Final project

PCA-Principal Component Analysis

Rami Amasha

322241373

Department of mathematics

המכללה האקדמית להנדסה בכרמיאל
בראודה

Component

1.Introduction

1.1)What is PCA ?

- Principal component analysis (PCA) is a statistical technique that is useful for compression and extract useful information from multivariate data sets.
- The purpose is to reduce the dimensionality of a data set (sample) by finding a new set of variables (Uncorrelated), smaller than the original set of variables (Correlated), without much loss of information and still retains most of the sample's information.
- Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

1.2)PCA applications.

The algorithm can be used on its own, or it can serve as a data cleaning or data preprocessing technique used before another machine learning algorithm.

On its own, PCA is used across a variety of use cases:

- Visualize multidimensional data- data visualizations are a great tool for communicating multidimensional data as 2-3 dimensional plots.
- Compress information- PCA is used to compress information to store and transmit data more efficiently. For example, it can be used to compress images or videos without losing to much quality, or signal processing. The technique has successfully been applied across a wide range of compression problems in pattern recognition (specifically face recognition), image recognition, and more.
- Simplify complex business decisions-PCA has been employed to simplify traditionally complex business decisions. For example, traders use over 300 financial instruments to manage portfolios. The algorithm has proven successful in the risk management of interest rate derivative portfolios, lowering the number of financial instruments from more than 300 to just 3-4 principal components.

1.3)What are the advantages and disadvantages of PCA ?

- Advantages:
  1) Easy to compute – PCA is based on linear algebra, which is computationally easy to solve by computers.
  2) Speeds up other machine learning algorithms – machine learning algorithms converge faster when trained on principal components instead of the original dataset.
  3) Counteracts the issues of high-dimensional data – high dimensional data causes regression-based algorithms to overfit easily. By using PCA to lower the dimensions of the training dataset, we prevent the algorithm from overfitting.

- Disadvantages:
  1) Low interpretability of principal components – principal components are linear combinations of features form the original data, but they are not as easy to interpret. For example, its difficult to specify which feature is the most important after computing principal components.
  2) The trade-off between information loss and dimensionality reduction – although dimensionality reduction is very useful, it comes at a cost. Information loss is a necessary part of PCA. Balancing the trade-off between dimensionality reduction and information loss is unfortunately a necessary compromise that we have to make when using PCA.

1.4)Assumptions and limitations of PCA .

- PCA assumes correlation between the features- if the features (or dimensions or columns...) are not correlated, the PCA will not be able to determine principal components.
- PCA is sensitive to the scale of the features- For example, if we have to features- one takes values between0 and 1000, while the other takes values between 0 and 1. PCA will be extremely biased towards the first feature being the first principal component, regardless of the actual maximum variance within the data. This is why it's so important to standardize the values first.
- PCA assumes a linear relationship between features- The algorithm is not well suited to capturing non-linear relationships. That's why its advised to turn non-linear features or relationships between features into linear, using the standard methods such as log transforms.
- Technical implementations often assume no missing values- When computing PCA using statistical software tools, they often assume that the feature set has no missing values (no empty rows).

2.Mathematics behind PCA.

Principal component analysis (PCA) is the problem of fitting a low-dimensional affine subspace to a set of data points in a high-dimensional space. PCA is, by now, well established in the literature, and has become one of the most useful tools for data modeling, compression, and visualization.

2.1. Modeling Data (Statistical Models versus Geometric Models).

There are essentially two main categories of models and approaches for modeling a data set. Methods of the first category model the data as random samples from a probability distribution and try to learn this distribution from the data. We call such models statistical models. Models of the second category model the overall geometric shape of the data set with deterministic models such as subspaces, smooth manifolds, or topological spaces. We call such models geometric models.

2.2. Statistical view of PCA.

Historically, PCA was first formulated in a statistical setting to estimate the principal components of a multivariate random variable. Specifically, given a zero-mean multivariate random variable $x \in R^D$ and an integer d<D, the d " principal components" of x, y∈ D are defined as the d uncorrelated linear components of x.

$$y_i = u_i{}^T x, \qquad u_i \in R^D, \qquad i = 1,2, \dots \dots \dots d, \qquad (2.1)$$

Such that the variance of y_i is maximized subject to

$$u_i{}^T u_i = 1 \text{ and } \quad Var(y_1) \geq Var(y_2) \dots \dots \dots \dots \geq Var(y_d) > 0 \qquad (2.2)$$

For example, to find the first principal component y_1, we seek a vector $u^*{}_1 \epsilon R^D$ such that

$$u^*{}_1 = \max \ Var(u_1{}^T x) \quad s.t. \ u_1{}^T u_1 = 1 \qquad (2.3)$$

The following theorem shows that the principal components of x can be computed from the eigenvectors of its covariance matrix $\Sigma_x = E[xx^T]$.

