Lecture 15: Principal Components and Eigenfaces

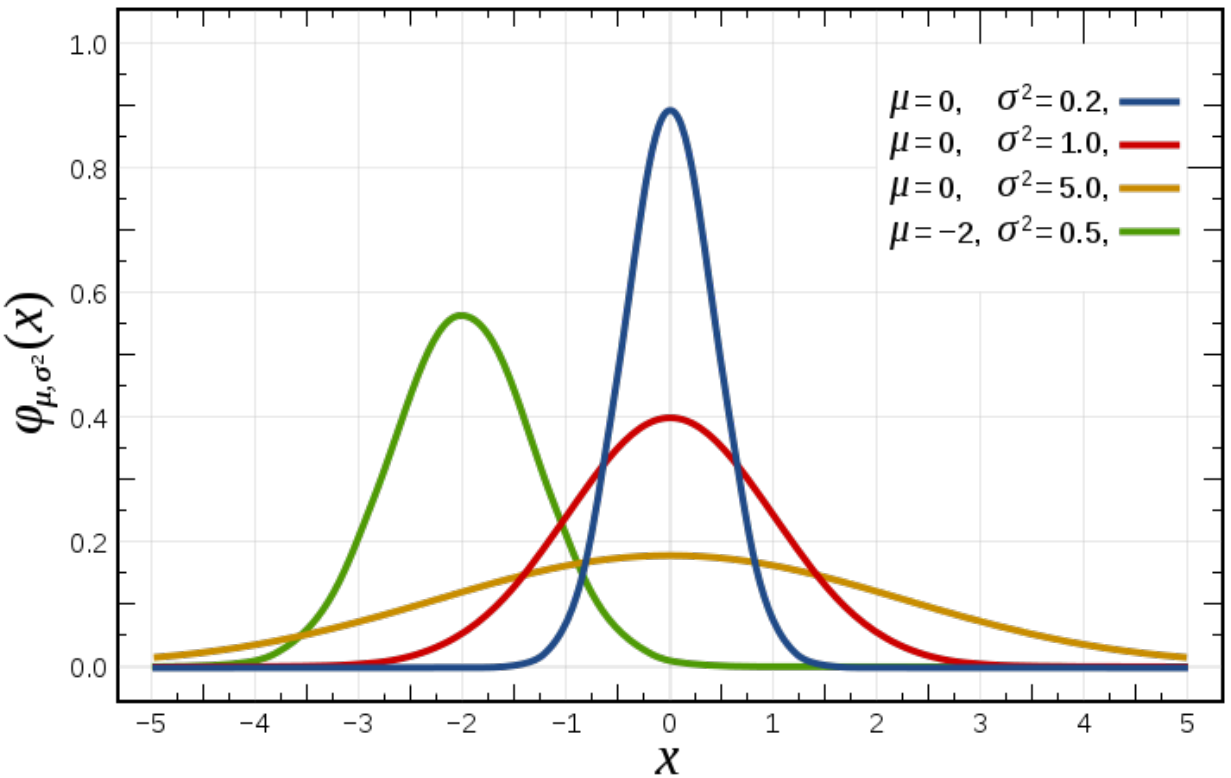
Mark Hasegawa-Johnson

ECE 417: Multimedia Signal Processing

Outline

Scalar Gaussian random variables

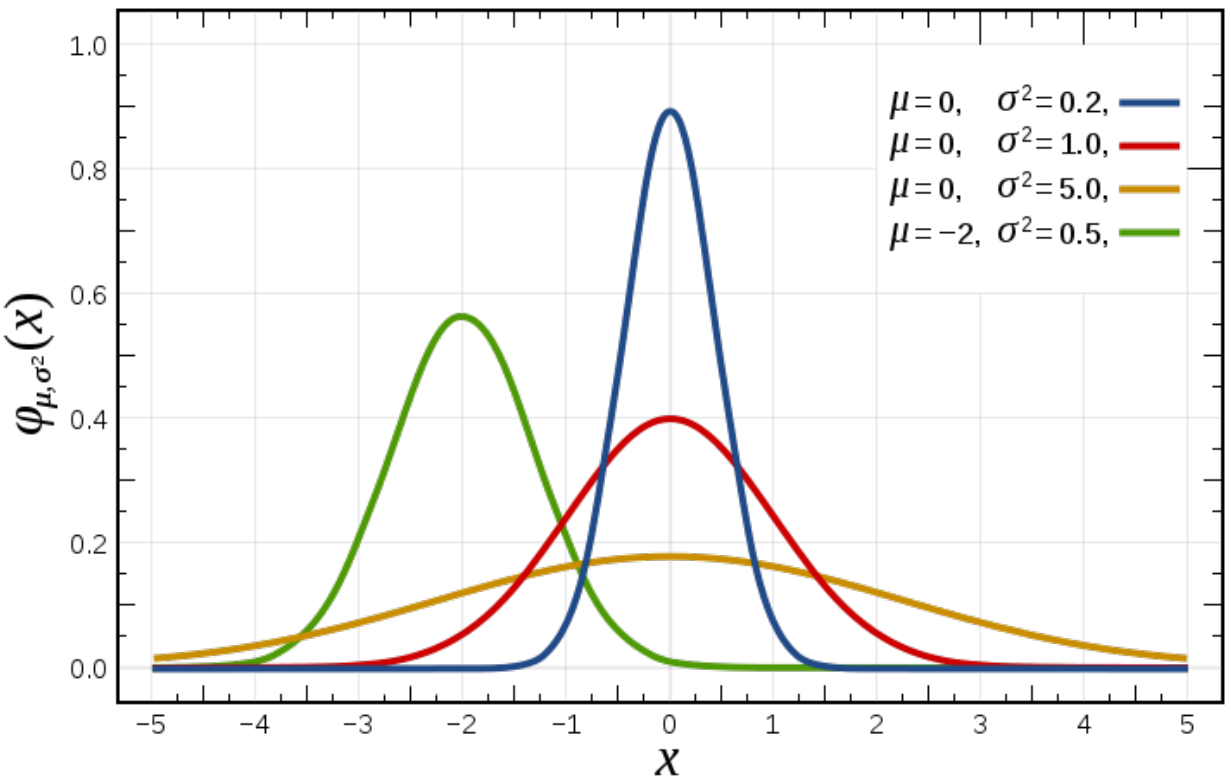
$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg

Scalar Gaussian random variables

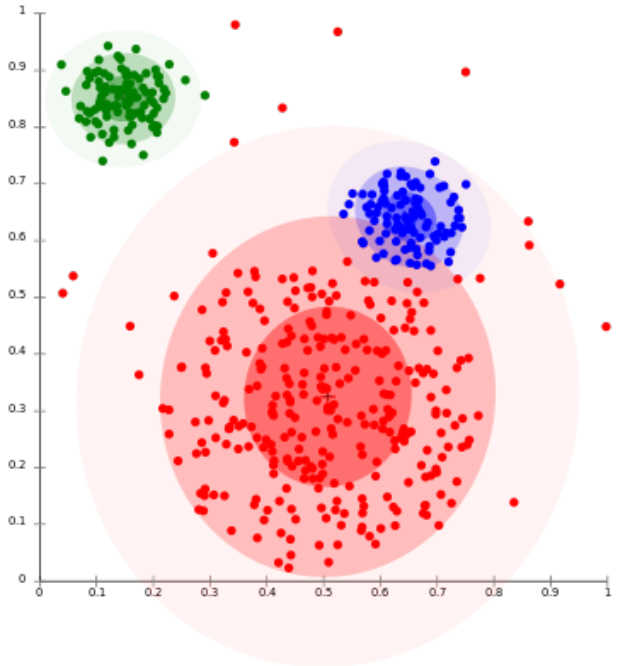
$$\mu = E[X], \quad \sigma^2 = E[(X - \mu)^2]$$



