

Гаусови процеси

Когато невронните мрежи не са достатъчни

29 юли, 2017, СофтУни

Мартин Асенов

В колаборация с Проф. Subramanian Ramamoorthy



EDINBURGH CENTRE FOR
ROBOTICS

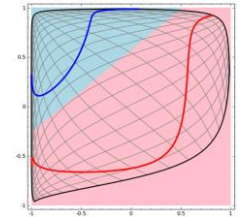
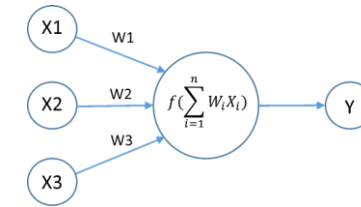
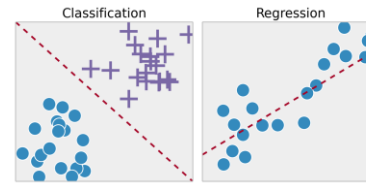


THE UNIVERSITY
of EDINBURGH

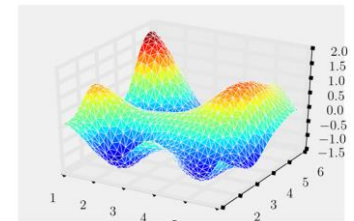
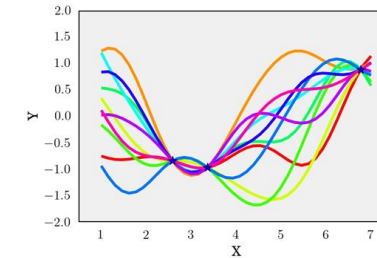
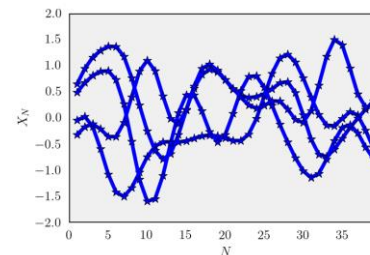
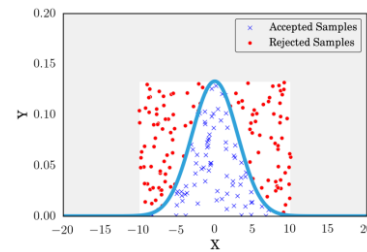


План за лекцията

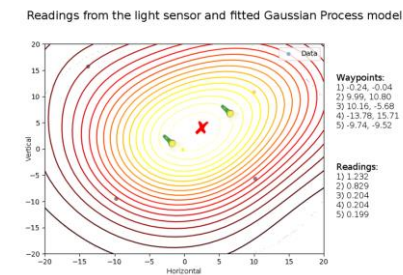
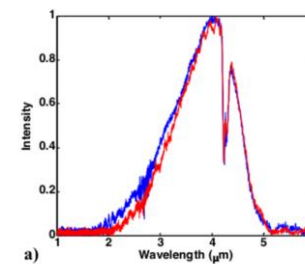
Машинно самообучение и невронни мрежи



Гаусови процеси

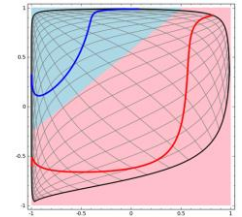
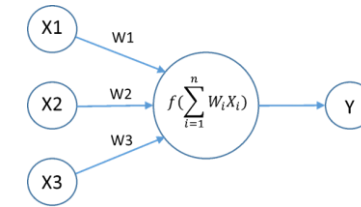
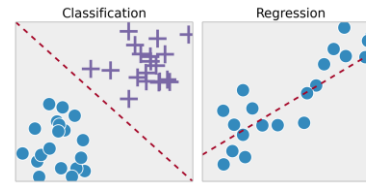


Активно семплиране с дрон използвайки Гаусови процеси за намиране на опасни газове

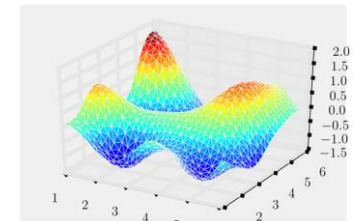
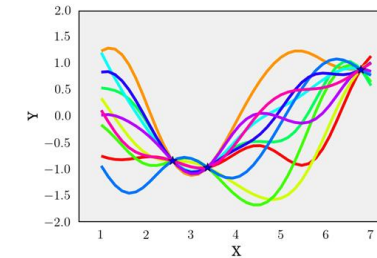
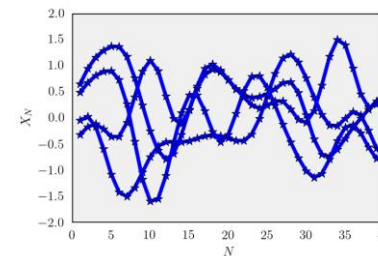
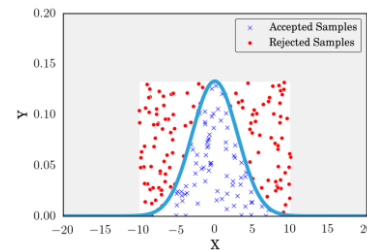


План за лекцията

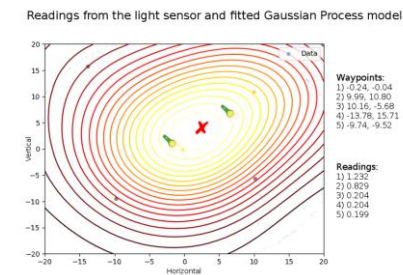
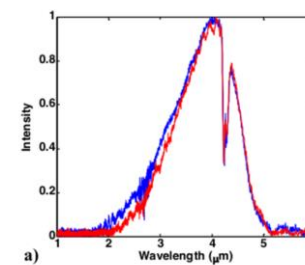
Машинно самообучение и невронни мрежи



Гаусови процеси

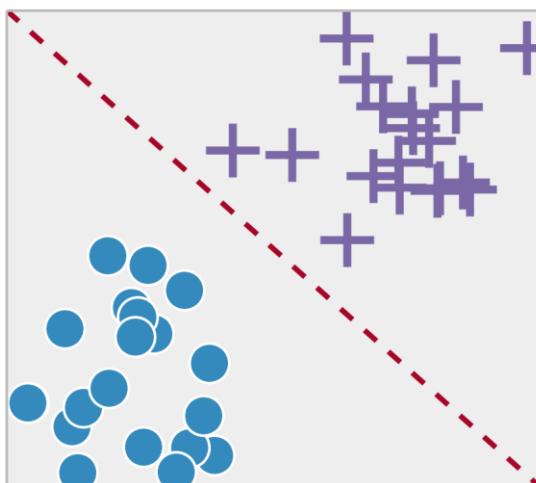


Активно семплиране с дрон използвайки Гаусови процеси за намиране на опасни газове

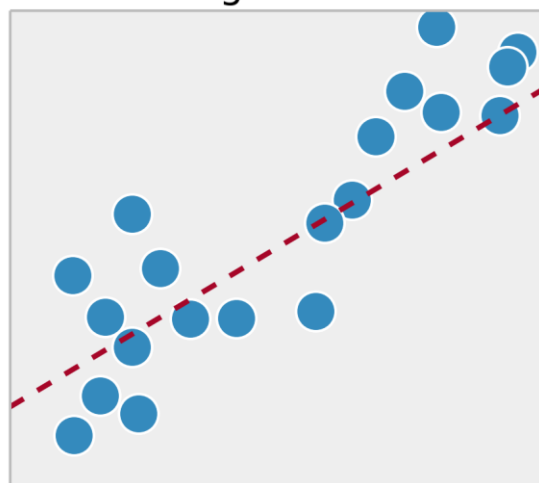


Машинно самообучение

Classification



Regression



Reinforcement Learning Framework

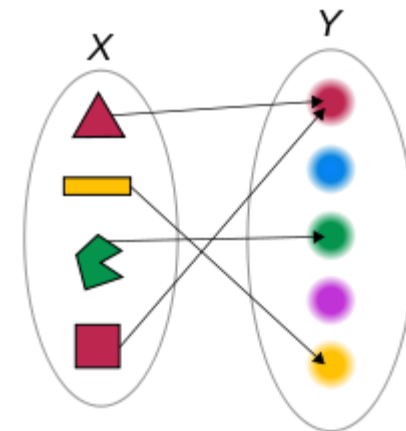
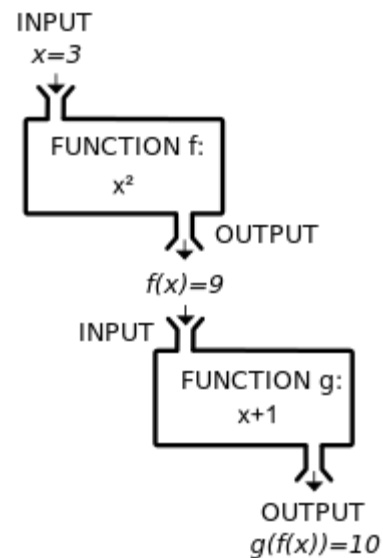
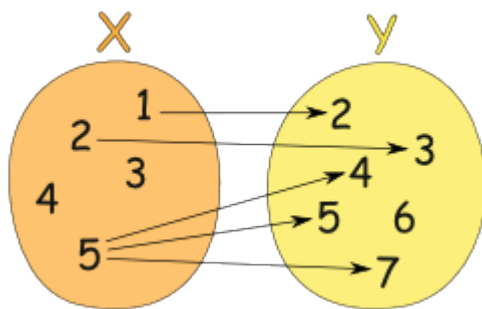


Функция

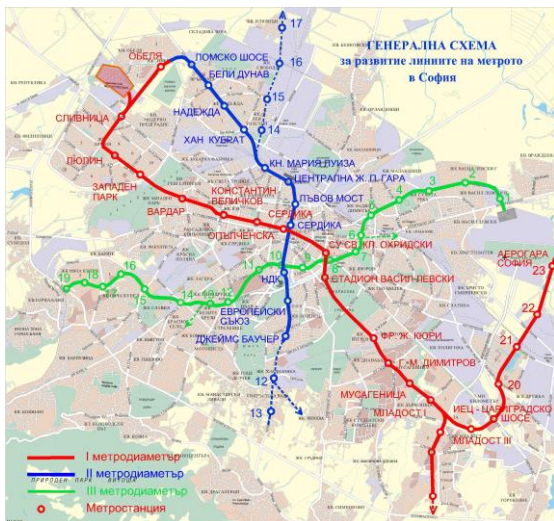
Какво е математическа функция?