Theorem 2.1(principal components of a random variable).

Assume that rank $(\Sigma_x) \geq d$. Then the first d principal components of a zero-mean multivariate random variable x, denoted by y_i for $i = 1,2, \dots . d$ are given by

$$y_i = u_i{}^T x \qquad (2.4)$$

Where $\{u_i\}^d{}_{i=1}$ are d orthonormal eigenvectors of $\Sigma_x = E[xx^T]$ associated with its largest eigenvalues $\{\lambda_i\}^d{}_{i=1}$. Moreover, $\lambda_i = Var(\lambda_i)$ for $i = 1,2, \dots . d$.

Proof . let us first assume that Σ_x does not have repeated eigenvalues. In this case, since the matrix Σ_x is real and symmetric, its eigenvalues are real and its eigenvectors form

a basis of R^D. Moreover the eigenvectors are unique(up to sign), and the eigenvectors corresponding to different eigenvalues are orthogonal to each other.

Now notice that for every $u \in R^D$, we have that

$$\mathrm{Var}(u^T x) = E[(u^T x)^2] = E[u^T x x^T u] = u^T \Sigma_x u \qquad (2.5)$$

Therefore, the optimization problem in (2.3) is equivalent to

$$\max_{u_1 \in R^D} u_1^T \Sigma_x u_1 \quad s.t.\ u_1^T u_1 = 1 \qquad (2.6)$$

to solve the above constrained optimization problem, we use the method of Lagrange multipliers. The Lagrangian function is given by

$$L = u_1^T \Sigma_x u_1 + \lambda_1 (1 - u_1^T u_1) \qquad (2.7)$$

Where $\lambda_1 \epsilon R$ is the Lagrange multiplier, from computing the derivatives of L with respect to (u_1, λ_1) and setting them to zero, we obtain the following necessary conditions for (u_1, λ_1) to be an extremum of L:

$$\Sigma_x u_1 = \lambda_1 u_1 \quad \text{and} \quad u_1^T u_1 = 1 \qquad (2.8)$$

This means that u_1 is an eigenvector of Σ_x with associated eigenvalue λ_1. Since the extremum value is $u_1^T \Sigma_x u_1 = \lambda_1 u_1^T u_1 = \lambda_1$, the optimal solution for u_1 is given by the eigenvector of Σ_x associated with its largest eigenvalue $\lambda_1 = \mathrm{Var}(y_1) > 0$. To find the second principal component, u_2, we use the fact that $u_1^T x$ and $u_2^T x$ need to be uncorrelated. This implies that u_2 is orthogonal to u_1. Indeed, from

$$E[(u_1^T x)(u_2^T x)] = E[u_1^T x x u_2^T] = u_1^T \Sigma_x u_2 = \lambda_1 u_1^T u_2 = 0 \qquad (2.9)$$

And $\lambda_1 \neq 0$, we have $u_1^T u_2 = 0$. Thus, to find u_2 we need to solve the following optimization problem:

$$\max_{u_2 \in R^D} u_2^T \Sigma_x u_2 \quad s.t.\ u_1^T u_1 = 1 \quad \text{and} \quad u_1^T u_2 = 0 \qquad (2.10)$$

As before, we define the Lagrangian:

$$L = u_2^T \Sigma_x u_2 + \lambda_2 (1 - u_2^T u_2) + \gamma u_1^T u_2 \qquad (2.11)$$

The necessary conditions for (u_2, λ_2, γ) to be an extremum are:

$$\Sigma_x u_2 + \frac{\gamma}{2} u_1 = \lambda_2 u_2 \,, u_2^T u_2 = 1 \text{ and } u_1^T u_2 = 0 \qquad (2.12)$$

From which it follows that $u_1^T \Sigma_x u_2 + \frac{\gamma}{2} u_1^T u_1 = \lambda_1 u_1^T u_2 + \frac{\gamma}{2} = \lambda_2 u_1^T u_2$, and so $\gamma = 2(\lambda_2 - \lambda_1) u_1^T u_2 = 0$. This implies that $\Sigma_x u_2 = \lambda_2 u_2$ and that the extremum value is $u_2^T \Sigma_x u_2 = \lambda_2 = \mathrm{Var}(y_2)$. Therefore, u_2 is the leading eigenvector of Σ_x restricted

to the orthogonal complement of u_1. Since the eigenvalues of Σ_x are distinct, u_2 is the eigenvector of Σ_x associated with its second-largest eigenvalue.