https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg

Gaussian random vectors

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$



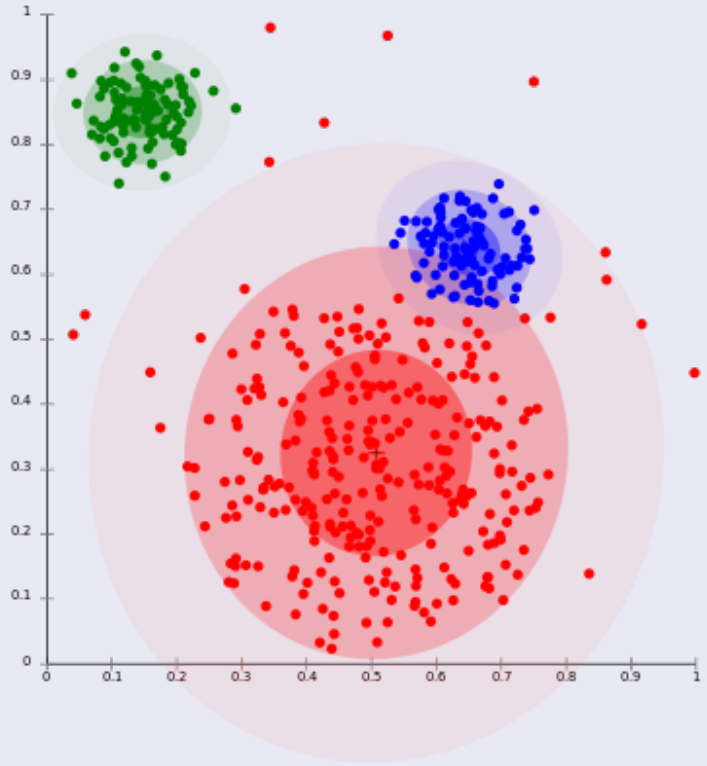
https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

Gaussian random vectors

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \cdots \\ x_D \end{bmatrix}$$

$$\boldsymbol{\mu} = E[\mathbf{x}] = \begin{bmatrix} \mu_1 \\ \cdots \\ \mu_D \end{bmatrix}$$

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

Gaussian random vectors

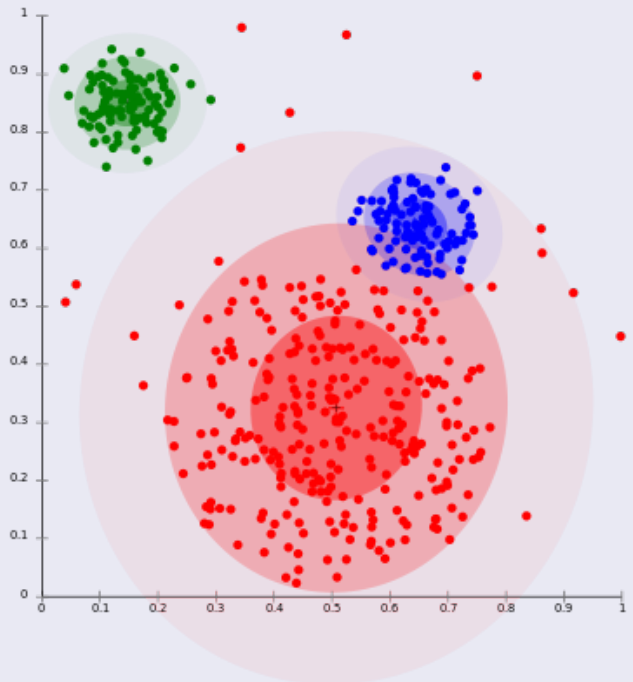
If the Gaussians are independent but not identical, then:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots \\ 0 & \sigma_2^2 & \cdots \\ \vdots & \vdots & \sigma_D^2 \end{bmatrix}$$

where

$$\sigma_i^2 = E[(x_i - \mu_i)^2]$$

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

Maximum Likelihood Parameter Estimation

In the real world, we don't know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$!

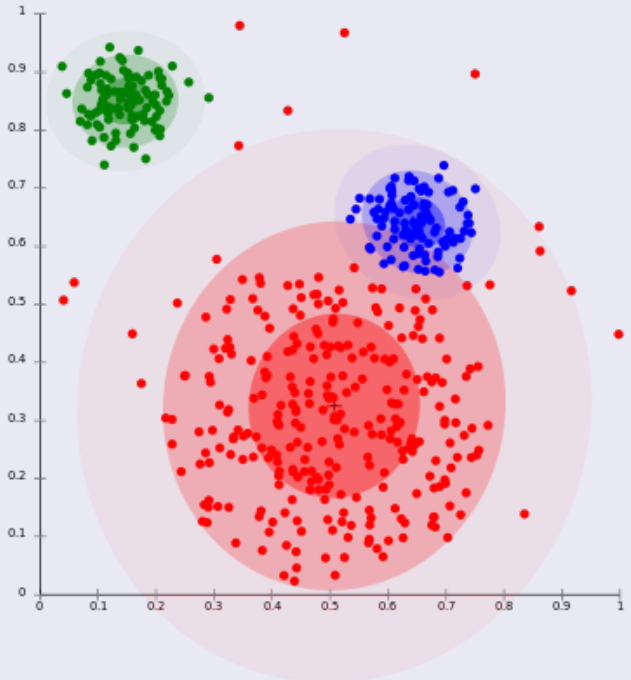
If we have a training database $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, we can estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ according to

$$\left\{\hat{\mu}_{ML}, \hat{\boldsymbol{\Sigma}}_{ML}\right\}$$

$$= \text{argmax} \prod_{m=1}^{M} p(\mathbf{x}_m|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \text{argmax} \sum_{m=1}^{M} \ln p(\mathbf{x}_m|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

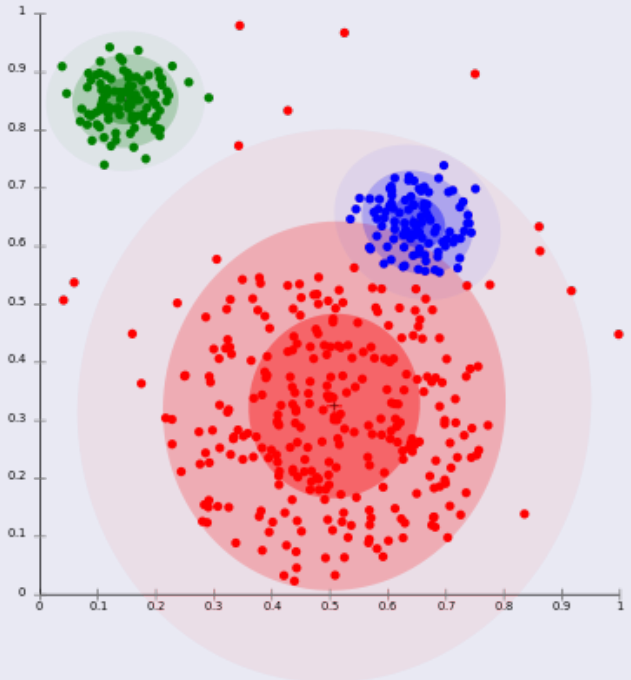
Maximum Likelihood Parameter Estimation

If you differentiate the RHS on the previous slide, and set it to zero, you find that the maximum likelihood solution is

$$\hat{\mu}_{ML} = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{x}_m$$

$$\sigma^2_{i,\mathrm{ML}} = \frac{1}{M}\sum_{m=0}^{M-1}(x_i - \mu_i)^2$$

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

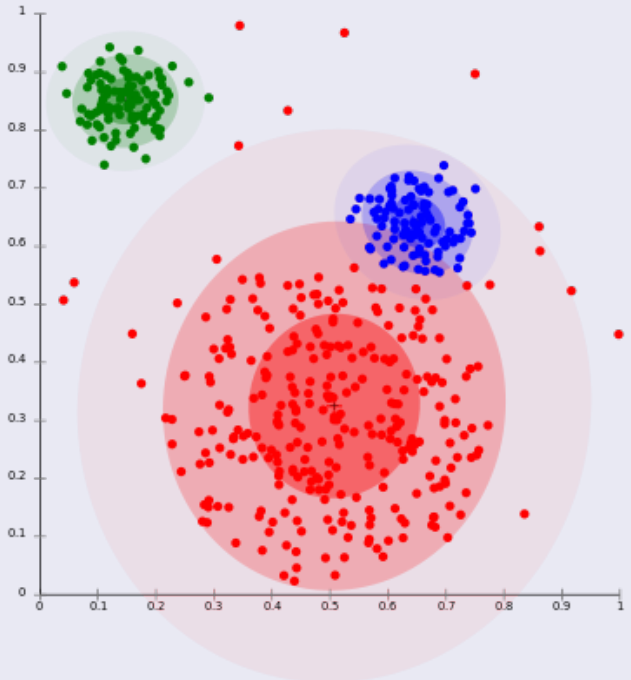
Sample Mean, Sample Variance

The ML estimate of σ_i^2 is usually too small. It is better to adjust it slightly. The following are the **unbiased estimators** of $\boldsymbol{\mu}$ and σ_i^2, also called the **sample mean** and **sample variance**:

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{x}_m$$

$$\sigma_i^2 = \frac{1}{M-1} \sum_{m=0}^{M-1} (x_i - \mu_i)^2$$

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

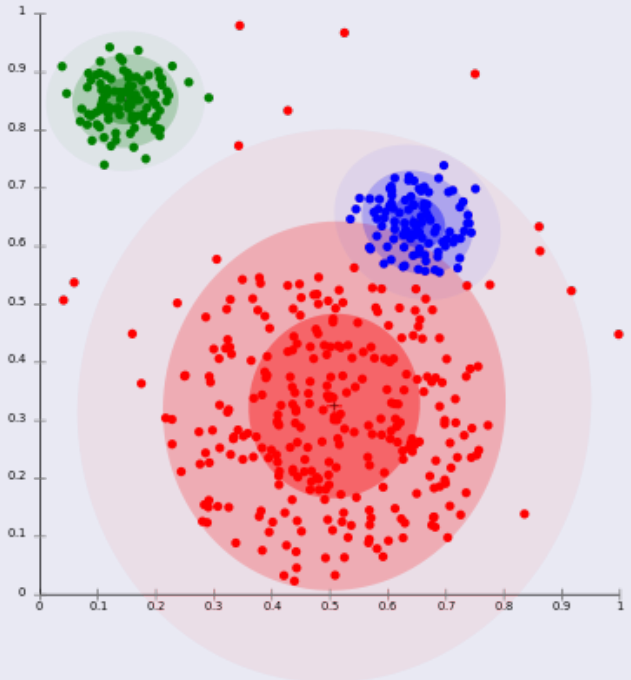
Sample Mean, Sample Variance

$$\boldsymbol{\mu} = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{x}_m$$

$$\sigma_i^2 = \frac{1}{M-1}\sum_{m=0}^{M-1}(\mathbf{x}_m-\boldsymbol{\mu})(\mathbf{x}_m-\boldsymbol{\mu})^T$$

Sample mean and sample covariance are not the same as real mean and real covariance, but we'll use the same letters ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$) unless the problem requires us to distinguish.

Example: Instances of Gaussian random vectors



https://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg

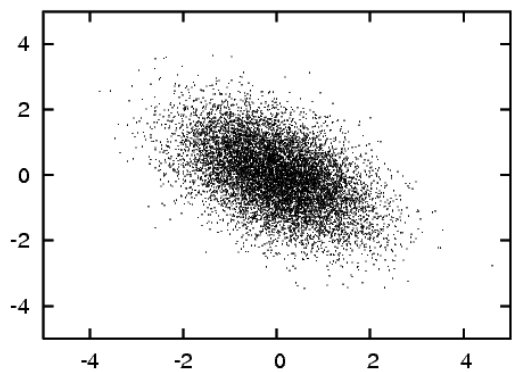
Outline

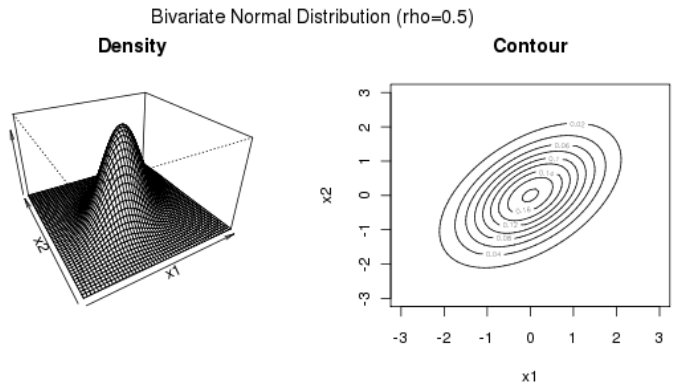
Gaussians with non-diagonal covariance matrix

If the dimensions are jointly Gaussian but **not independent** then we can still write the multivariate Gaussian as

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

. . . but now the off-diagonal elements of the covariance matrix, $\boldsymbol{\Sigma}$, are no longer zero.





https://commons.wikimedia.org/wiki/File:Multinormal_3_true.png,https://commons.wikimedia.org/wiki/File:Gaussian_copula_gaussian_marginals.png

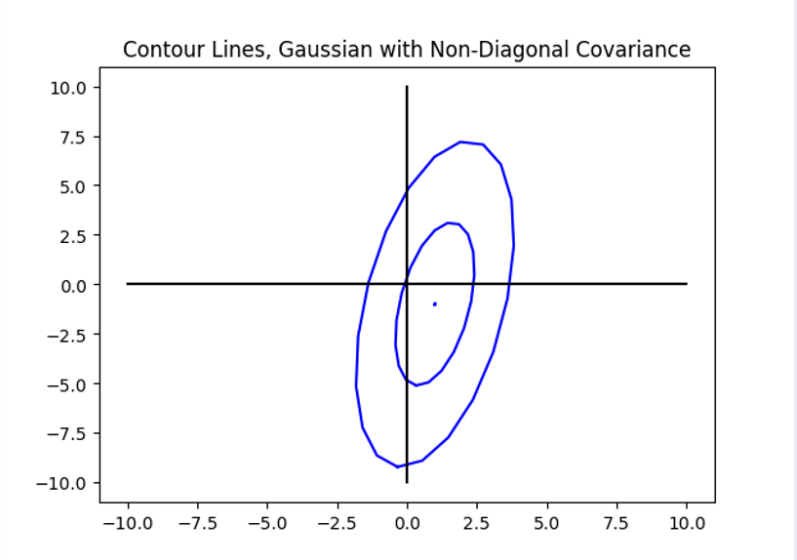
Example

Suppose that X_1 and X_2 are Gaussian RVs with means 1 and -1, variances 1 and 4, and covariance 1.

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{aligned} \sigma_1^2 &= E\left[(x_1 - \mu_1)^2\right] \\ &= 1 \\ \sigma_2^2 &= E\left[(x_2 - \mu_2)^2\right] \\ &= 4 \\ \rho_{1,2} &= E\left[(x_1 - \mu_1)(x_2 - \mu_2)\right] \\ &= 1 \end{aligned}$$

Example



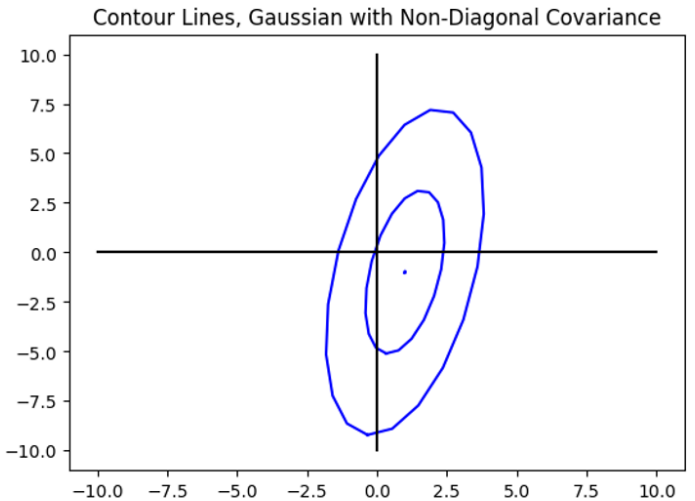
Example

Suppose that X_1 and X_2 are Gaussian RVs with means 1 and -1, variances 1 and 4, and covariance 1.

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\Sigma = E\left[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\right] = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$$

Example



Determinant and inverse of a 2×2 matrix

You should know the determinant and inverse of a 2×2 matrix. If

$$\mathbf{\Sigma} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Then $|\mathbf{\Sigma}| = ad - bc$, and

$$\mathbf{\Sigma}^{-1} = \frac{1}{|\mathbf{\Sigma}|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

If you've never done it before, please prove this formula for yourself by multiplying $\mathbf{\Sigma}^{-1}\mathbf{\Sigma}$ and verifying that the result is the identity matrix.

