**Plan**

1. General idea of the topic
2. Multivariate Gaussian distribution
   1. Gaussian distribution
   2. Multivariate Gaussian distribution
3. Gaussian processes
   1. Matlab example
   2. Example with bounds
4. Understand the code
   1. Make a list of tools used in the code
   2. Learn Python
   3. Try to rewrite the code in Python

**General idea**

Input data: video data

Output data: generated video data

Steps

1. Dimensionality reduction

Dimensionality: N x M x F, where

N x M – resolution of a video, F – number of frames

Apply kind of PCA technic, choose top 20 features

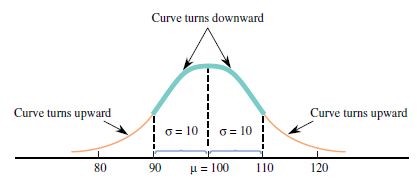
1. Learn dynamic texture

Apply Gaussian process

1. Generate new data

**Gaussian distribution**

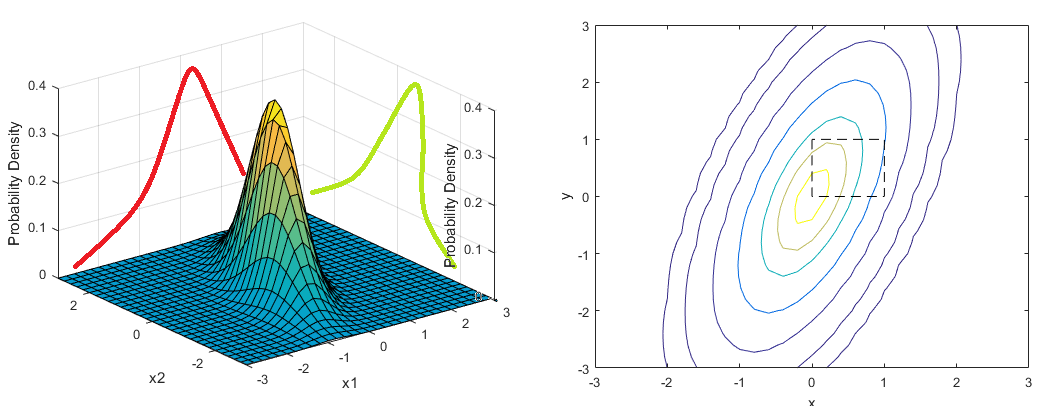
Gaussian (normal) distribution – continuous probability distribution, bell shaped and symmetric. Characterized by mean and standard deviation. Total area under the distribution curve equals to 1.



Mean – describes where corresponding curve is centered.

Standard deviation – describes how much the curve spreads out around center.

Joint distribution of two dependent/independent events

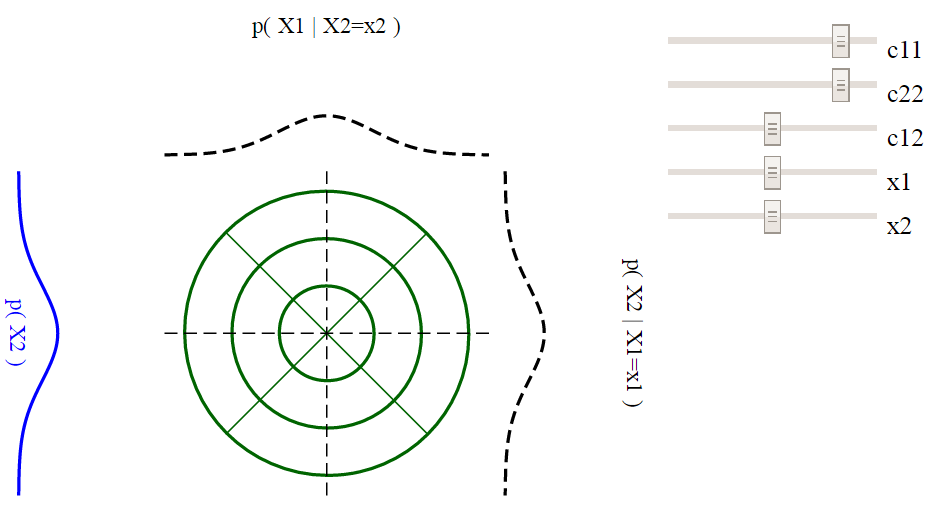


**Multivariate Gaussian distribution**

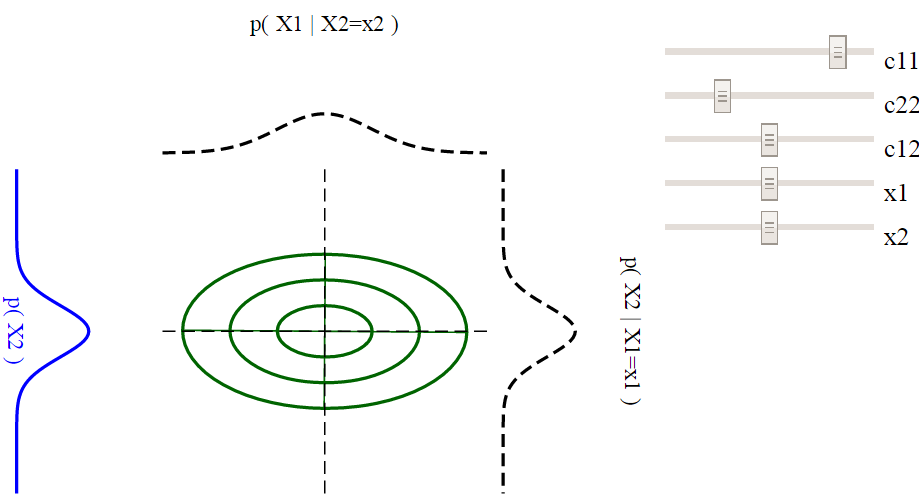
Bivariate case – for any fixed X1 value the distribution of associated X2 values is normal and for any fixed X2 value the distribution of X1 value is normal.