To find the remaining principal components, we use the fact that for all $i \neq j$, $y_i = u_i^T x$ and $y_j = u_j^T x$ need to be uncorrelated, whence

$$\mathrm{Var}(y_i y_j) = \mathbb{E}[\boldsymbol{u}_i^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{u}_j] = \boldsymbol{u}_i^\top \Sigma_{\boldsymbol{x}} \boldsymbol{u}_j = 0.$$

Using induction, assume that $u_1 \ldots\ldots\ldots\ldots\ldots u_{i-1}$ are the unit-length eigenvectors of Σ_x associated with its top i-1 eigenvalues, and let u_i be the vector defining the i-th principal component, y_i . Then, $\Sigma_x u_j = \lambda_j u_j$ for j=1…….i-1 and $u_i^T \Sigma_x u_j = \lambda_j u_i^T u_j = 0$ for all j=1…….i-1. Since $\lambda_j > 0$, we have that $u_i^T u_j = 0$ for j=1…….i-1. To compute u_i, we build the Lagrangian:

$$\mathscr{L} = \boldsymbol{u}_i^\top \Sigma_{\boldsymbol{x}} \boldsymbol{u}_i + \lambda_i (1 - \boldsymbol{u}_i^\top \boldsymbol{u}_i) + \sum_{j=1}^{i-1} \gamma_j \boldsymbol{u}_i^\top \boldsymbol{u}_j. \qquad (2.13)$$

The necessary conditions for $(u_i, \lambda_i, \gamma_1, \ldots\ldots \gamma_{j-1})$ to be an extremum are

$$\Sigma_{\boldsymbol{x}} \boldsymbol{u}_i + \sum_{j=1}^{i-1} \frac{\gamma_j}{2} \boldsymbol{u}_j = \lambda_i \boldsymbol{u}_i,\ \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 \text{ and } \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0, j = 1, \ldots, i-1, \qquad (2.14)$$

From which it follows that for all j=1…..i-1, we have $u_j^T \Sigma_x u_i + \frac{\gamma_j}{2} = \lambda_j u_j^T u_i + \frac{\gamma_j}{2} = \lambda_i u_j^T u_i$, and $\gamma_j = 2(\lambda_j - \lambda_i) u_j^T u_i = 0$. Since the associated extremum value is $u_i^T u_j = \lambda_i = \mathrm{Var}(y_i)$, u_i is the leading eigenvector of Σ_x restricted to the orthogonal complement of the span of $u_1, \ldots . u_{i-1}$. Since the eigenvalues of Σ_x are distinct, u_i is the eigenvector of Σ_x associated with the i-th largest eigenvalue.

Therefore, when the eigenvalues of Σ_x are distinct, each eigenvector u_i is unique (up to sign), and hence so are the principal components of x.

Now let us consider the case in which Σ_x has repeated eigenvalues. In this case, Σ_x still admits a basis of orthonormal eigenvectors. Specifically, the eigenvectors of Σ_x associated with different eigenvalues are still orthogonal, while the eigenvectors associated with a repeated eigenvalue form an eigen subspace, and every orthonormal basis for this eigen subspace gives a valid set of eigenvectors.

As a consequence, the principal directions $\{u_i\}_{i=1}^d$ are not uniquely defined. For example, if λ_1 is repeated, every eigenvector associated with λ_1 can be chosen as u_1 and any other eigenvector associated with λ_1 and orthogonal to u_1 can be chosen as u_2. Nonetheless, the principal components can still be obtained from any set of the top d eigenvectors of Σ_x , as claimed. ■

The solution to PCA provided by Theorem(2.1)suggests that we may find the d principal components of x simultaneously, rather than one by one. Specifically, if we define a random vector $y = [y_1, y_2 \ldots . y_d]^T \epsilon \mathbb{R}^d$ and a matrix $U=[u_1, u_2 \ldots . u_d] \epsilon \mathbb{R}^{D \times d}$, then, since $y = U^T x$, we have that:

$$\Sigma_y \doteq \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\top] = U^\top \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] U = U^\top \Sigma_x U. \qquad (2.15)$$

From the definition of principal components, the entries of y are uncorrelated. As a result, the matrix Σ_x must be diagonal, and from the proof of theorem (2.1), we showed that the matrix U must be orthonormal, i.e. $U^T U = I_d$.

Recall that every diagonalizable matrix A can be transformed into a diagonal matrix $\Lambda = V^{-1} A V$, where the columns of V are the eigenvectors of A and the diagonal entries of Λ are the corresponding eigenvalues. Recall also that if A is real, symmetric and positive semi-definite, its eigenvalues are real and nonnegative, i.e., $\lambda_i > 0$, and its eigenvectors can be chosen to be orthonormal, so that $V^{-1} = V^T$. Since the matrix Σ_x is real, symmetric, and positive semi-definite, one solution to the equation $\Sigma_y = U^T \Sigma_x U$ is obtained by choosing the columns of U as d eigenvectors of Σ_x and the diagonal entries of Σ_y as the corresponding d eigenvalues. Moreover, since our goal is to maximize the variance of each y_i and $\lambda_i = \mathrm{var}(y_i)$, we conclude that the columns of U are the top d eigenvectors of Σ_x and the entries of Σ_y are the corresponding top d eigenvalues.