Example Multivariate Gaussian

The contour lines of our example are the contours along which the Mahalanobis distance is equal to a constant:

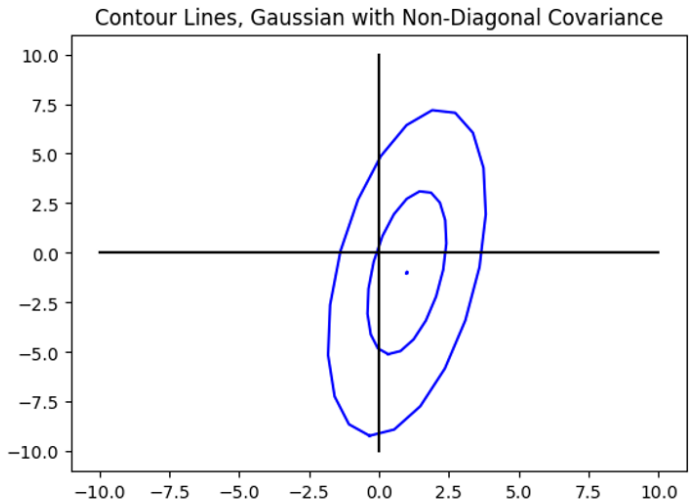
$$
\begin{aligned}
d^2_{\boldsymbol{\Sigma}}(\mathbf{x}, \boldsymbol{\mu}) &= (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\
&= \frac{1}{\sigma_1^2 \sigma_2^2 - \rho_{1,2}^2} [(x_1 - \mu_1), (x_2 - \mu_2)] \begin{bmatrix} \sigma_2^2 & -\rho_{1,2} \\ -\rho_{1,2} & \sigma_1^2 \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \\
&= \frac{\sigma_2^2}{|\boldsymbol{\Sigma}|}(x_1 - \mu_1)^2 + \frac{\sigma_1^2}{|\boldsymbol{\Sigma}|}(x_2 - \mu_2)^2 - 2\frac{\rho_{1,2}}{|\boldsymbol{\Sigma}|}(x_1 - \mu_1)(x_2 - \mu_2)
\end{aligned}
$$

This is the formula for an ellipse.

Contours of equal Mahalanobis distance are ellipses

$$d_{\Sigma}^{2}(\mathbf{x}, \boldsymbol{\mu}) = (\mathbf{x} - \boldsymbol{\mu})^{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

Example



Maximum Likelihood Parameter Estimation

In the real world, we don't know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$!
If we have a training database $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, we can estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ according to

$$\left\{\hat{\mu}_{ML}, \hat{\boldsymbol{\Sigma}}_{ML}\right\} = \text{argmax} \prod_{m=1}^{M} p(\mathbf{x}_m|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \text{argmax} \sum_{m=1}^{M} \ln p(\mathbf{x}_m|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\hat{\mu}_{ML} = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{x}_m$$

$$\hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{M} \sum_{m=0}^{M-1} (\mathbf{x}_m - \boldsymbol{\mu})(\mathbf{x}_m - \boldsymbol{\mu})^T$$

Sample Mean, Sample Covariance

The ML estimate of $\boldsymbol{\Sigma}$ is usually too small. It is better to adjust it slightly. The following are the **unbiased estimators** of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, also called the **sample mean** and **sample covariance**:

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{x}_m$$

$$\boldsymbol{\Sigma} = \frac{1}{M-1} \sum_{m=0}^{M-1} (\mathbf{x}_m - \boldsymbol{\mu})(\mathbf{x}_m - \boldsymbol{\mu})^T$$

Outline

Review: Eigenvalues and eigenvectors of a symmetric matrix

The eigenvectors and eigenvalues of a $D \times D$ square matrix, $\mathbf{A}$, are the vectors $\mathbf{u}$ and scalars λ such that

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$$
$$|\mathbf{A} - \lambda\mathbf{I}| = 0$$

If $\mathbf{A}$ is symmetric, then we can also multiply from the left:

$$\mathbf{u}^T\mathbf{A} = \lambda\mathbf{u}^T$$

Review: Positive semi-definite matrix

A positive-semidefinite matrix (we write $\mathbf{A} \succcurlyeq 0$) is one such that, for every vector $\mathbf{u} \in \Re^D$,

$$\mathbf{u}^T \mathbf{A} \mathbf{u} \geq 0$$

Every $D \times D$ matrix has D eigenvalues. A positive semi-definite matrix is also guaranteed to have D eigenvectors, though some of them may not be uniquely specified (if eigenvalues repeat, then the corresponding eigenvectors can be any orthonormal vectors spanning the corresponding subspace).

Symmetric positive semi-definite matrices: eigenvectors are orthonormal

If $\mathbf{A}$ is symmetric and positive-semidefinite, then

$$(\mathbf{u}_i^T \mathbf{A})\mathbf{u}_j = (\lambda_i \mathbf{u}_i)^T \mathbf{u}_j$$
$$\mathbf{u}_i^T (\mathbf{A}\mathbf{u}_j) = \mathbf{u}_i^T (\lambda_j \mathbf{u}_j)$$

These can only both be true if either $\lambda_i = \lambda_j$ or $\mathbf{u}_i^T \mathbf{u}_j = 0$. By defining $|\mathbf{u}_i| = 1$, we can choose eigenvectors such that

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

If we define the eigenvectors matrix as $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_D]$, then

$$\mathbf{U}^T \mathbf{U} = \mathbf{I}$$

Symmetric positive semi-definite matrices: eigenvectors diagonalize the matrix

If $\mathbf{A}$ is symmetric and positive-semidefinite, then

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_j = \mathbf{u}_i^T (\lambda_j \mathbf{u}_j) = \lambda_j \mathbf{u}_i^T \mathbf{u}_j = \begin{cases} \lambda_j, & i = j \\ 0, & i \neq j \end{cases}$$

In other words, the eigenvectors orthogonalize $\mathbf{A}$:

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{\Lambda}$$

. . . where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \lambda_D \end{bmatrix}$$

A symmetric positive semidefinite matrix is the weighted sum of its eigenvectors

The previous slide showed that $\mathbf{\Lambda} = \mathbf{U}^T\mathbf{A}\mathbf{U}$. Wrap that whole equation in $\mathbf{U} \cdots \mathbf{U}^T$, and you get:

$$\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{U}\mathbf{U}^T\mathbf{A}\mathbf{U}\mathbf{U}^T = \mathbf{A}$$

In other words, any symmetric positive semidefinite matrix can be expanded as:

$$\mathbf{A} = [\mathbf{u}_1, \ldots, \mathbf{u}_D] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \lambda_D \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_D^T \end{bmatrix} = \sum_{d=1}^{D} \lambda_d \mathbf{u}_d \mathbf{u}_d^T$$

Summary: properties of symmetric positive semidefinite matrices

If $\mathbf{A}$ is any positive semidefinite matrix, then:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

and

$$\mathbf{\Lambda} = \mathbf{U}^T\mathbf{A}\mathbf{U},$$

where $\mathbf{\Lambda}$ is diagonal and $\mathbf{U}$ is orthonormal:

$$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

The covariance matrix is symmetric

Covariance is symmetric:

$$\rho_{1,2} = E\left[(x_1 - \mu_1)(x_2 - \mu_2)\right] = \rho_{2,1}$$

. . . and therefore the covariance matrix is symmetric:

$$\begin{aligned}\boldsymbol{\Sigma} &= E\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right] \\ &= \begin{bmatrix} \sigma_1^2 & \rho_{1,2} & \cdots & \rho_{1,M} \\ \rho_{1,2} & \sigma_2^2 & \cdots & \rho_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1,M} & \rho_{2,M} & \cdots & \sigma_M^2 \end{bmatrix}\end{aligned}$$

The covariance matrix is positive semidefinite

The covariance matrix is also positive semidefinite. Here's a proof. Suppose we multiply it by any vector, $\mathbf{u}$:

$$
\begin{aligned}
&\mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u} \\
&= \mathbf{u}^T E\left[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T\right]\mathbf{u} \\
&= E\left[\mathbf{u}^T(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{u}\right] \\
&= E\left[\left(\mathbf{u}^T(\mathbf{x}-\boldsymbol{\mu})\right)^2\right] \geq 0
\end{aligned}
$$

Summary: properties of the covariance matrix

If $\mathbf{\Sigma}$ is the covariance matrix of any Gaussian, then

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

and

$$\mathbf{\Lambda} = \mathbf{U}^T\mathbf{\Sigma}\mathbf{U},$$

where $\mathbf{\Lambda}$ is diagonal and $\mathbf{U}$ is orthonormal:

$$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

Outline

How do you classify an image?