Функция

Допустими стойности -> Правило -> Стойности, които функцията заема



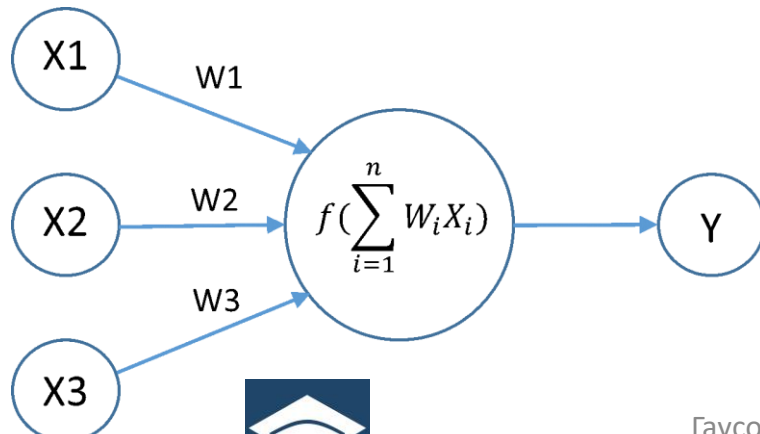
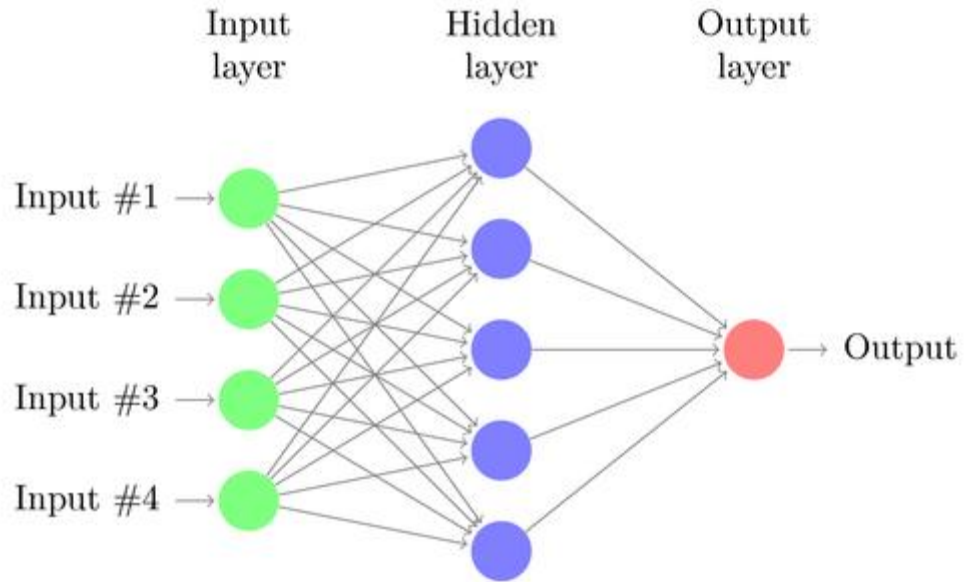
Функция



► Cat



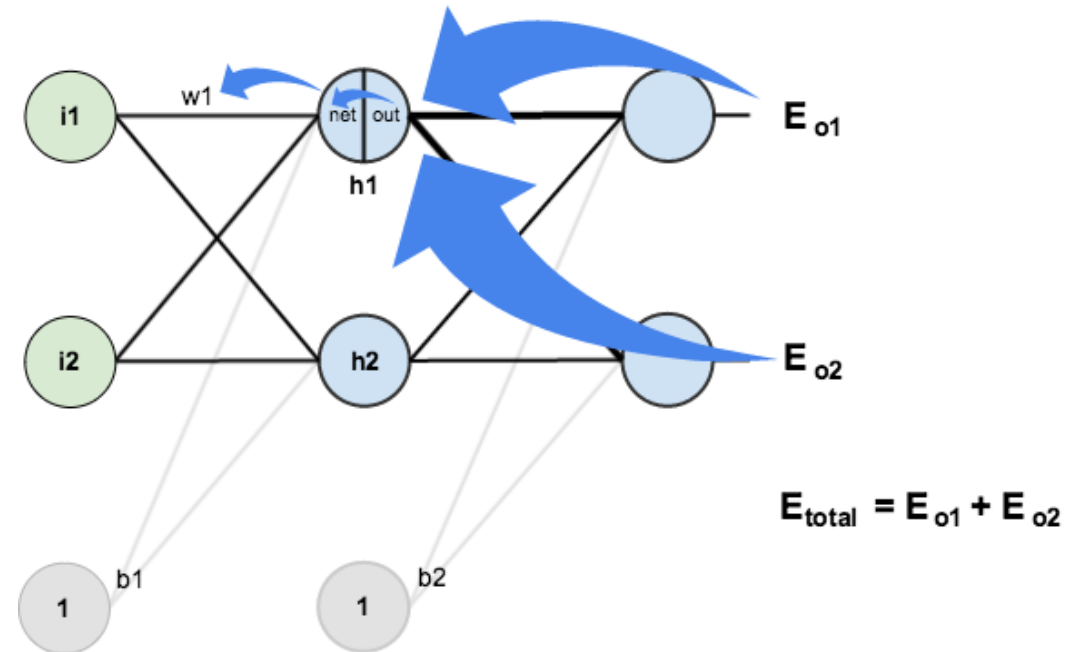
Невронни мрежи



$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



Невронни мрежи

A neural network can represent ANY given function.

Невронни мрежи

A neural network can represent ANY given function.

Теоретично невронна мрежа с един hidden layer може да научи всяка функция...

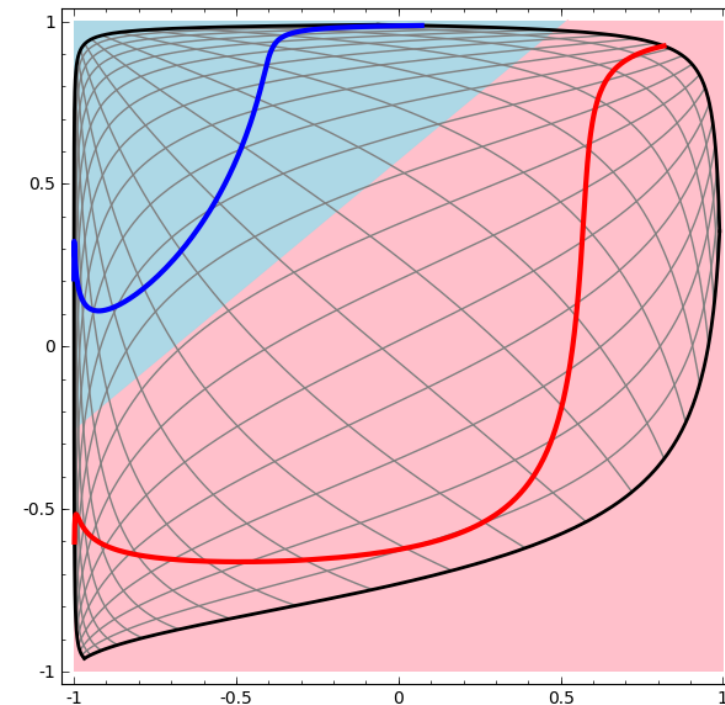
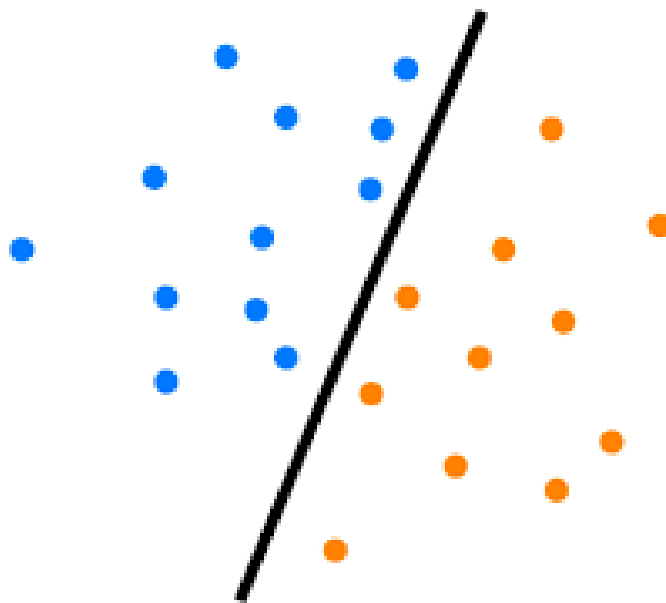
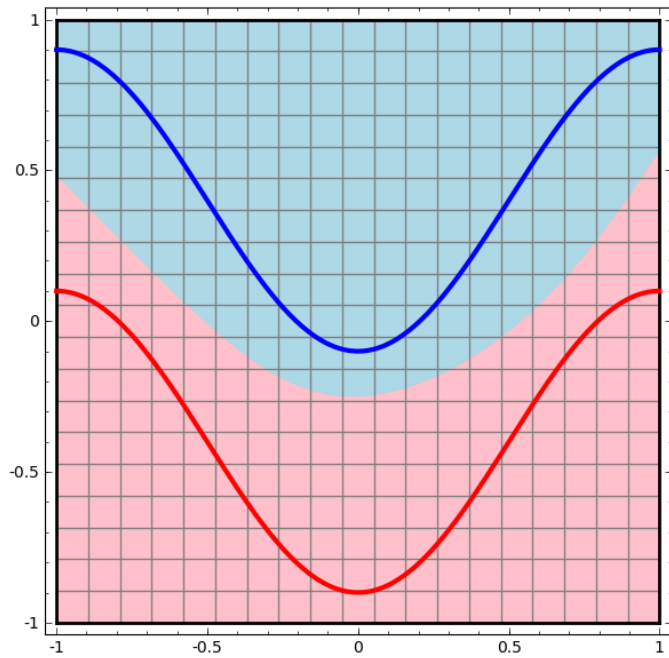
Невронни мрежи

A neural network can represent ANY given function.

Теоретично невронна мрежа с един hidden layer може да научи **всяка функция...**

...стига да имаш безкрайни данни и безброй неврони, които трябва да тренираш

Невронни мрежи



Невронни мрежи

Как да моделираме несигурност?

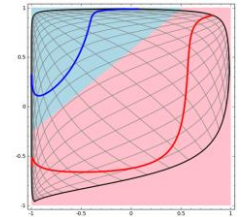
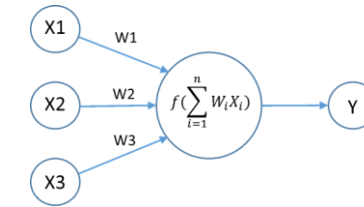
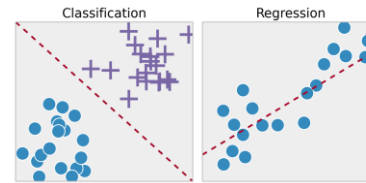
Как може да се възползваме от структурата на проблема, който искаме да решим?

Какво да правим, ако нямаме достатъчно данни за трениране на мрежата?

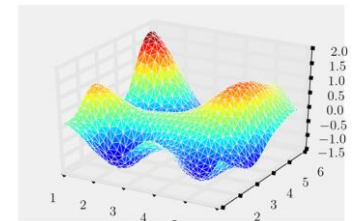
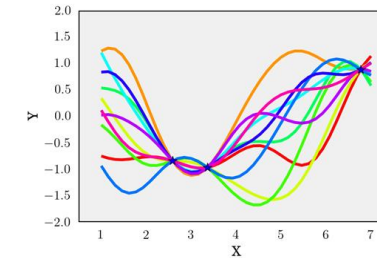
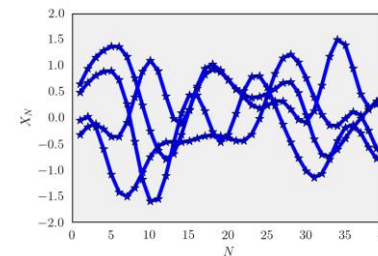
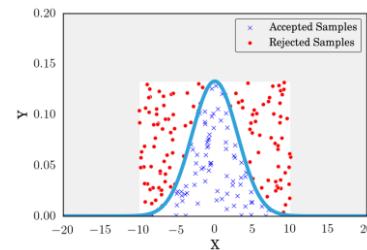
Има ли алтернативен модел, който може да реши тези проблеми?