Principal components of a Nonzero-Mean random variable

When x does not have zero mean, then d principal components of x are defined as the d uncorrelated affine components

$$y_i = \boldsymbol{u}_i^\top \boldsymbol{x} + a_i \in \mathbb{R}, \quad \boldsymbol{u}_i \in \mathbb{R}^D, \quad i = 1, 2, \ldots, d, \tag{2.16}$$

of x such that the variance of y_i is maximized subject to

$$\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 \quad \text{and} \quad \text{Var}(y_1) \geq \text{Var}(y_2) \geq \cdots \geq \text{Var}(y_d) > 0. \tag{2.17}$$

The principal directions $\{u_i\}_{i=1}^d$ are the d eigenvectors of $\Sigma_x \doteq \mathbb{E}[(x-\mu)(x-\mu)^T]$, where $\mu = \mathbb{E}(x)$, associated with its d largest eigenvalues $\{\lambda_i\}_{i=1}^d$. Moreover, $\lambda_i = \text{Var}(y_i)$ and $a_i = -u_i^T \mu$ for i=1, 2,……d.

Sample Principal Components of a Zero-Mean Random Variable

In practice, we may not know the population covariance matrix Σ_x . Instead, we may be given N i.i.d. samples of the zero-mean random variable x, $\{u_j\}_{j=1}^N$, which we collect into a data matrix $X = [x_1, x_2, \ldots . x_N]$. It is well known from statistics that the minimum likelihood estimate of Σ_x is given by

$$\hat{\Sigma}_N \doteq \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{x}_j \boldsymbol{x}_j^\top = \frac{1}{N} X X^\top. \tag{2.18}$$

We define the d "sample principal components" of x as

$$\hat{y}_i = \hat{\boldsymbol{u}}_i^\top \boldsymbol{x}, \quad i = 1, 2, \ldots, d, \tag{2.19}$$

Where $\{\hat{u}_i\}_{i=1}^d$ are the top eigenvectors of $\hat{\Sigma}_N$, or equivalently those of XX^T . Notice that when the dimension of D of the data is very high, we can avoid computing the eigenvectors of a large matrix XX^T by exploiting the fact that the top eigenvectors of XX^T are the same as the top singular vectors of X. Therefore, the sample principal components of x may be computed from the singular value decomposition (SVD) of $X = U_X \Sigma_X V_X^T$ as $y = U^T x$, where the columns of U are the first d columns of U_X.

Remark 2.2 (Relationship between principal components and sample principal components).

Even though the principal components of x and the sample principal components of x are different notions, under certain assumptions on the distribution of x, they can be related to each other. Specifically, one can show that if x is Gaussian, then every eigenvector $\hat{u}\ of\ \hat{\Sigma}_N$is an asymptotically consistent unbiased estimate for the corresponding eigenvector u of Σ_x.

2.2. A geometric view of PCA.

An alternative geometric view of PCA, which is very related to the SVD, assumes that we are given a set of points $\{x_j\}_{j=1}^{N}$ in $\mathbb{R}^D$ and seeks to find an affine subspace in $S \subset \mathbb{R}^D$ of dimension d that best first these points. Each point $x_j \in S$ can be represented as :

$$x_j = \mu + Uy_j, \quad j = 1 \dots N \qquad (2.20)$$

Where $\mu \in S$ is a point in the subspace, U is a D X d matrix whose columns form a basis for the subspace, and $y_j \in \mathbb{R}^d$ is simply the vector of new coordinates of x_j in the subspace.

Notice that there are some redundancy in the above representation due to the arbitrariness in the choice of μ and U. More precisely, for every $y_0 \in \mathbb{R}^d$, we cab re-present x_j as $x_j = (\mu + Uy_0) + U(y_j - y_0)$. We call this ambiguity the translation ambiguity. Also, for every invertible $A \in \mathbb{R}^{d\,x\,d}$, we can re-present x_j as $x_j = \mu + (UA)(A^{-1}y_j)$. We call this ambiguity the change of basis ambiguity. Therefore, we need some additional constraints in order to end up with a unique solution to the problem of finding an affine subspace for the data.