Suppose we have a test image, $\mathbf{x}_{\text{test}}$. We want to figure out: who is this person?

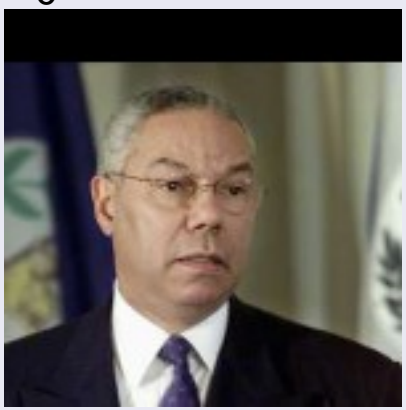
Test Datum $\mathbf{x}_{\text{test}}$:

Training Data?

In order to classify the test image, we need some training data. For example, suppose we have the following four images in our training data. Each image, $\mathbf{x}_m$, comes with a label, ℓ_m, which is just a string giving the name of the individual.

Training Datum:
ℓ_1 =**Colin Powell:**

$\mathbf{x}_0$ =



Training Datum
ℓ_2 =**Gloria Arroyo:**

$\mathbf{x}_1$ =

Training Datum
ℓ_3 =**Megawati Sukarnoputri:**

$\mathbf{x}_2$ =

Training Datum
ℓ_4 =**Tony Blair:**

$\mathbf{x}_3$ =

Nearest Neighbors Classifier

A "nearest neighbors classifier" makes the following guess: the test vector is an image of the same person as the closest training vector:

$$\hat{\ell}_{\text{test}} = \ell_{m^*}, \quad m^* = \operatorname*{argmin}_{m=1}^{M} \|\mathbf{x}_m - \mathbf{x}_{\text{test}}\|$$

where "closest," here, means Euclidean distance:

$$\|\mathbf{x}_m - \mathbf{x}_{\text{test}}\| = \sqrt{\sum_{d=1}^{D} (x_{m,d} - x_{\text{test},d})^2}$$

Improved Nearest Neighbors: Eigenface

- The problem with nearest-neighbors is that subtracting one image from another, pixel-by-pixel, results in a measurement that is dominated by noise.
- We need a better measurement.
- The solution is to find a signal representation, $\mathbf{y}_m$, such that $\mathbf{y}_m$ summarizes the way in which $\mathbf{x}_m$ differs from other faces.
- If we find $\mathbf{y}_m$ using principal components analysis, then $\mathbf{y}_m$ is called an "eigenface" representation.

Outline

Sample covariance

Remember that the sample covariance is defined as:

$$\begin{aligned}\boldsymbol{\Sigma} &= \frac{1}{M-1}\sum_{m=0}^{M-1}(\mathbf{x}_m - \boldsymbol{\mu})(\mathbf{x}_m - \boldsymbol{\mu})^T \\ &= \frac{1}{M-1}\mathbf{X}\mathbf{X}^T\end{aligned}$$

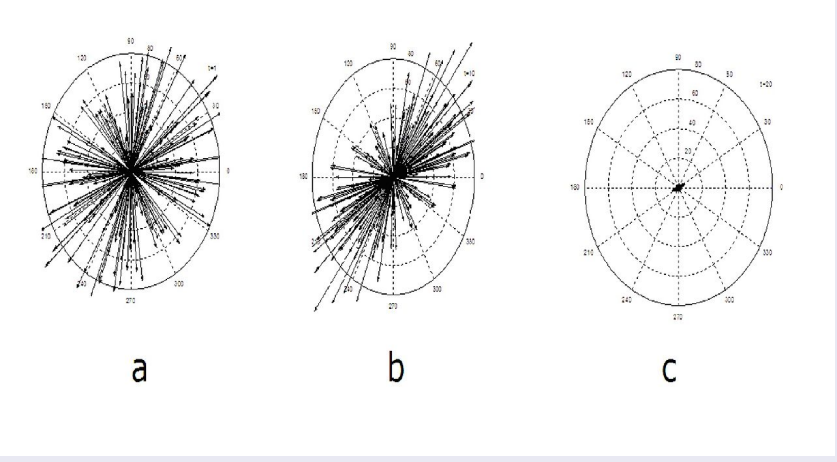
. . . where $\mathbf{X}$ is the centered data matrix,

$$\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}, \ldots, \mathbf{x}_M - \boldsymbol{\mu}]$$

Centered data matrix

$$\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}, \ldots, \mathbf{x}_M - \boldsymbol{\mu}]$$

Examples of $\mathbf{x}_m - \boldsymbol{\mu}$

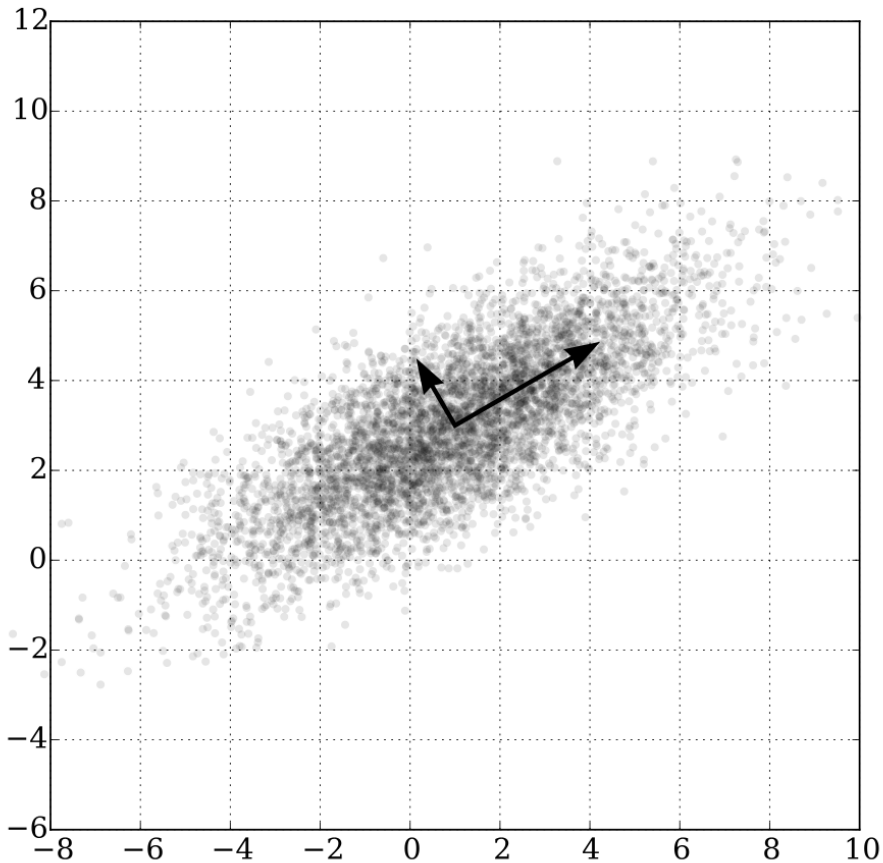


https://commons.wikimedia.org/wiki/File:PCA_Pires.jpg

Principal component axes

The eigenvectors of the sample covariance are called the principal component axes, or principal component directions.

