r \Delta_r \boldsymbol{R}_r^T$$

where $\boldsymbol{L}_r$ is the $n \times r$ matrix of the left singular vectors, $\boldsymbol{R}_r$ is the $d \times r$ matrix of the right singular vectors, and Δ_r is the $r \times r$ diagonal matrix containing the positive singular vectors.

The reduced SVD leads directly to the *spectral decomposition* of $\boldsymbol{D}$ given as

$$\boldsymbol{D} = \sum_{i=1}^{r} \delta_i \boldsymbol{l}_i \boldsymbol{r}_i^T$$

The best rank q approximation to the original data $\boldsymbol{D}$ is the matrix $\boldsymbol{D}_q = \sum_{i=1}^{q} \delta_i \boldsymbol{l}_i \boldsymbol{r}_i^T$ that minimizes the expression $\|\boldsymbol{D} - \boldsymbol{D}_q\|_F$, where $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} \boldsymbol{A}(i,j)^2}$ is called the *Frobenius Norm* of $\boldsymbol{A}$.

Connection Between SVD and PCA

Assume $\boldsymbol{D}$ has been centered, and let $\boldsymbol{D} = \boldsymbol{L}\Delta\boldsymbol{R}^T$ via SVD. Consider the *scatter matrix* for $\boldsymbol{D}$, given as $\boldsymbol{D}^T\boldsymbol{D}$. We have

$$\boldsymbol{D}^T\boldsymbol{D} = \left(\boldsymbol{L}\Delta\boldsymbol{R}^T\right)^T\left(\boldsymbol{L}\Delta\boldsymbol{R}^T\right) = \boldsymbol{R}\Delta^T\boldsymbol{L}^T\boldsymbol{L}\Delta\boldsymbol{R}^T = \boldsymbol{R}(\Delta^T\Delta)\boldsymbol{R}^T = \boldsymbol{R}\Delta_d^2\boldsymbol{R}^T$$

where Δ_d^2 is the $d \times d$ diagonal matrix defined as $\Delta_d^2(i,i) = \delta_i^2$, for $i = 1, \ldots, d$.

The covariance matrix of centered $\boldsymbol{D}$ is given as $\Sigma = \frac{1}{n}\boldsymbol{D}^T\boldsymbol{D}$; we get

$$\begin{aligned}\boldsymbol{D}^T\boldsymbol{D} &= n\Sigma \\ &= n\boldsymbol{U}\Lambda\boldsymbol{U}^T \\ &= \boldsymbol{U}(n\Lambda)\boldsymbol{U}^T\end{aligned}$$

The right singular vectors $\boldsymbol{R}$ are the same as the eigenvectors of Σ. The singular values of $\boldsymbol{D}$ are related to the eigenvalues of Σ as

$$n\lambda_i = \delta_i^2, \text{ which implies } \lambda_i = \frac{\delta_i^2}{n}, \text{ for } i = 1, \ldots, d$$

Likewise the left singular vectors in $\boldsymbol{L}$ are the eigenvectors of the matrix $n \times n$ matrix $\boldsymbol{D}\boldsymbol{D}^T$, and the corresponding eigenvalues are given as δ_i^2.

Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 7: Dimensionality Reduction