План за лекцията

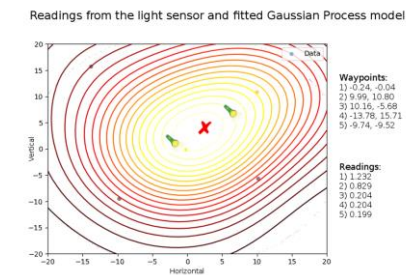
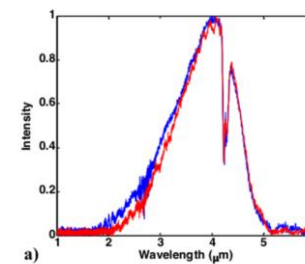
Машинно самообучение и невронни мрежи



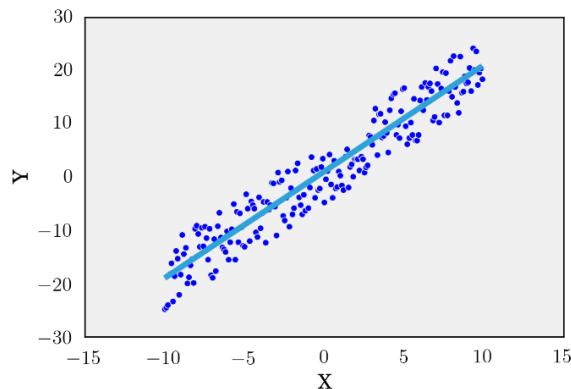
Гаусови процеси



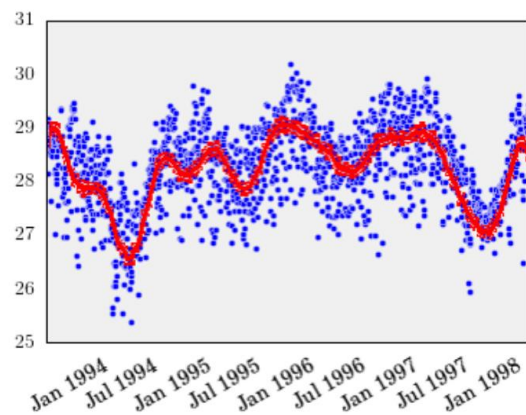
Активно семплиране с дрон използвайки Гаусови процеси за намиране на опасни газове



Нелинейна регресия

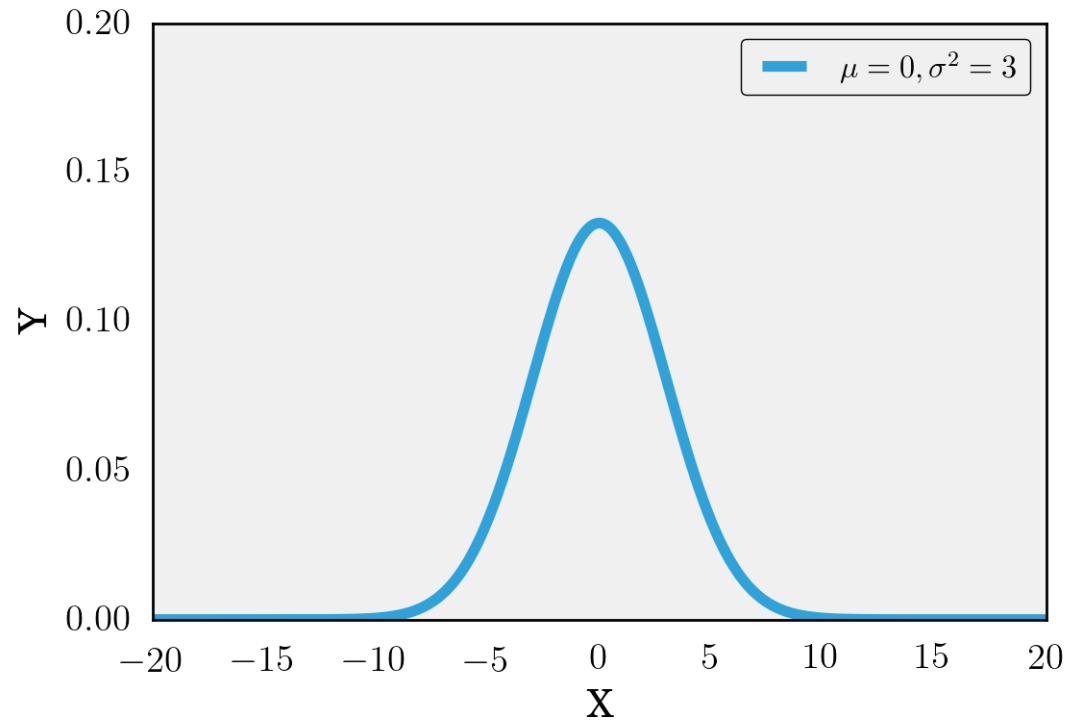


Как може да моделираме нелинейна регресия използвайки само Гаусова дистрибуция/разпределение?



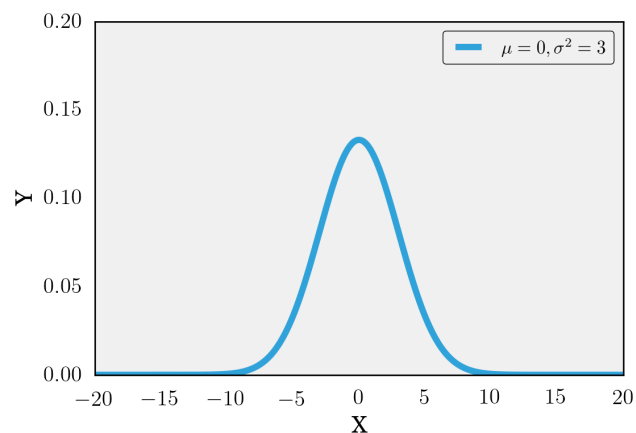
Има ли това някакви предимства и може ли да решим някои от проблемите, които невронните мрежи имат?

Гаусова дистрибуция/разпределение (Gaussian distribution)

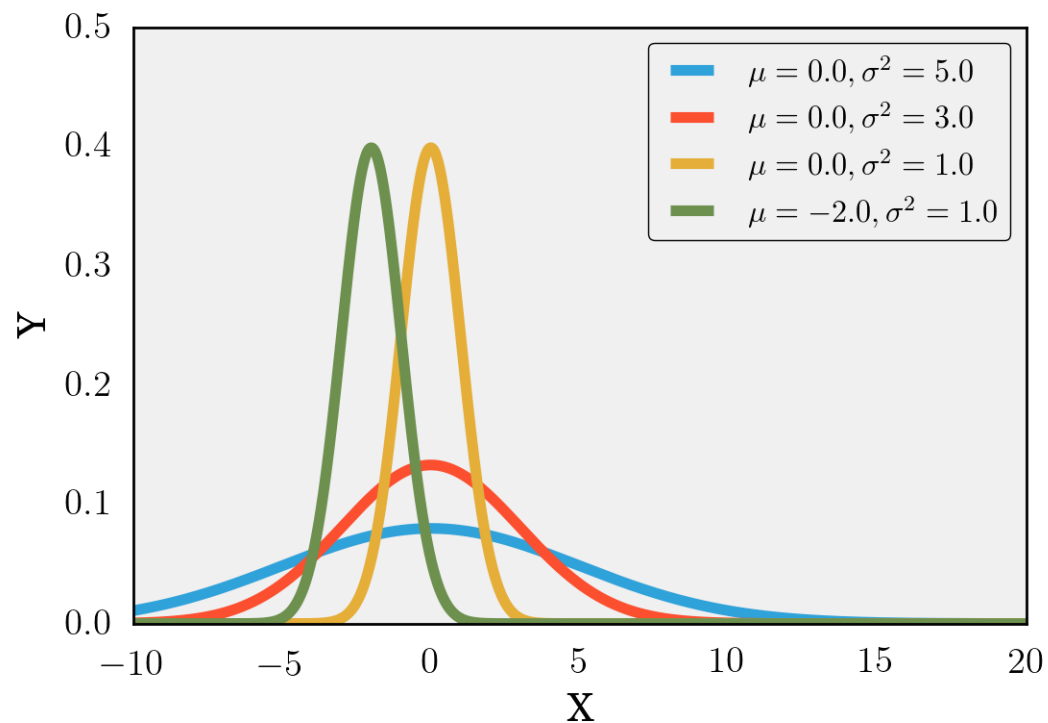


$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

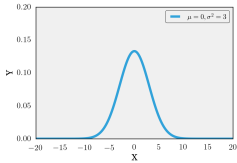
Гаусова дистрибуция



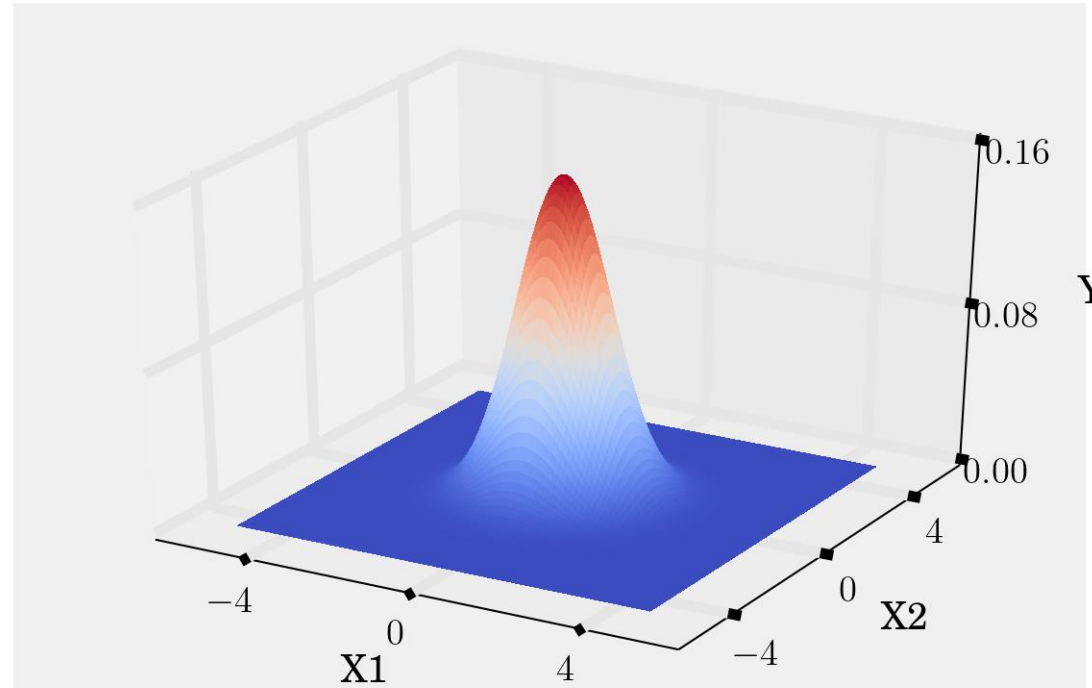
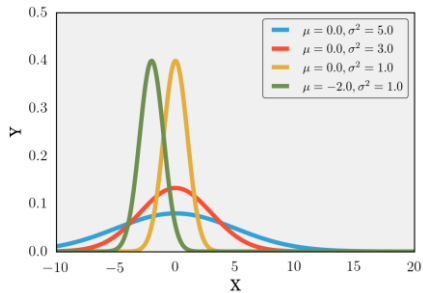
$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Многомерна Гаусова дистрибуция