Principal component axes



https://commons.wikimedia.org/wiki/File:GaussianScatterPCA.svg

Principal components

The principal component analysis of $\mathbf{x}_m$ is the vector $\mathbf{y}_m = \mathbf{U}^T(\mathbf{x}_m - \boldsymbol{\mu})$, where $\mathbf{U}$ are the eigenvectors of the covariance. We can compute the principal components analysis (PCA) of every vector in the training dataset by computing

$$\begin{aligned}\mathbf{Y} &= [\mathbf{y}_1, \ldots, \mathbf{y}_M] \\ &= \mathbf{U}^T\mathbf{X}\end{aligned}$$

Now let's ask: what is the sample covariance of the PCA vectors? The sample covariance is defined as:

$$\begin{aligned}\frac{1}{M-1}\mathbf{Y}\mathbf{Y}^T &= \frac{1}{M-1}\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U} \\ &= \mathbf{U}^T\boldsymbol{\Sigma}\mathbf{U} \\ &= \boldsymbol{\Lambda}\end{aligned}$$

So the covariance of $\mathbf{Y}$ is a diagonal matrix, $\boldsymbol{\Lambda}$.

Principal components

The vector $\mathbf{y}_m$ is called the **principal components analysis** (PCA) of $\mathbf{x}_m$. Let's examine its structure a little.

$$\mathbf{y}_m = \mathbf{U}^T(\mathbf{x}_m - \boldsymbol{\mu}) = \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_D^T \end{bmatrix} (\mathbf{x}_m - \boldsymbol{\mu}) = \begin{bmatrix} \mathbf{u}_1^T(\mathbf{x}_m - \boldsymbol{\mu}) \\ \vdots \\ \mathbf{u}_D^T(\mathbf{x}_m - \boldsymbol{\mu}) \end{bmatrix}$$

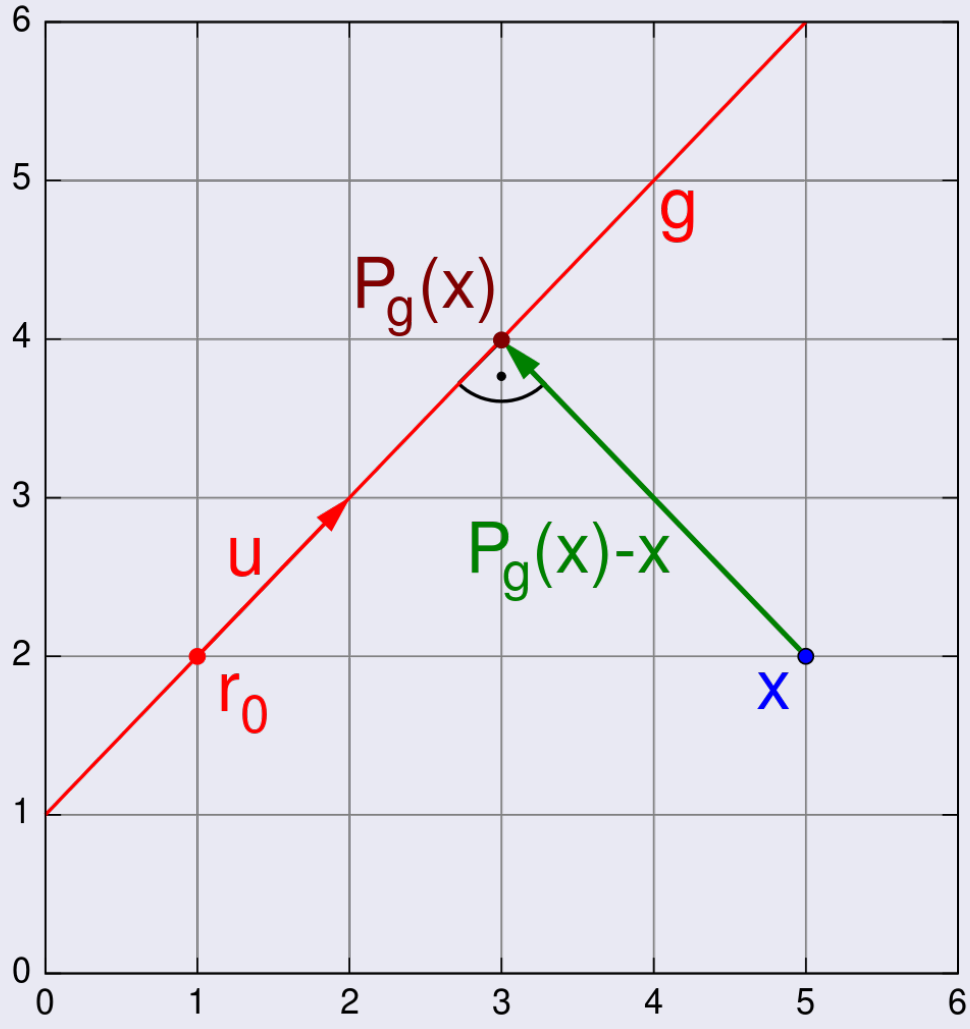
We can say that the j^{th} principal component of $\mathbf{x}_m$ is

$$y_{j,m} = \mathbf{u}_j^T(\mathbf{x}_m - \boldsymbol{\mu})$$

Orthogonal projection

Remember that the eigenvectors were defined to have unit length, therefore $y_{j,m} = \mathbf{u}_j(\mathbf{x}_m - \boldsymbol{\mu})$ is the orthogonal projection of $\mathbf{x}_m - \boldsymbol{\mu}$ onto the j^{th} principal component direction.

Principal component axes

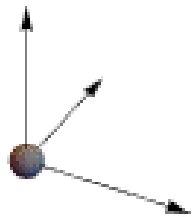


https://commons.wikimedia.org/wiki/File:Orthogonal_Projection_qtl1.svg

Principal components = Orthonormal projection onto principal component directions

Remember that the eigenvector matrix is orthonormal, therefore $\mathbf{y} = \mathbf{U}^T(\mathbf{x}_m - \boldsymbol{\mu})$ is just an expression of $\mathbf{x}_m - \boldsymbol{\mu}$ in a new set of axes. This operation is sometimes called "rotation."

$$\begin{pmatrix} -5.20 & -2.28 & 7.06 \\ -2.28 & 6.22 & -3.51 \\ 7.06 & -3.51 & -0.09 \end{pmatrix}$$



https://commons.wikimedia.org/wiki/File:Diagonalization_as_rotation.gif

The principal components are linearly independent

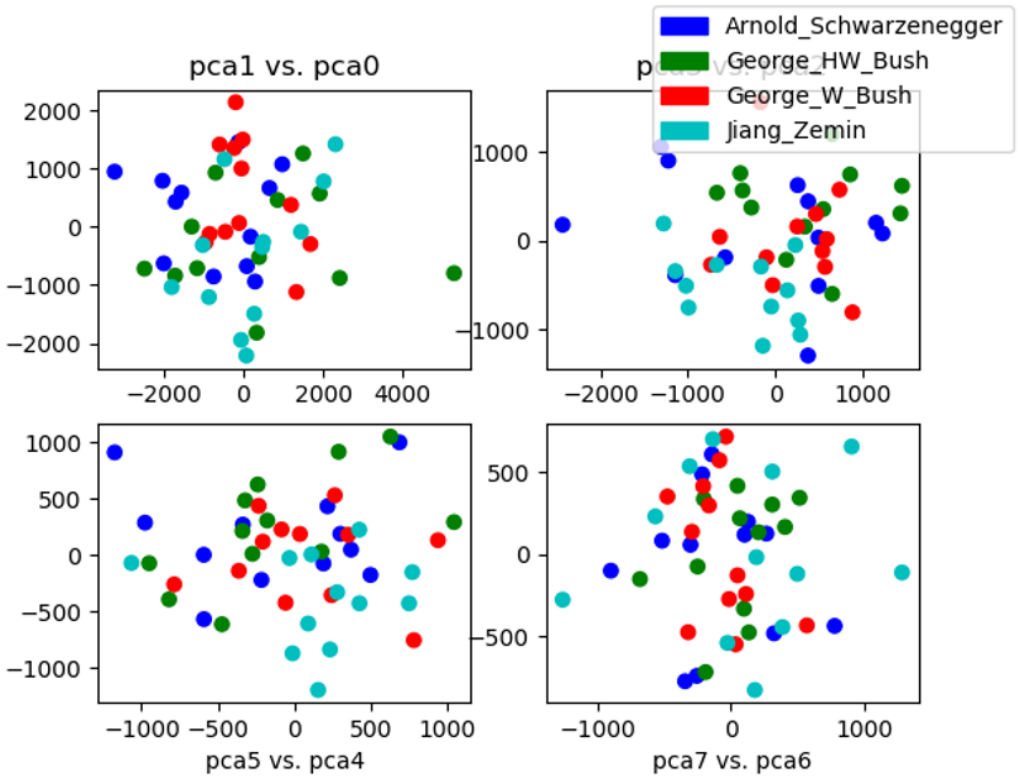
Suppose that $\mathbf{x}$ is a Gaussian random vector:

$$
\begin{aligned}
p_X(\mathbf{x}) &= \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \\
&= \frac{1}{\sqrt{|2\pi\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T(\mathbf{x}-\boldsymbol{\mu})} \\
&= \frac{1}{\sqrt{|2\pi\boldsymbol{\Lambda}|}} e^{-\frac{1}{2}\mathbf{y}^T\boldsymbol{\Lambda}^{-1}\mathbf{y}} \\
&= \prod_{i=1}^{D} \frac{1}{\sqrt{2\pi\lambda_i}} e^{-\frac{1}{2}\frac{y_i^2}{\lambda_i}}
\end{aligned}
$$

So if $\mathbf{x}$ is a Gaussian random vector, then the principal components are independent zero-mean Gaussian random variables with variances of λ_i.