A common constraint used to resolve the translational ambiguity is to require that the average of the y_j be zero, i.e.

$$\frac{1}{N}\sum_{j=1}^{N} y_j = 0 \qquad (2.21)$$

Where $0 \in \mathbb{R}^d$ is the vector of all zeros, while a common constraint used to resolve the change of basis ambiguity is to require that the columns of U be orthonormal, i.e., $U^TU = I$. This last constraint eliminates the change of the basis ambiguity only up to rotation, because we can still re-present x_j as $x_j = \mu + (UR)(R^Ty_j)$ for some rotation R in $\mathbb{R}^d$. However, this rotational ambiguity can easily be dealt with during optimization, as we shall soon see. The model in (2.20) now assumes that each point x_j lies perfectly in an affine subspace S. In practice, the given points are imperfect and have noise. For example, if point x_j is contaminated by additive noise as ε_j, we have

$$x_j = \mu + Uy_j + \varepsilon_j, \quad j = 1 \dots N \qquad (2.22)$$

In this case we define the "optimal" affine subspace to be the one that minimizes the of squared errors, i.e.,

$$\min_{\mu,U,\{y_j\}} \sum_{j=1}^{N} \|x_j - \mu - Uy_j\|^2, \text{ s.t. } U^TU = I_d \text{ and } \sum_{j=1}^{N} y_j = 0. \qquad (2.23)$$

In order to solve this optimization problem, we define the Lagrangian function

$$\mathscr{L} = \sum_{j=1}^{N} \|\boldsymbol{x}_j - \boldsymbol{\mu} - U\boldsymbol{y}_j\|^2 + \gamma^\top \sum_{j=1}^{N} \boldsymbol{y}_j + \operatorname{trace}\left((I_d - U^\top U)\Lambda\right), \qquad (2.24)$$

Where $\gamma \in \mathbb{R}^d$ and $\Lambda = \Lambda^T \in \mathbb{R}^{d\,x\,d}$ are, respectively, a vector and a matrix of Lagrange multipliers. A necessary condition for μ to be an extremum is

$$-2\sum_{j=1}^{N}(\boldsymbol{x}_j - \boldsymbol{\mu} - U\boldsymbol{y}_j) = \boldsymbol{0} \implies \hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}_N \doteq \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}_j. \qquad (2.25)$$

A necessary condition for y_j to be an extremum is: $-2U^T(x_j - \mu + Uy_j) + \gamma = 0$ (2.26)

Summing over j yields $\gamma = 0$, from which we obtain

$$\hat{\boldsymbol{y}}_j = U^\top(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N). \qquad (2.27)$$

The vector $\hat{y}_j \in \mathbb{R}^d$ is simply the coordinates of the projection of $x_j \in \mathbb{R}^D$ onto the subspace S. we may call such a $\hat{y}$ the "geometric principal components" of x. before optimizing over U, we can replace the optimal values for μ and y_j int the objective function. This leads to the following optimization problem:

$$\min_U \sum_{j=1}^{N} \|(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N) - UU^\top(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_N)\|^2 \quad \text{s.t.} \quad U^\top U = I_d. \qquad (2.28)$$

Note that this is a restatement of the original problem with the mean $\hat{\mu}_N$ subtracted from each of the sample points. Therefore, from now on, we will consider only the case in which data points have zero mean. If such is not the case, simply subtract the mean from each point before computing U.

The following theorem gives a constructive solution for finding an optimal U .

Theorem 2.3 (PCA via SVD).

Let $X = [x_1, x_2, \ldots . x_N] \in \mathbb{R}^{D\,x\,N}$ be the matrix formed by stacking the (zero-mean) data points as its column vectors. Lex $X = U_X\Sigma_X{V_X}^T$ be the SVD of the matrix X. Then for a given d<D, an optimal solution for U is given by the d columns of U_X , an optimal solution for y_j is given by the j-th column of the top d x N submatrix of $\Sigma_X{V_X}^T$, and the optimal objective value is given by $\Sigma^D{}_{i=d+1}\,{\sigma_i}^2$, where σ_i is the i-th singular value of X.

Proof. Since $U^TU = I$, we have $(I - U^TU)(I - U^TU) = I - U^TU$. Then, recalling that $x^TAx = \operatorname{trace}(Axx^T)$, we can rewrite the least-squares error:

$$\sum_{j=1}^{N} \|x_j - UU^\top x_j\|^2 = \sum_{j=1}^{N} x_j^\top (I_D - UU^\top) x_j \qquad (2.29)$$

as trace $((\mathbf{I_D} - \mathbf{UU^T})(\mathbf{XX^T}))$. The first term trace$(\mathbf{XX^T})$ does not depend on U. Therefore, we can transform the minimization of (2.29) to:

$$\max_U \quad \text{trace}(UU^\top XX^\top) \quad \text{s.t.} \quad U^\top U = I_d. \qquad (2.30)$$