Multivariate case (2 or more dimensions) – characterized by:

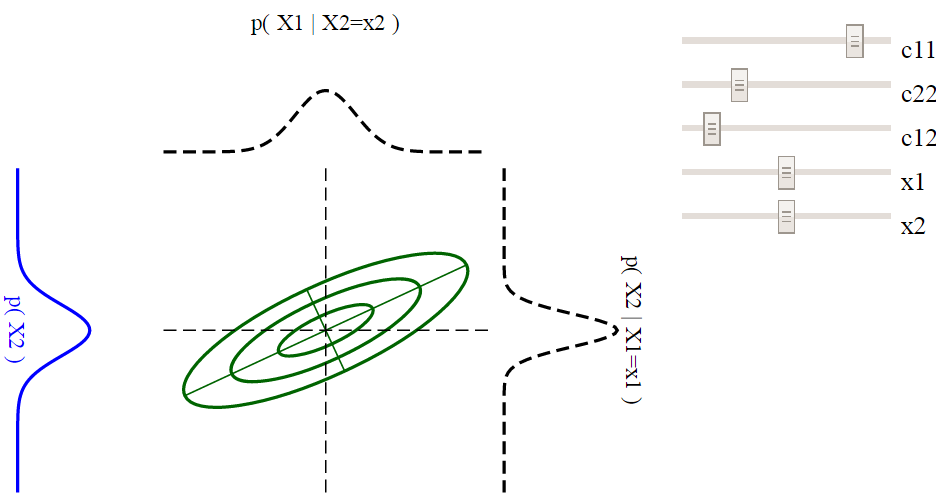
1. Mean vector – the same size as data
2. Covariance matrix – squared matrix DxD, where D – dimensionality (in bivariate case 2x2)
3. Shape of a cut (2D projection) – diagonal elements of covariance matrix are equal, sizes of directions are the same – contour is spherical



otherwise contour is shaped



also, it can be rotated as well

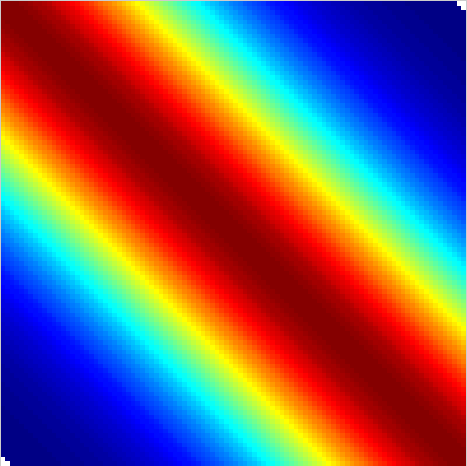


size and orientation of ellipse can be understood by looking at eigenvalues and eigenvectors of covariance matrix.

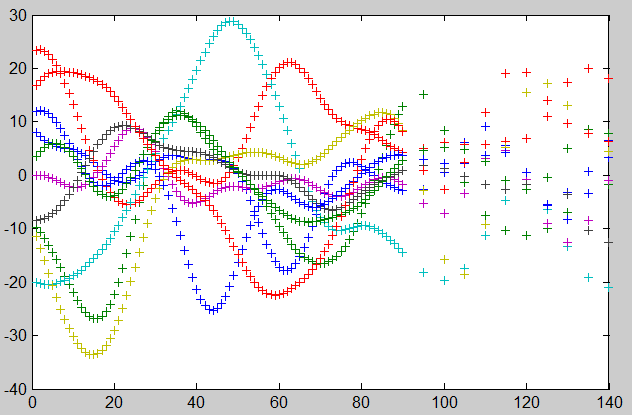
Eigenvectors shows directions, eigenvalues – scale.

**Matlab example**

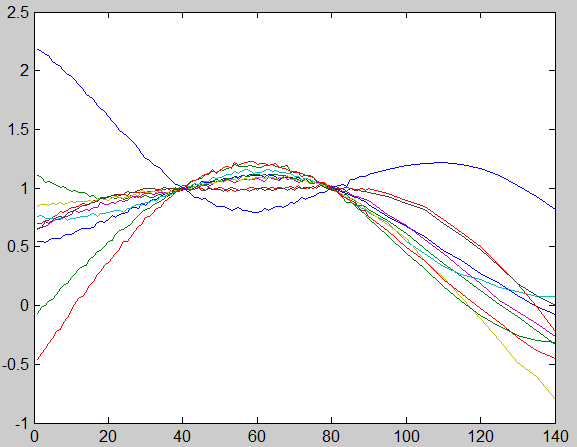
Kernel matrix



10 possible realisations with respect to this kernel matrix



**Example with bounds**



**List of tools used in the code**

*Preprocessing:*

1. Open a video file

2. Extract frames (images)

3. Resize frames (images)

*Processing:*

1. Convert image to grayscale

2. PCA

3. Matrix multiplacation

4. Kernels: linear, RBF (Radial Basis Function), Polynomial, RATQUAD, MLP (Multilayer Perceptron), Matern32

5. Creation of a kernel

6. Function initKernWeight for combined kernels defenition

7. Function weightsConstrain for hyperparameters initialization

8. Function kernExpandParam for combined kernel structure definition

9. Function updateKernWeight

10. Hyperparameters optimization function gpdm

11. Prediction and sample reconstruction

12. floor, ceil

13. reshape an array into the matrix

14. Save data back to video file