Principal components are uncorrelated, and PC with larger eigenvalues have more energy

In the following figure, notice that (1) the principal components are uncorrelated with one another, (2) the eigenvalues have been sorted so that $\lambda_0 > \lambda_1 > \lambda_2$ and so on. With this sorting, you see that the the first PC has the biggest variance:



Outline

Feature embedding

- Suppose $\mathbf{x}$ is a vector we'd like to classify.
  - It has very high dimension, e.g., 6 million.
  - High-dimensional classifiers are hard to learn.
  - Probably most of the features are redundant.
- Let's learn an embedding $\mathbf{z} = f(\mathbf{x})$ such that:
  - $\mathbf{z}$ has low dimension, e.g., 1000
  - $\mathbf{z}$ contains the important information from $\mathbf{x}$

Measuring "importance" as "variance"

There are many ways to formulate the problem of finding the important information in $\mathbf{x}$. One of the simplest methods that works well is to find $\mathbf{z} = [z_1, \ldots, z_K]^T$, $K \ll D$, such that:

- Each z_i is just an orthogonal projection of $\mathbf{x}$, i.e., $z_i = \mathbf{f}_i^T(\mathbf{x} - \boldsymbol{\mu})$, where $|\mathbf{f}_i| = 1$.
- Each z_i is uncorrelated with all the others, $E[z_i z_j] = 0$.
- Given those constraints, it makes sense to choose z_i to capture as much of the sample variance of $\mathbf{x}$ as possible, in other words, we want:

$$\mathbf{F} = \text{argmax} \sum_{m=1}^{M} \sum_{i=1}^{K} z_{i,m}^2$$

such that: $\mathbf{z}_m = \mathbf{F}^T(\mathbf{x}_m - \boldsymbol{\mu}), \mathbf{F} \in \Re^{D \times K}, \ \mathbf{F}^T\mathbf{F} = \mathbf{I}, \sum_{m=1}^{M} z_{i,m} z_{j,m} = 0$

PCA = Maximum-variance linear feature embedding

We have decided that we want to maximize $\sum_m \sum_i z_{i,m}^2$ subject to the following constraints:

$$\mathbf{z}_m = \mathbf{F}^T(\mathbf{x}_m - \boldsymbol{\mu}), \mathbf{F} \in \Re^{D \times K}, \quad \mathbf{F}^T\mathbf{F} = \mathbf{I}, \sum_{m=1}^{M} z_{i,m} z_{j,m} = 0$$

All of these constraints are satisfied if we set the feature vectors, $\mathbf{f}_i$, equal to any subset of the eigenvectors, $\mathbf{u}_i$. The eigenvectors that maximize $\sum_m \sum_i z_{i,m}^2$ are those associated with the largest eigenvalues:

$$\frac{1}{M-1} \sum_{m=1}^{M} \sum_{i=1}^{K} z_{i,m}^2 = \sum_{i=1}^{K} \lambda_i$$

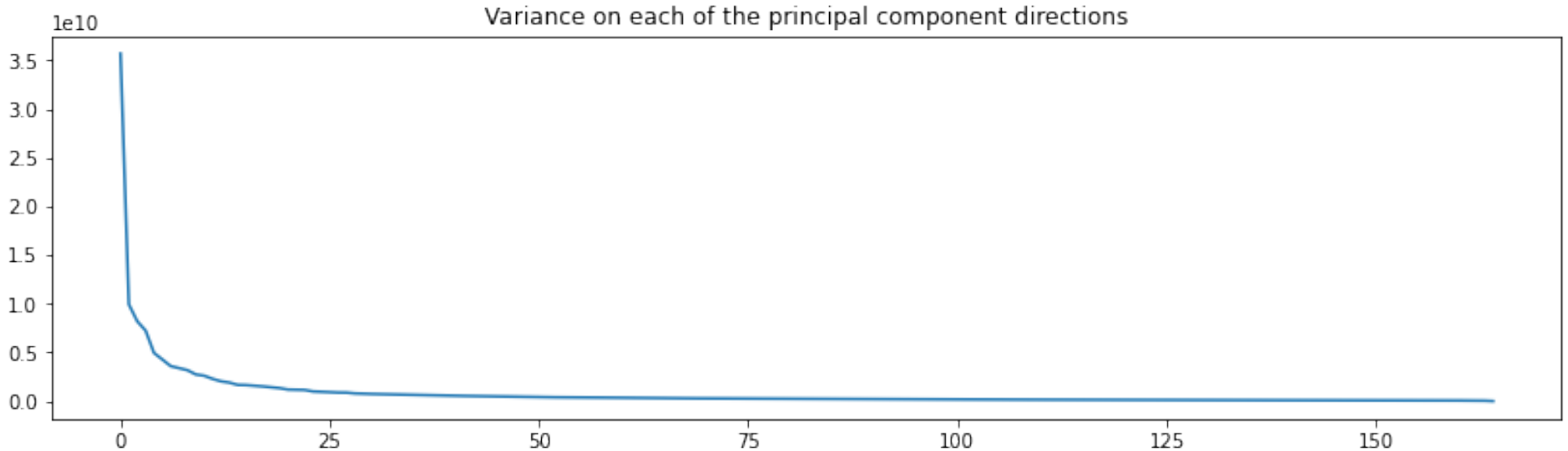
Energy spectrum=Fraction of energy explained

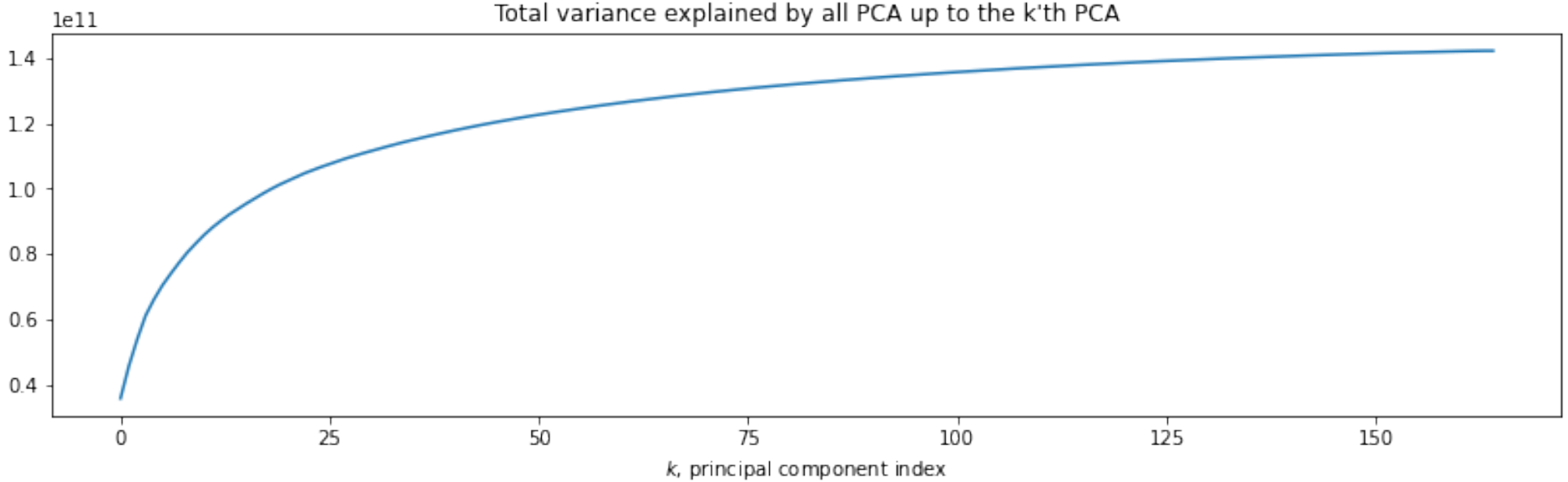
The "energy spectrum" is energy as a function of basis vector index. There are a few ways we could define it, but one useful definition is:

$$E[k] = \frac{\sum_{m=1}^{M} \sum_{i=1}^{k} y_{i,m}^2}{\sum_{m=1}^{M} \sum_{i=1}^{D} y_{i,m}^2}$$
$$= \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{D} \lambda_i}$$