Since trace(AB)=trace(BA), the Lagrangian for this problem can be written as:

$$\mathscr{L} = \text{trace}(U^\top XX^\top U) + \text{trace}((I_d - U^\top U)\Lambda), \qquad (2.31)$$

Where $\boldsymbol{\Lambda} = \boldsymbol{\Lambda^T} \in \mathbb{R}^{\boldsymbol{d\,x\,d}}$. The conditions for an extremum are given by

$$\boldsymbol{XX^TU = U\Lambda} \qquad (\mathbf{2.32})$$

Therefore, Where $\boldsymbol{\Lambda} = \boldsymbol{U^TXX^TU}$ and the objective function reduces to trace$(\boldsymbol{\Lambda})$. Recall not that U is defined only up to a rotation, i.e. U'=UR is also a valid solution, hence so is $\boldsymbol{\Lambda'} = \boldsymbol{R\Lambda R^T}$. Since $\boldsymbol{\Lambda}$ is symmetric, it has an orthogonal matric of eigenvectors. Thus, if we choose R to be the matrix of eigenvectors of $\boldsymbol{\Lambda}$, then $\boldsymbol{\Lambda'}$ is diagonal matrix. As a consequence, we can choose $\boldsymbol{\Lambda}$ to be diagonal without loss of generality. It follows from (2.32) that the columns of U must be d eigenvectors of $\boldsymbol{XX^T}$ with the corresponding eigenvalues in the diagonal entries of $\boldsymbol{\Lambda}$.since the goal is to maximize trace$(\boldsymbol{\Lambda})$, an optimal solution is given by the top d eigenvectors of $\boldsymbol{XX^T}$, i.e., the top d singular vectors of $\boldsymbol{X = U_X\Sigma_X V_X{}^T}$, which are the first d columns of $\boldsymbol{U_X}$. It then follows from (2.27) that $\boldsymbol{Y = [y_1, y_2, \ldots . y_N] = U^TX = U^TU_X\Sigma_X V_X{}^T = \Sigma V^T}$, where $\boldsymbol{\Sigma}$ is a diagonal matrix whose diagonal entries are the top d singular values of X and V a matrix whose columns are the top d right singular vectors of X. Finally, since $\boldsymbol{\Lambda = U^TU_X\Sigma^2{}_X V_X{}^T = \Sigma^2}$ the optimal least-squares error is given by trace$(\boldsymbol{\Sigma^2{}_X})$-trace$(\boldsymbol{\Sigma^2})$= $\boldsymbol{\Sigma^D{}_{i=d+1}\,\sigma_i{}^2}$, where $\boldsymbol{\sigma_i}$ is the i-th singular value of X. ■

According to the theorem, the SVD gives an optimal solution to the PCA problem. The resulting matrix U, together with the $\boldsymbol{\mu}$ if the data do not have zero mean, provides a geometric description of the dominant subspace structure for all the points, and the columns of the matrix $\Sigma V^T = [\hat{y}_1, \hat{y}_2, \ldots . \hat{y}_N] \in \mathbb{R}^{d\,x\,N}$,i.e. the principal components, give a more compact representation for the points $X = [x_1, x_2, \ldots . x_N] \in \mathbb{R}^{D\,x\,N}$, since d is typically much smaller than D.

Theorem 2.4 (Equivalence of Geometric and Sample Principal Components).

Let $X = [x_1, x_2, \ldots . x_N] \in \mathbb{R}^{D\,x\,N}$ be the mean-subtracted data matrix. The vectors $[\hat{u}_1, u_2, \ldots . u_d] \in \mathbb{R}^D$ associated with the d sample principal components of X are exactly the columns of the matrix U$\in \mathbb{R}^{D\,x\,d}$ that minimizes the least-squares error(2.29).

Proof. Notice that if X has the singular value decomposition $X = U_X \Sigma_X {V_X}^T$ then, $XX^T = U_X {\Sigma^2}_X {U_X}^T$ is the eigenvalue decomposition of XX^T. If Σ_X is ordered, then the first d columns of U_X are exactly the leading d eigenvectors of XX^T, which give the d sample principal components. ■

Theorem (2.4) shows that both the geometric and statistical formulations of PCA lead to exactly the same solution/estimate of the sample principal components. This equivalence is part of the reason why PCA has become the tool of choice for dimensionality reduction, since the optimality of the solution can be interpreted either statistically or geometrically in different application contexts. Figure 2.1 gives an example of a two-dimensional data set and its two principal components