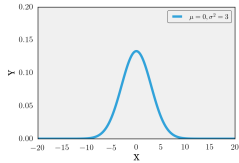


$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

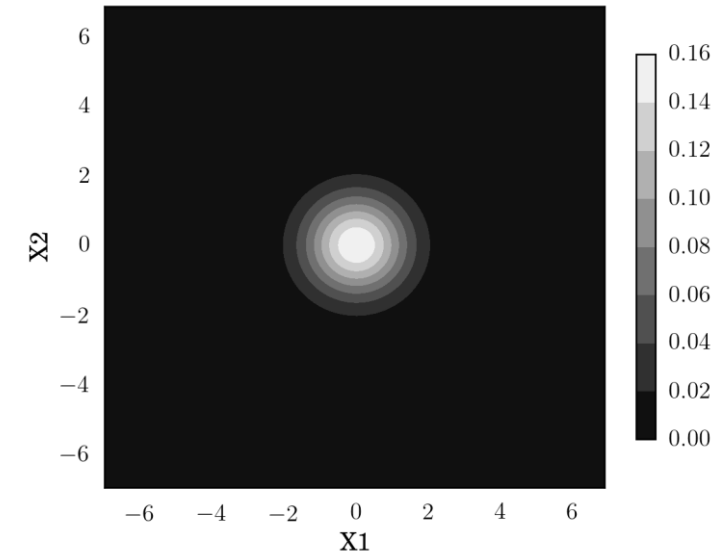
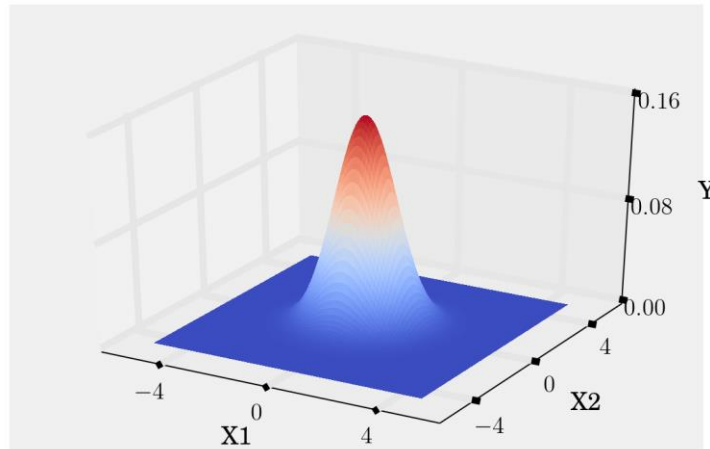
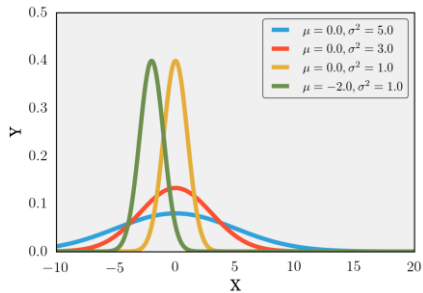


$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu)}$$

Многомерна Гаусова дистрибуция

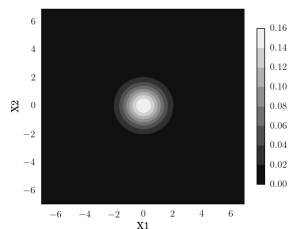
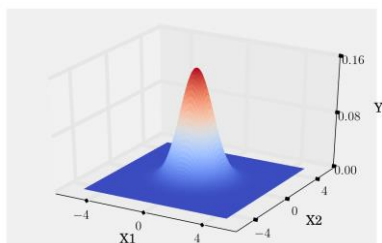


$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

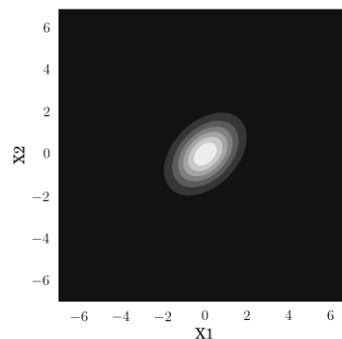


$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu)}$$

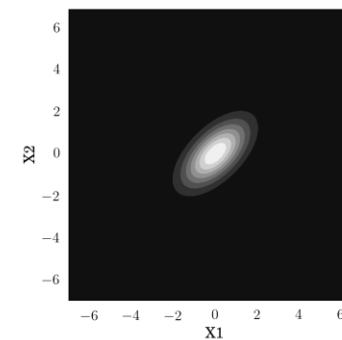
Многомерна Гаусова дистрибуция



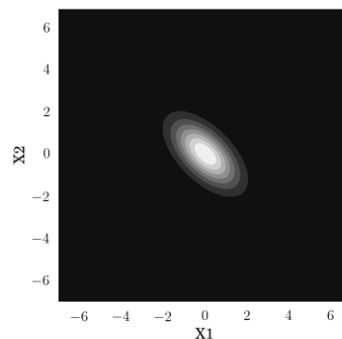
$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$



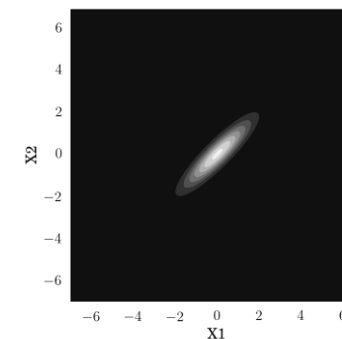
$$\Sigma = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

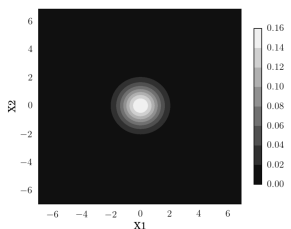
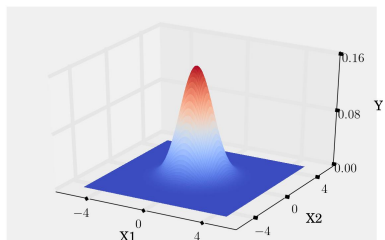