In words, $E[k]$ is the fraction of the sample variance that is captured by a feature embedding consisting of the first k principal components.

Energy spectrum=Fraction of energy explained





Outline

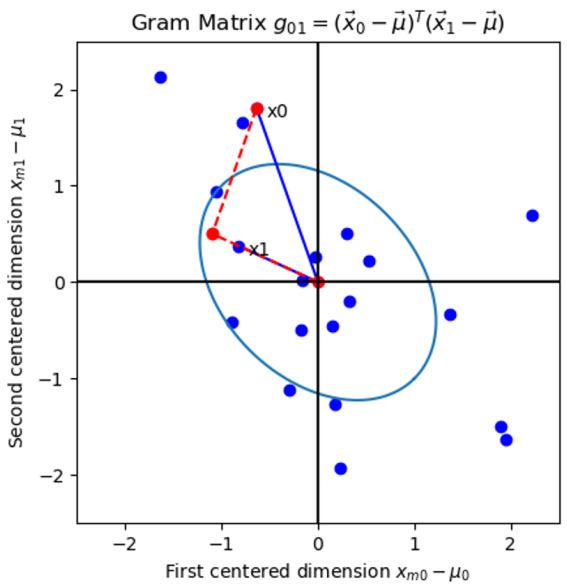
Gram matrix

- $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ is usually called the sum-of-squares matrix. $\frac{1}{M-1}\mathbf{S}$ is the sample covariance.
- $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ is called the gram matrix. Its $(i,j)^{\text{th}}$ element is the dot product between the i^{th} and j^{th} data samples:

$$g_{i,j} = (\mathbf{x}_i - \boldsymbol{\mu})^T(\mathbf{x}_j - \boldsymbol{\mu})$$

Gram matrix

$g_{01} = (\mathbf{x}_0 - \boldsymbol{\mu})^T(\mathbf{x}_1 - \boldsymbol{\mu})$



Eigenvectors of the Gram matrix

G is also symmetric and positive semidefinitex! So it has orthonormal eigenvectors:

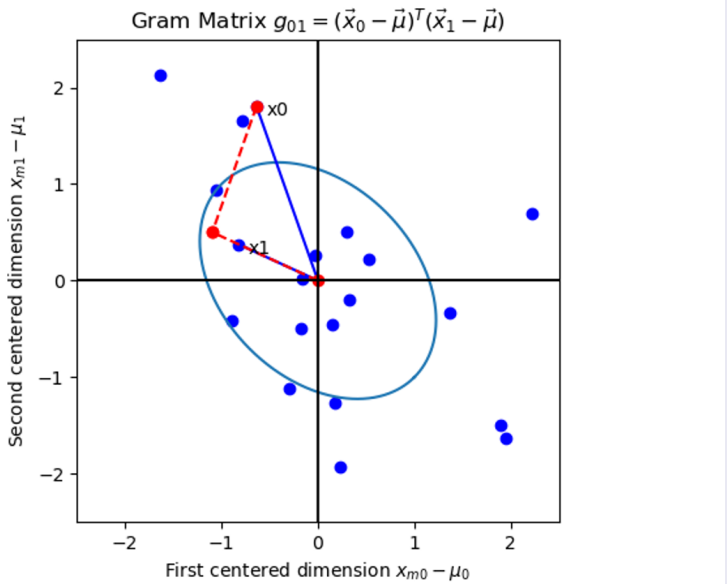
$$\mathbf{G} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

Surprising Fact: **G** and **S** have the same eigenvalues, but different eigenvectors (**U** vs. **V**). For this part of the lecture, let's say that $\mathbf{\Lambda}$ are the eigenvalues of the sum-of-squares matrix, which are also the eigenvalues of the gram matrix.

Gram matrix

$$g_{1,2} = (\mathbf{x}_1 - \boldsymbol{\mu})^T(\mathbf{x}_2 - \boldsymbol{\mu})$$



Why the Gram matrix is useful:

Suppose that $D \sim 240000$ pixels per image, but $M \sim 240$ different images. Then,

- $\mathbf{S}$ is a 240000×240000 matrix, and finding its eigenvectors is an $\mathcal{O}\left\{(240000)^3\right\}$ operation.
- $\mathbf{G}$ is a 240×240 matrix, and finding its eigenvectors is an $\mathcal{O}\left\{(240000)^3\right\}$ operation.

Singular Value Decomposition

Suppose that $\mathbf{X} \in \Re^{D \times M}$ and

- $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_D]$ are the eigenvectors of $\mathbf{G} = \mathbf{X}^T\mathbf{X}$, and λ_i are its eigenvalues, of which at most $\min(M, K)$ are nonzero.
- $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$ are the eigenvectors of $\mathbf{S} = \mathbf{X}\mathbf{X}^T$, and λ_i are its eigenvalues, of which at most $\min(M, K)$ are nonzero. Then:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{V}^T,$$

where $\mathbf{\Lambda}^{1/2}$ is a $D \times M$ diagonal matrix with the singular values, $\lambda_i^{1/2}$, on the diagonal.

How to find the principal components from the eigenvectors of the gram matrix

- Suppose you've computed the eigenvectors of the gram matrix, $\mathbf{V}$, and you want to find the principal component directions, $\mathbf{U}$. How can you do that?
- Answer: multiply by the data matrix, $\mathbf{X}$, then divide by the singular values:

$$\begin{aligned}\mathbf{X}\mathbf{V}\mathbf{\Lambda}^{-1/2} &= \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{V}^T\mathbf{V}\mathbf{\Lambda}^{-1/2} \\ &= \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{-1/2} \\ &= \mathbf{U}\end{aligned}$$

How to find the principal components directly from the data matrix, without ever computing either the sample covariance matrix or the gram matrix

Use `np.linalg.svd(X)`. This function will:

- Check whether $\mathbf{X}$ has more rows or more columns.
- Depending on the answer, find eigenvalues and eigenvectors of either $\mathbf{X}^T\mathbf{X}$ or $\mathbf{X}\mathbf{X}^T$.
- Find the other set of eigenvectors using one of the following two equations:

$$\mathbf{U} = \mathbf{X}\mathbf{V}\mathbf{\Lambda}^{-1/2}, \quad \text{or}$$
$$\mathbf{V} = \mathbf{X}^T\mathbf{U}\mathbf{\Lambda}^{-1/2}$$

Outline

Summary

- Symmetric positive semidefinite matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad \mathbf{U}^T\mathbf{A}\mathbf{U} = \mathbf{\Lambda}, \quad \mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$$

- Centered dataset:

$$\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}, \ldots, \mathbf{x}_M - \boldsymbol{\mu}]$$

- Singular value decomposition:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{V}^T$$

  where $\mathbf{V}$ are eigenvectors of the gram matrix, $\mathbf{U}$ are eigenvectors of the covariance matrix, and $\mathbf{\Lambda}$ are their shared eigenvalues.
- The principal components are the first K elements of $\mathbf{y} = \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$. Principal component analysis maximizes the variance of $\mathbf{y}$ subject to the constraints that each dimension is a linearly independent orthonormal projection of $\mathbf{x}$.