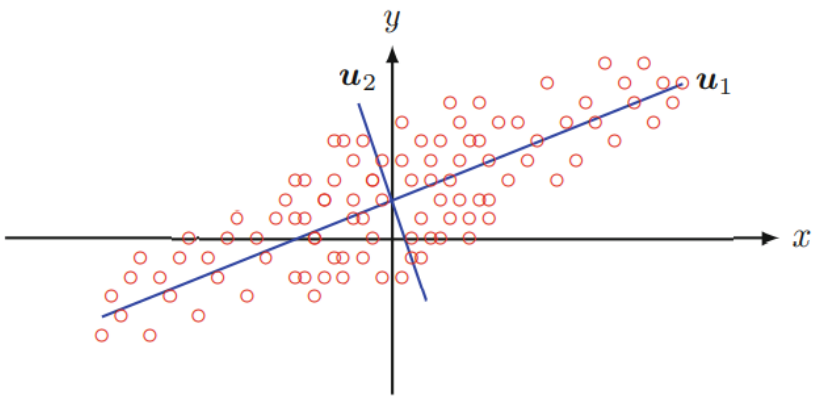


Fig. 2.1 Example showing a two-dimensional data set and its two principal components.

3)PCA algorithm.

Principal component analysis can be broken down into the following five steps:

i) Standardize the range of continuous initial variables.
ii) Compute the covariance matrix to identify correlations.
iii) Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components.
iv) Create a feature vector to decide which principal components to keep.
v) Recast the data along the principal components axes.

STEP 1: STANDARDIZATION.

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis. The PCA algorithm is sensitive regarding the variances of the initial variables so its so important to do this step. Mathematically transforming the data to comparable scale is done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

STEP 2: COVARIANCE MATRIX COMPUTATION.

The aim of this step is to understand how the input variables of the input data set are varying from the mean with respect to each other, in other word to check if there is any relationship between them. Because sometimes, variables are highly correlated in a such way that they contain redundant information. In order to identify these correlations, we compute the covariance matrix.

The covariance matrix is n x n symmetric matrix (where n is the number of dimensions) that has an entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set and variables x, y and z we get the following matrix:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Note 1: since the covariance of a variable with itself is its variance, (Cov(x,x)=Var(x)), we get that on the main diagonal we have the variances of each initial variable.

Note 2: since the covariance is commutative (Cov(x,y)=Cov(y,x)) we get that the entries of the covariance matrix are symmetric with respect to the main diagonal.

What do the covariance matrix tell us about the correlations between the variables ?

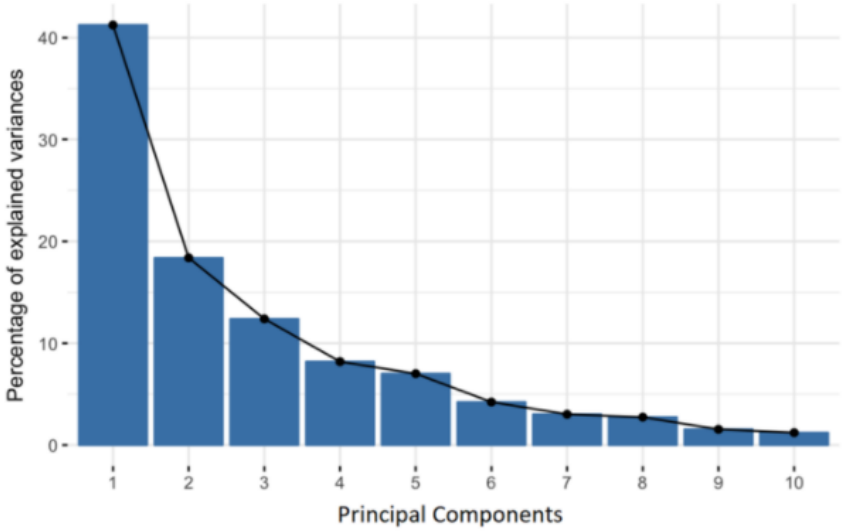
The sign of the covariance that matters:

- If positive: the two variables increase or decrease together(correlated).
- If negative: if one increases then the other decreases(inversely correlated).

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS.

In order to determine the principal components of the data we need to compute the eigenvectors and eigenvalues. But before that lets understand what do we mean by principal components.