$$\Sigma = \begin{bmatrix} 1 & -0.6 \\ -0.6 & 1 \end{bmatrix}$$

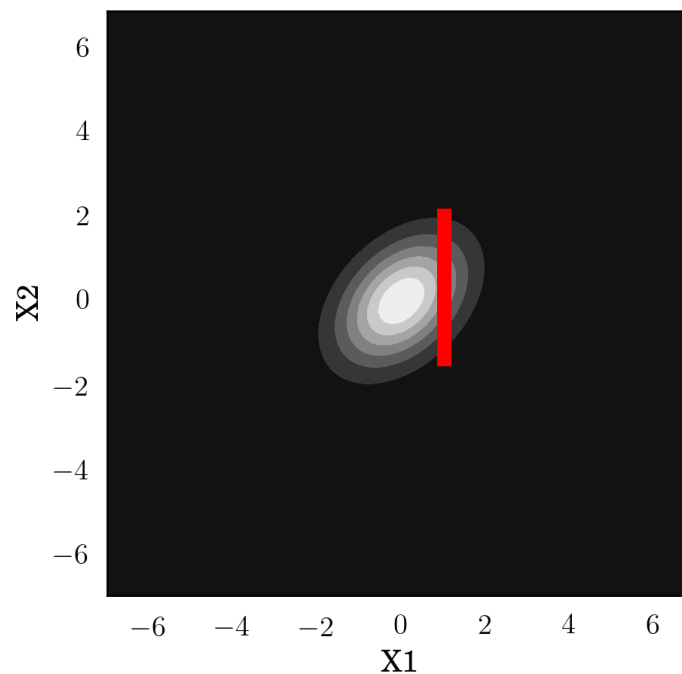


$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

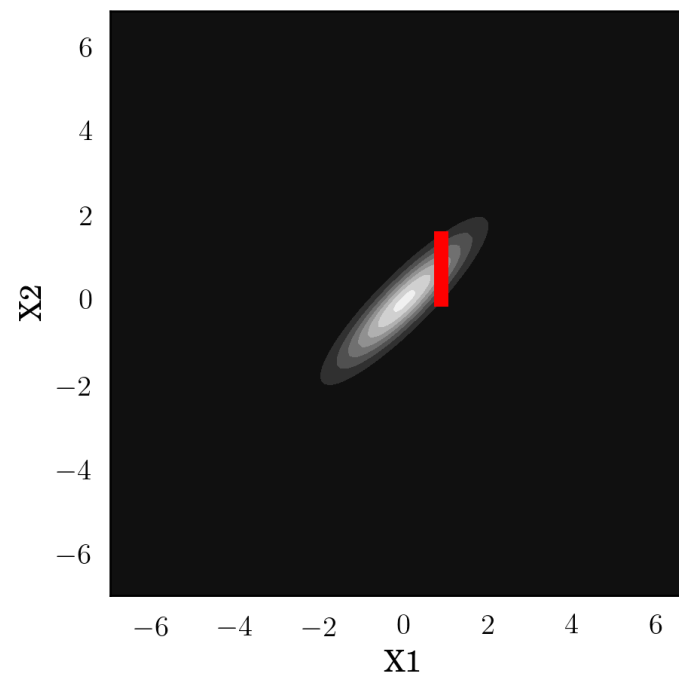
Многомерна Гаусова дистрибуция



$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

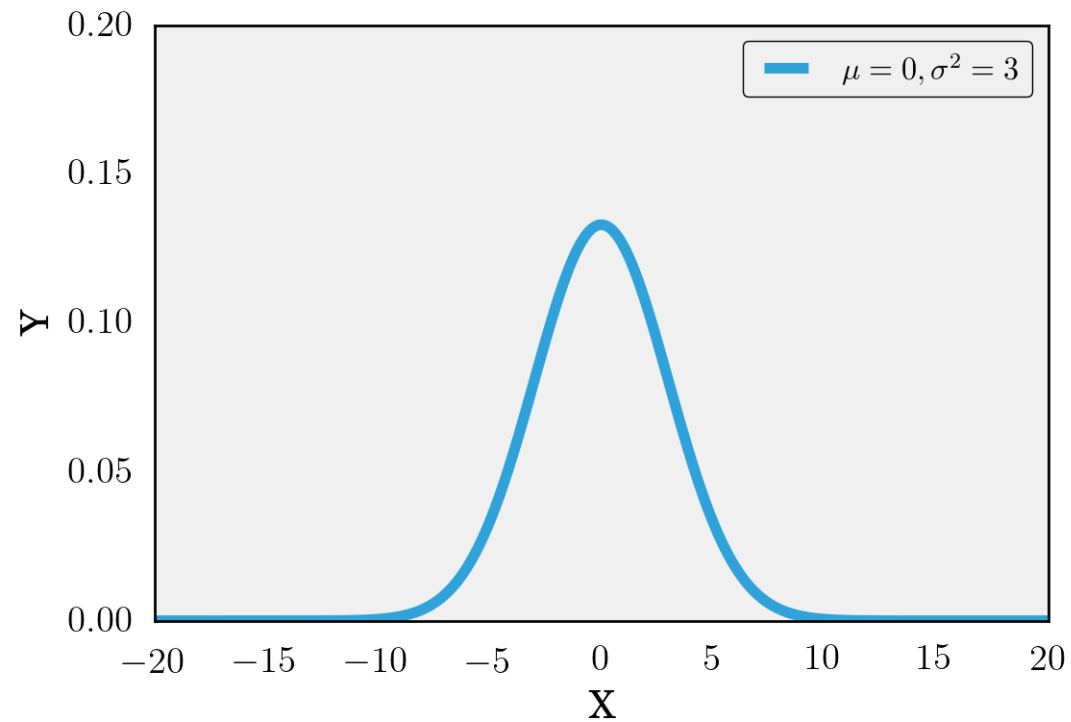


$$\Sigma = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$$

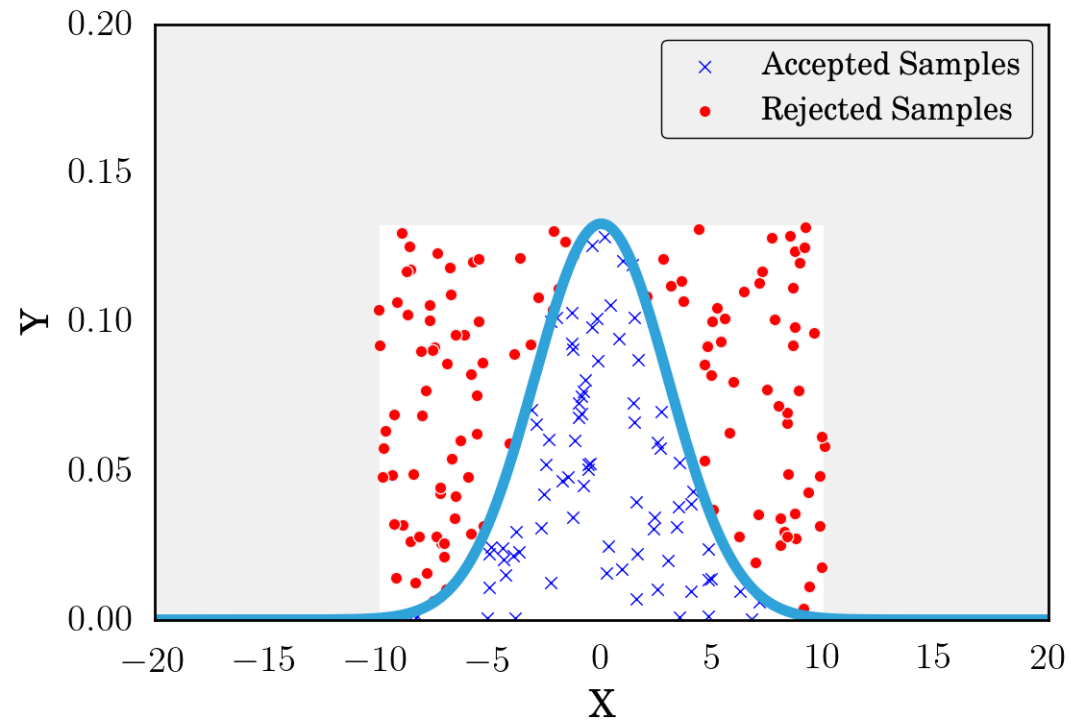


$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

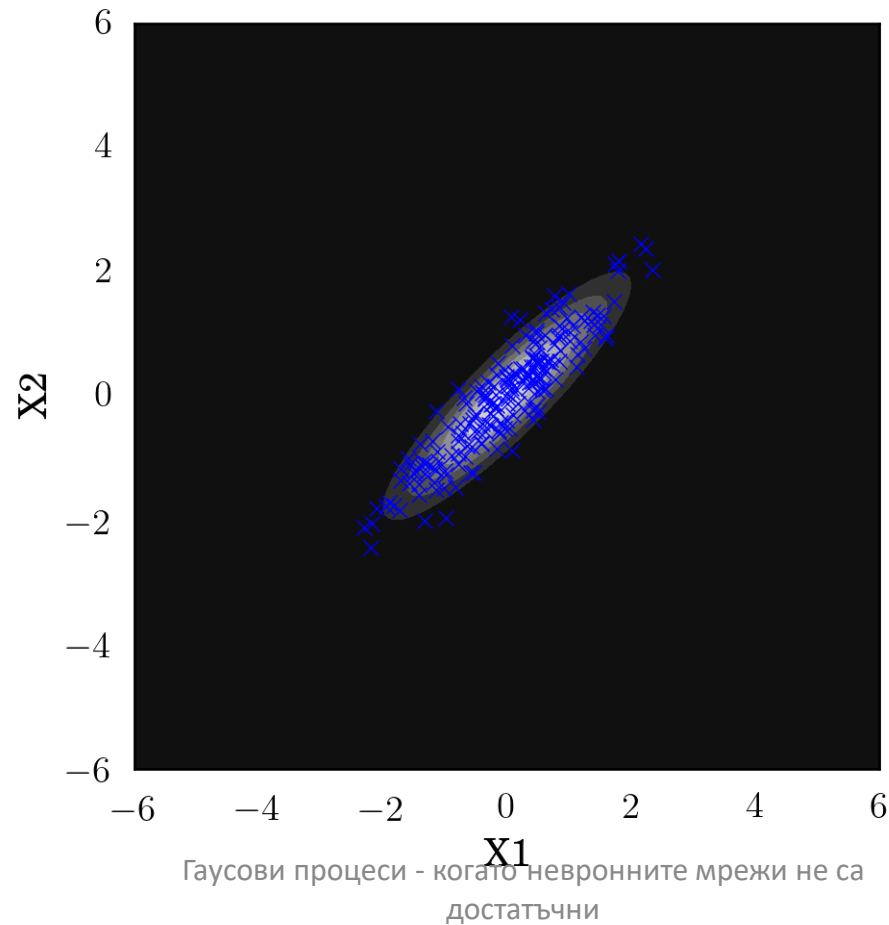
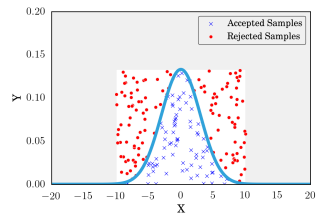
Семплиране (Sampling)



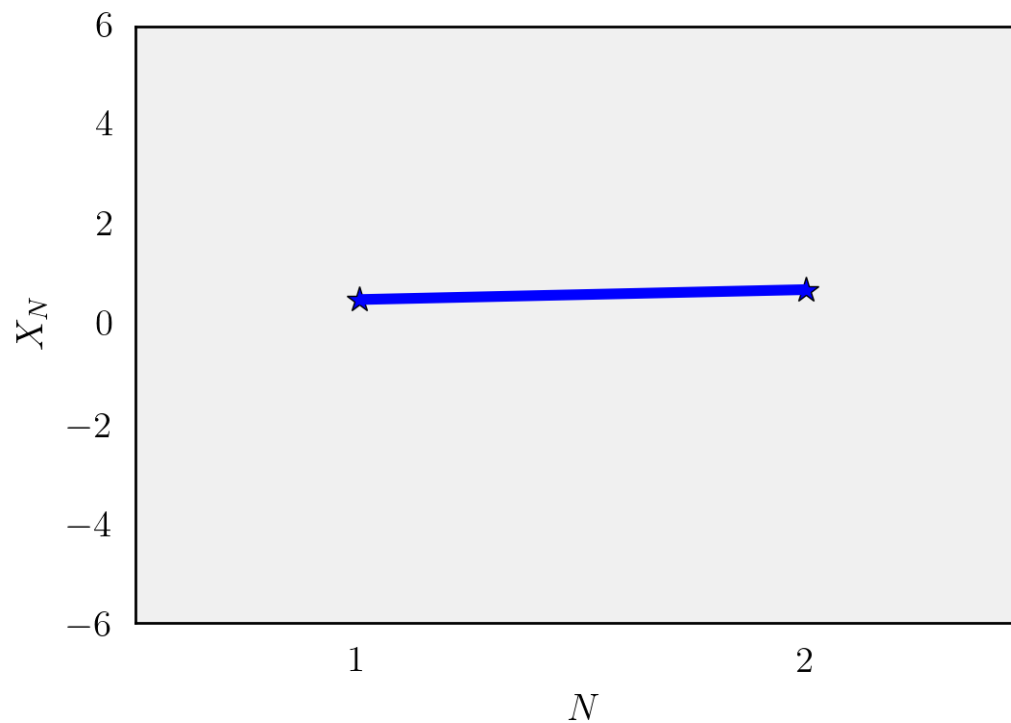
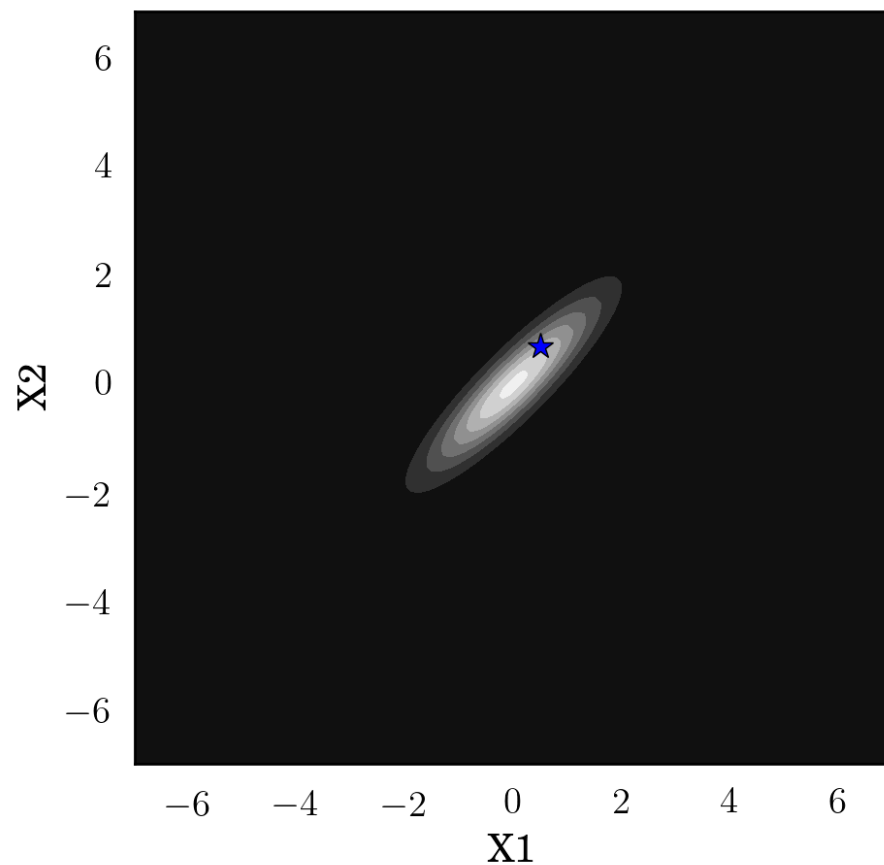
Семплиране (Rejection Sampling)



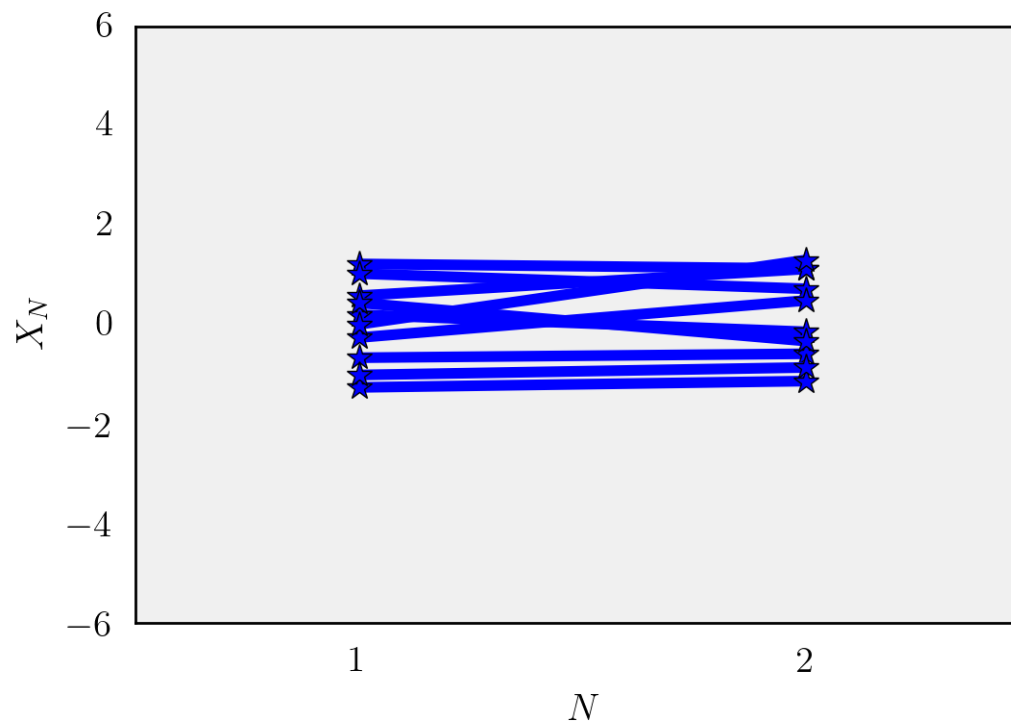
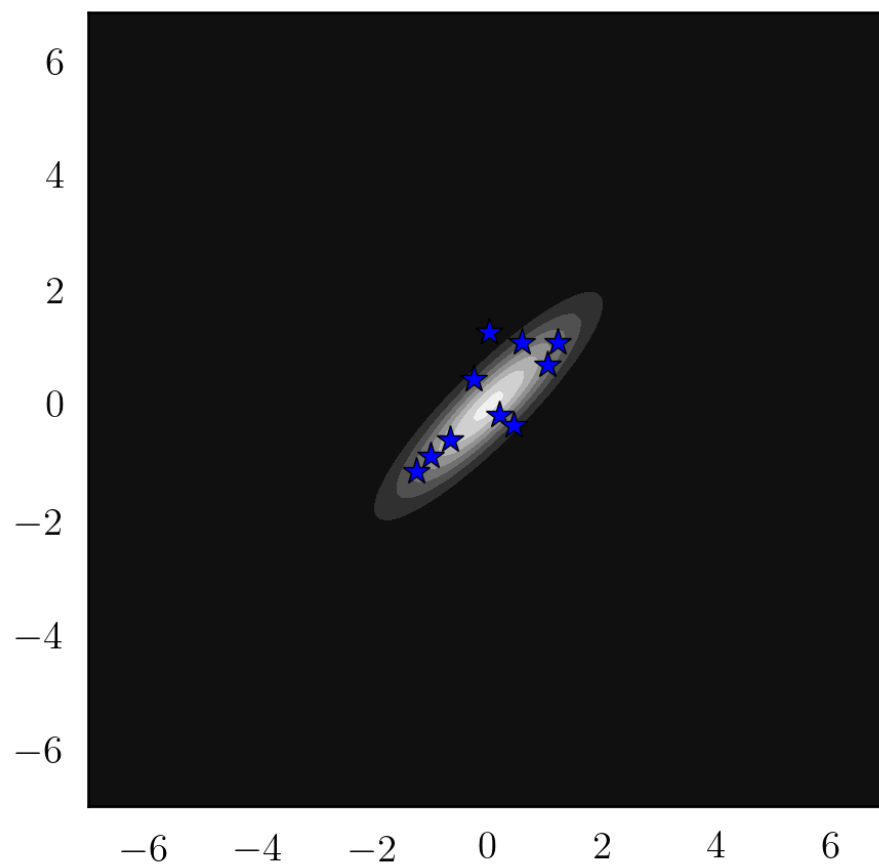
Семплиране на двумерна Гаусова дистрибуция



Алтернативна визуализация на семпълите



Алтернативна визуализация на семпълите



Алтернативна визуализация на семпълите

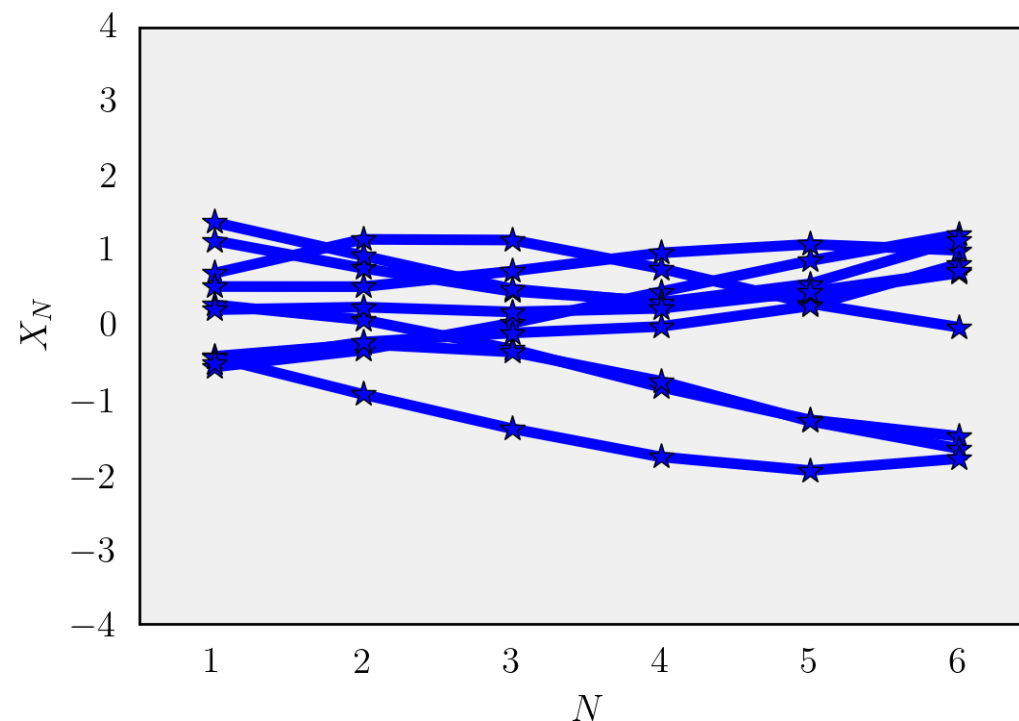
$$\mu = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\Sigma = \begin{bmatrix} 1 & 0.95 & 0.8 & 0.6 & 0.41 & 0.25 \\ 0.95 & 1 & 0.95 & 0.8 & 0.6 & 0.41 \\ 0.8 & 0.95 & 1 & 0.95 & 0.8 & 0.6 \\ 0.6 & 0.8 & 0.95 & 1 & 0.95 & 0.8 \\ 0.41 & 0.6 & 0.8 & 0.95 & 1 & 0.95 \\ 0.25 & 0.41 & 0.6 & 0.8 & 0.95 & 1 \end{bmatrix}$$

Алтернативна визуализация на семпълите

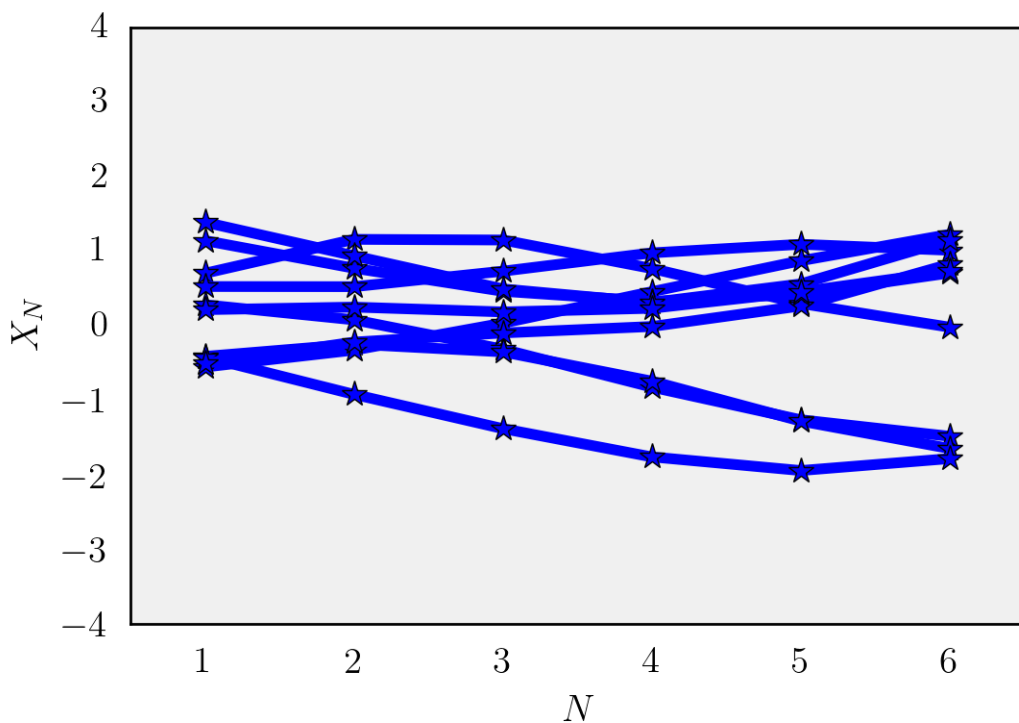
$$\mu = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$\Sigma = \begin{bmatrix} 1 & 0.95 & 0.8 & 0.6 & 0.41 & 0.25 \\ 0.95 & 1 & 0.95 & 0.8 & 0.6 & 0.41 \\ 0.8 & 0.95 & 1 & 0.95 & 0.8 & 0.6 \\ 0.6 & 0.8 & 0.95 & 1 & 0.95 & 0.8 \\ 0.41 & 0.6 & 0.8 & 0.95 & 1 & 0.95 \\ 0.25 & 0.41 & 0.6 & 0.8 & 0.95 & 1 \end{bmatrix}$$