Principal components are the new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in a such way that the new variables are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So the idea is 10-diemeonsonal data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then the maximum remaining information in the second one and so on. So we will have something like :



Note 1: Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

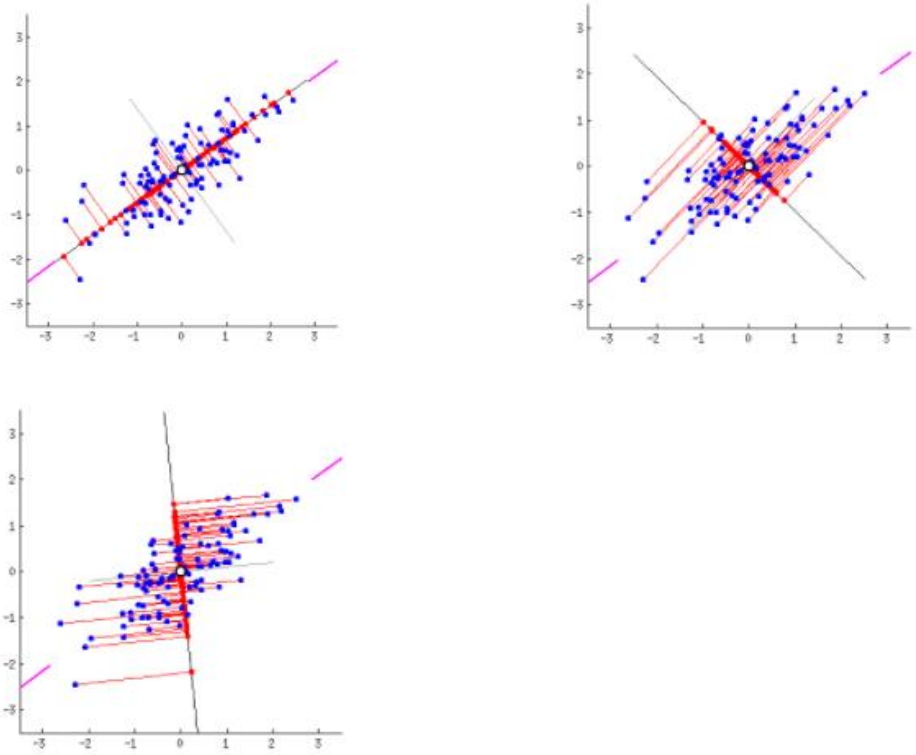
Note 2: The principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables.

Note 3 (geometric explanation): principal components represent the directions of the data that explain a maximal amount of variance (the lines that capture most information of the data). The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. In simpler words,

we can think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

How PCA constructs the principal components?

principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. For example, let's assume that the scatter plot of our data set is as shown below, can we guess the first principal component ? Yes, it's approximately the line that matches the purple marks because it goes through the origin and it's the line in which the projection of the points (red dots) is the most spread out. Or mathematically speaking, it's the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).



- The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

- This continues until a total of p principal components have been calculated, equal to the original number of variables.

Now, we are able to talk about the eigenvectors and eigenvalues. eigenvectors and eigenvalues who are behind all the magic explained above, because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance (most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component. By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

Example (Eigenvectors and Eigenvalues).

Let's suppose that our data set is 2-dimensional with 2 variables x,y and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

If we rank the eigenvalues in descending order, we get λ1>λ2, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) isv2.
After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

STEP 4: FEATURE VECTOR.

In this step we choose whether to keep all of the components or discard those of lesser significance(of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature Vector. So the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction.

Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors v1 and v2:

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

Or discard the eigenvector v2, which is the one of lesser significance, and form a feature vector with v1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

Discarding the eigenvector v2 will reduce dimensionality by 1, and will consequently cause a loss of information in the final data set. But given that v2 was carrying only 4% of the information, the loss will be therefore not important and we will still have 96% of the information that is carried by v1.

STEP 5: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES.

Note that in the previous steps we didn't make any changes on the data, we just selected the principal components and form the feature vector and the input data set remains in terms of the original axes. The aim of this step is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

References

1) Rene' Vidal, Yi Ma, S.Shankar Sastry, Generalized Principal Components Analysis.
2) Madhav Samariya, Mathematics Behind PCA:https://medium.com/analytics-vidhya/mathematics-behind-principal-component-analysis-pca-1cdff0a808a9
3) Sera Giz Ozel, The Math Behind: Everything about Principle Component Analysis(PCA):https://www.datadriveninvestor.com/2021/04/08/the-math-behind-everything-about-principle-component-analysis-pca/
4) Dakshya Mishra, PCA algorithm:https://iq.opengenus.org/algorithm-principal-component-analysis-pca/
5) http://www.billconnelly.net/?p=697
6) Keboola site, A Guide to PCA FOR Machine Learning:https://www.keboola.com/blog/pca-machine-learning
7) Rina Buoy, Introduction to PCA:https://towardsdatascience.com/introduction-to-principle-component-analysis-d705d27b88b6
8) Akash Dubey, The Mathematics Behind PCA: https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643
9) Himanshu Kunwar, Simplifying Maths behind PCA:https://www.analyticsvidhya.com/blog/2021/05/simplifying-maths-behind-pca/
10) Himanshu Kunwar, Demystifying the working of PCA:https://www.analyticsvidhya.com/blog/2021/05/demystifying-the-working-of-principal-component-analysis/