Each line is one sample from a 6D Gaussian

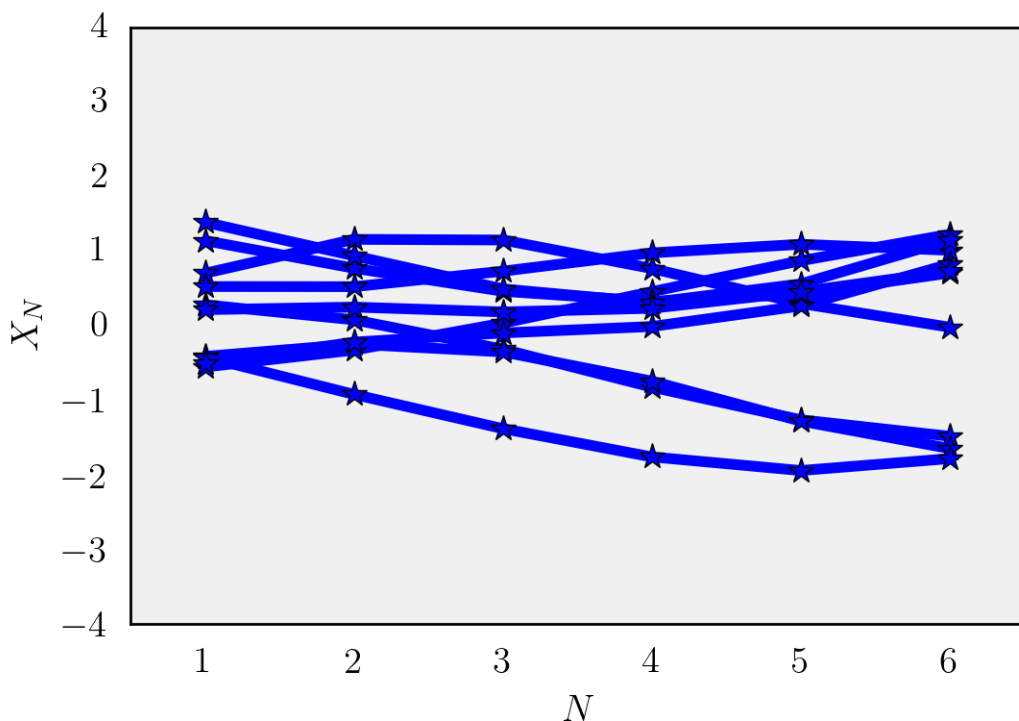
Алтернативна визуализация на семпълите



- Линиите започват да приличат на нелинейна регресия

Each line is one sample from a 6D Gaussian

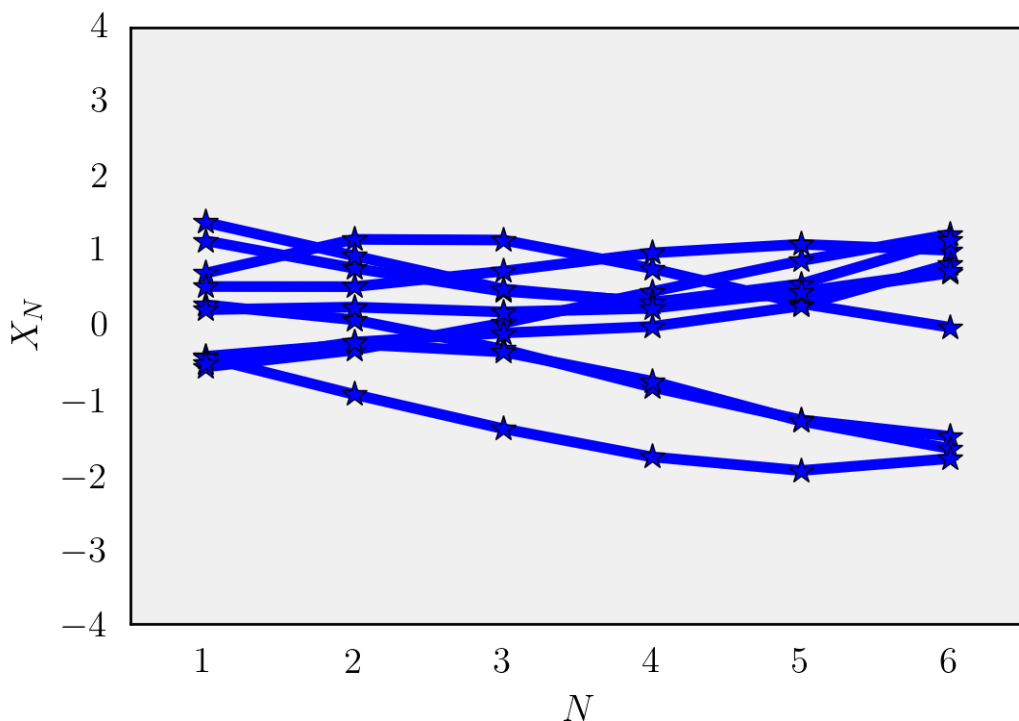
Алтернативна визуализация на семпълите



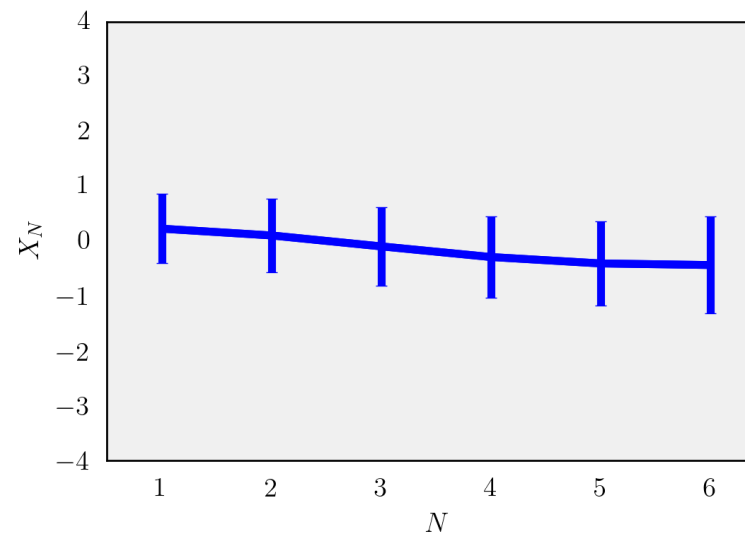
- Линиите започват да приличат на нелинейна регресия
- Точки близо една до друга имат висока корелация

Each line is one sample from a 6D Gaussian

Алтернативна визуализация на семпълите

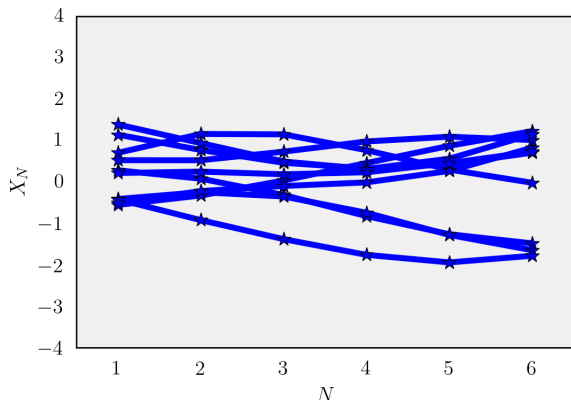


- Линиите започват да приличат на нелинейна регресия
- Точки близо една до друга имат висока корелация
- Може да измерим вариацията във всяка точка



Each line is one sample from a 6D Gaussian

Генериране на covariance матрицата



Кернел функцията определя всяко число в covariance матрицата.

‘Squared exponential’ kernel

$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

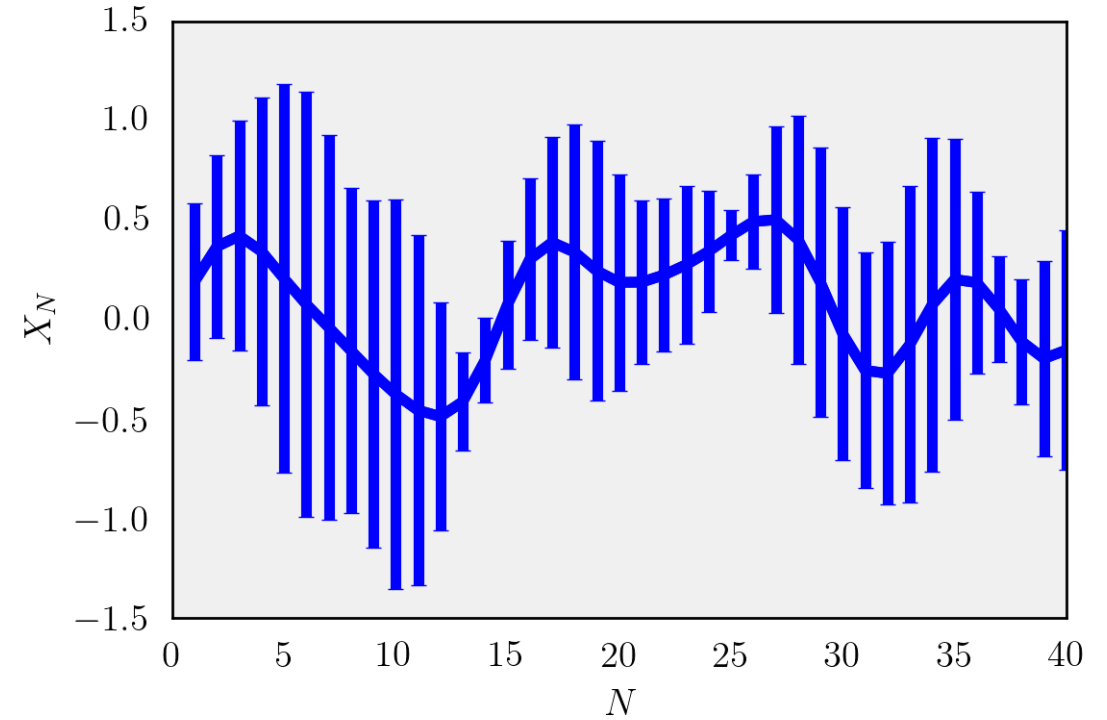
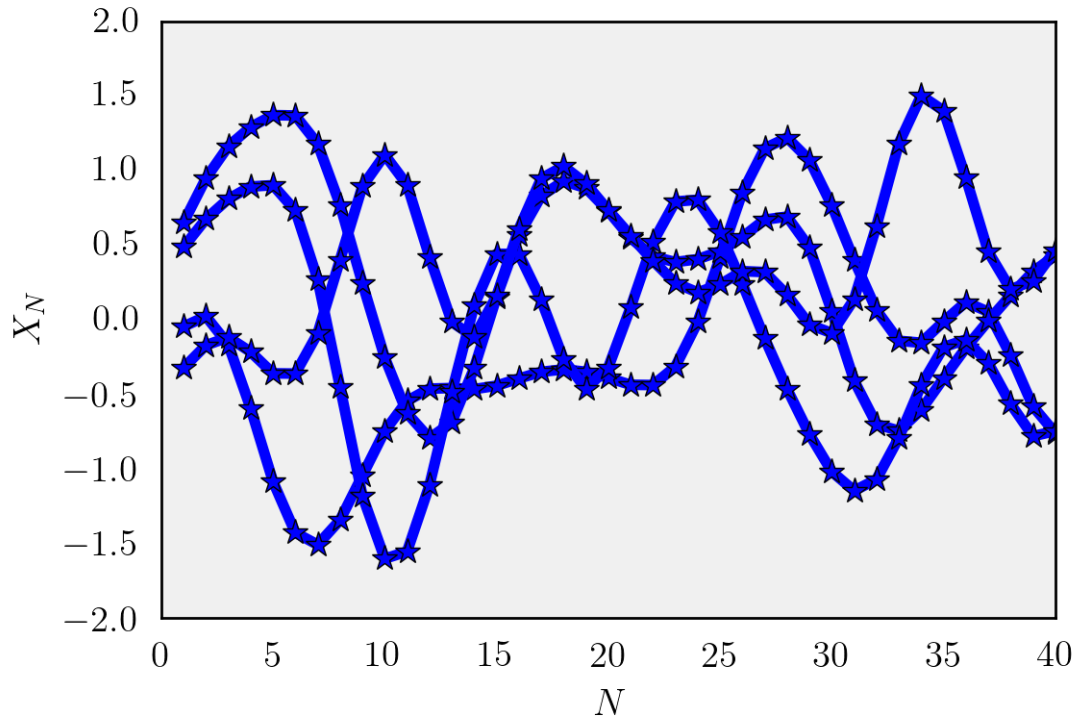
Гаусовите процеси са **напълно определени от mean и covariance функции**.

Гаусовите процеси са **дистрибуция от функции с определени свойства**.

$$f \sim GP(m, k)$$

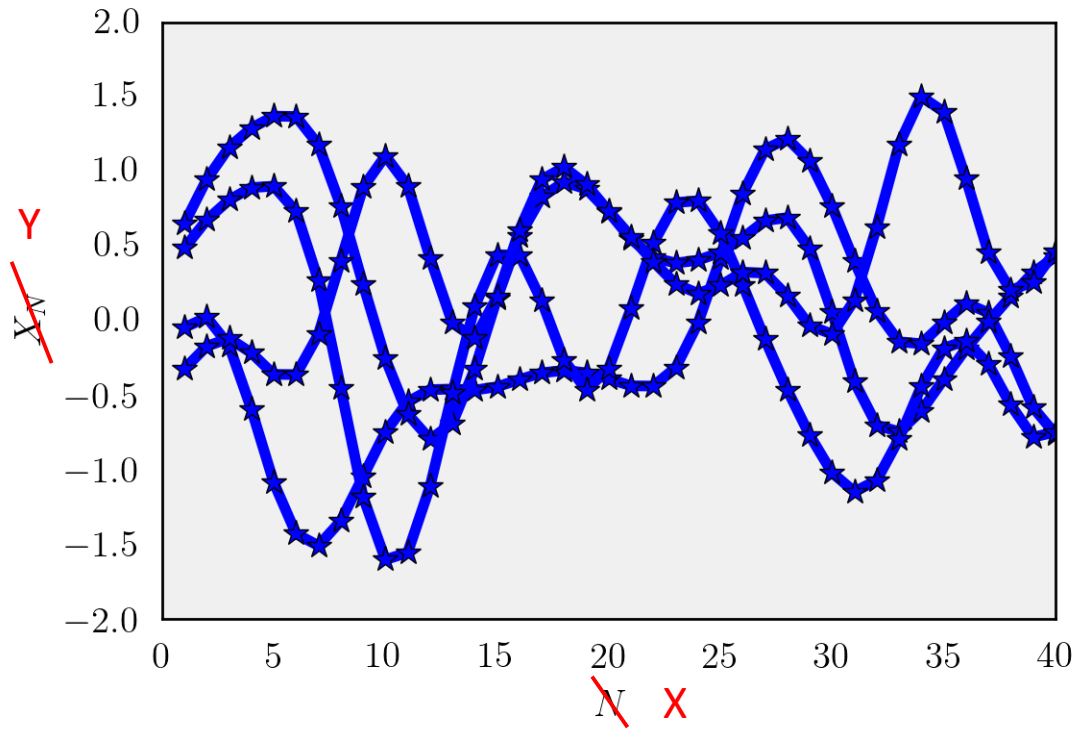
$$\Sigma = \begin{bmatrix} 1 & 0.95 & 0.8 & 0.6 & 0.41 & 0.25 \\ 0.95 & 1 & 0.95 & 0.8 & 0.6 & 0.41 \\ 0.8 & 0.95 & 1 & 0.95 & 0.8 & 0.6 \\ 0.6 & 0.8 & 0.95 & 1 & 0.95 & 0.8 \\ 0.41 & 0.6 & 0.8 & 0.95 & 1 & 0.95 \\ 0.25 & 0.41 & 0.6 & 0.8 & 0.95 & 1 \end{bmatrix}$$

Използване на многомерна Гаусова дистрибуция



Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

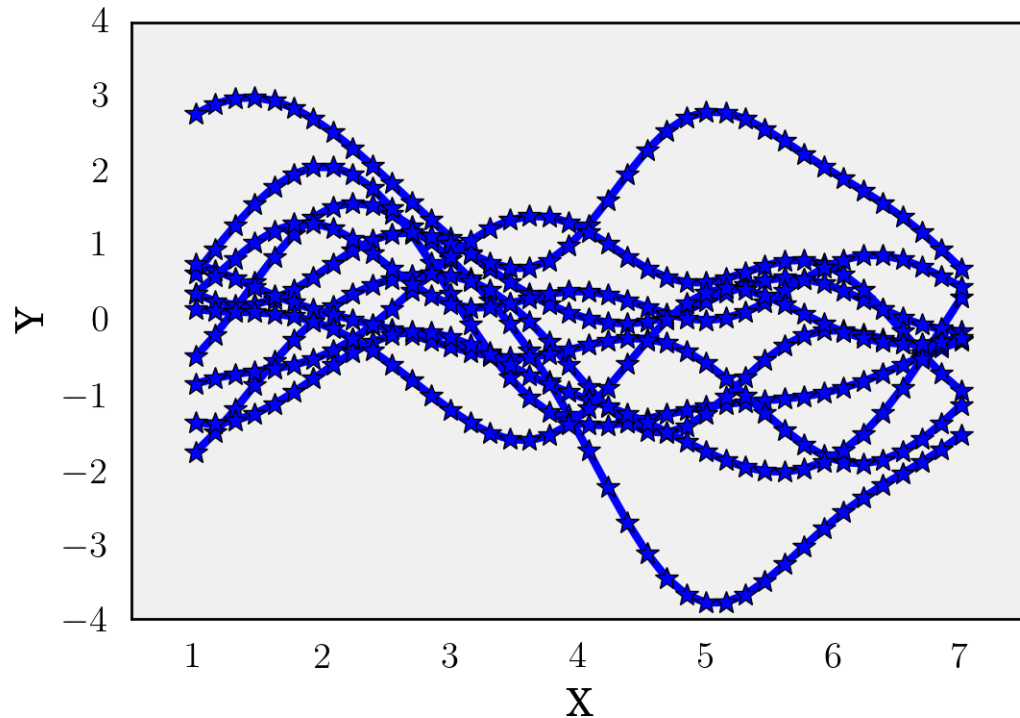
Още една промяна в нотацията



- Преименуване на осите

Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

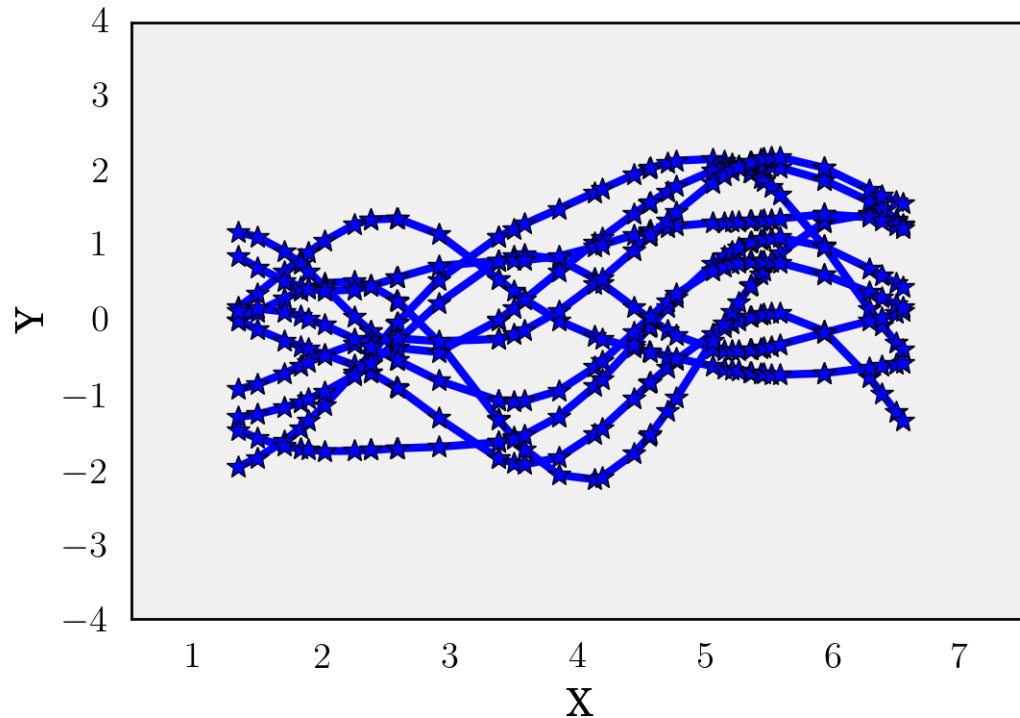
Още една промяна в нотацията



- Преименуване на осите
- Не е задължително да увеличаваме X-оста с измерението на Гаусовата дистрибуция

Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

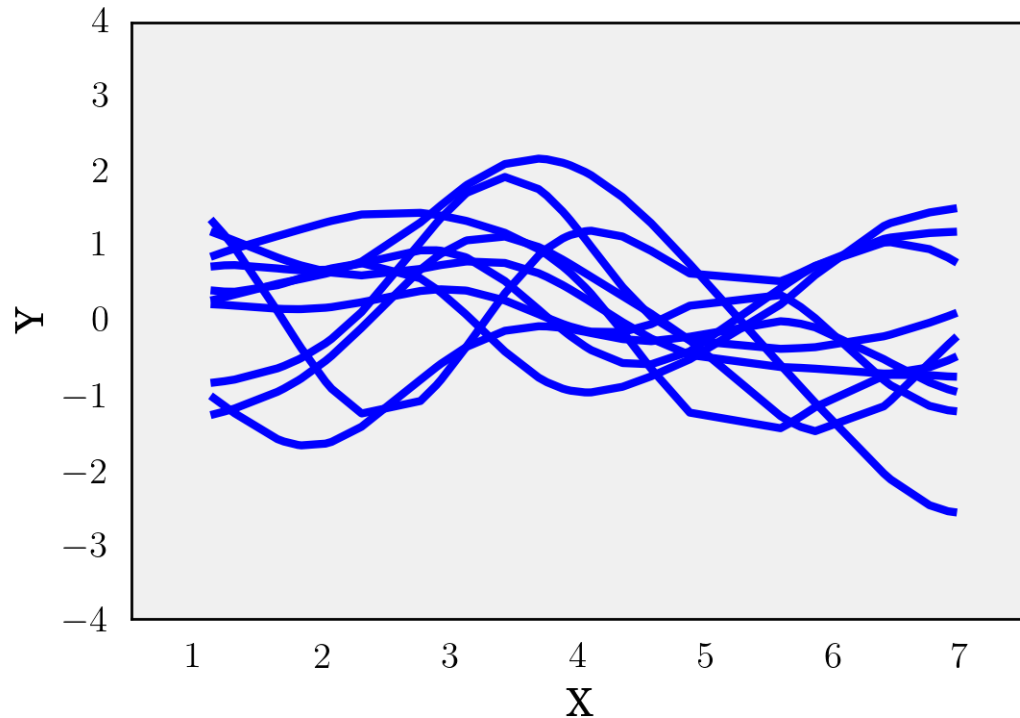
Още една промяна в нотацията



- Преименуване на осите
- Не е задължително да увеличаваме X-оста с измерението на Гаусовата дистрибуция
- Не е задължително да взимаме равно разделчени точки

Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

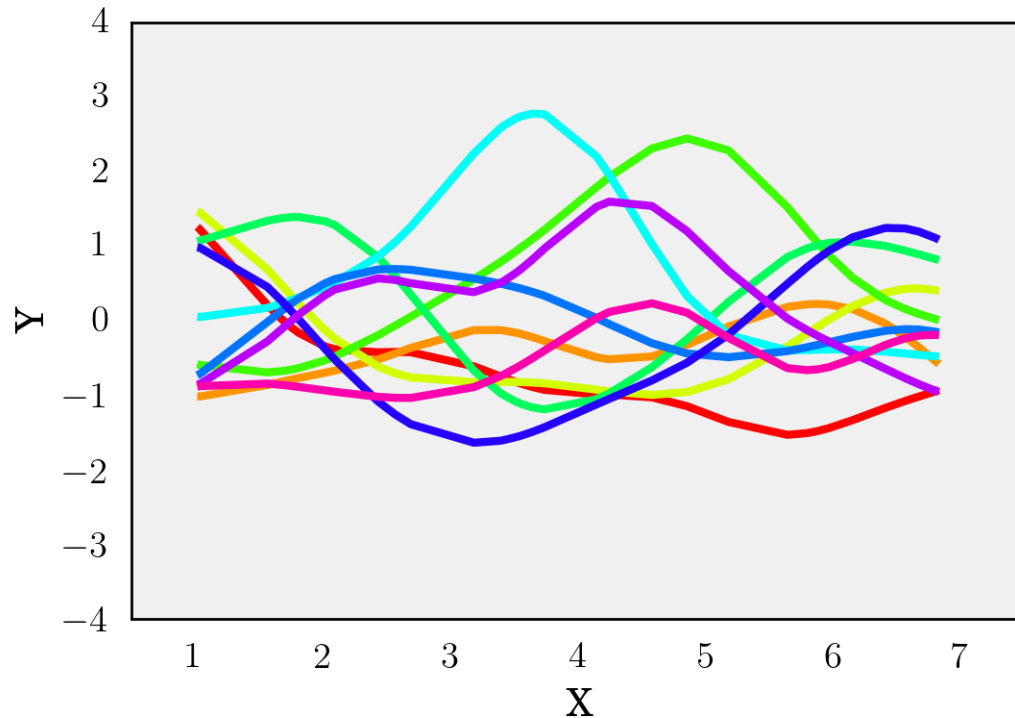
Още една промяна в нотацията



- Преименуване на осите
- Не е задължително да увеличаваме X-оста с измерението на Гаусовата дистрибуция
- Не е задължително да взимаме равно разделчени точки
- (Може да махнем точките за по-ясна графика)

Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

Още една промяна в нотацията



- Преименуване на осите
- Не е задължително да увеличаваме X-оста с измерението на Гаусовата дистрибуция
- Не е задължително да вземаме равно разделчени точки
- (Може да махнем точките за по-ясна графика)
- (Може да използваме различни цветове, отново за яснота)

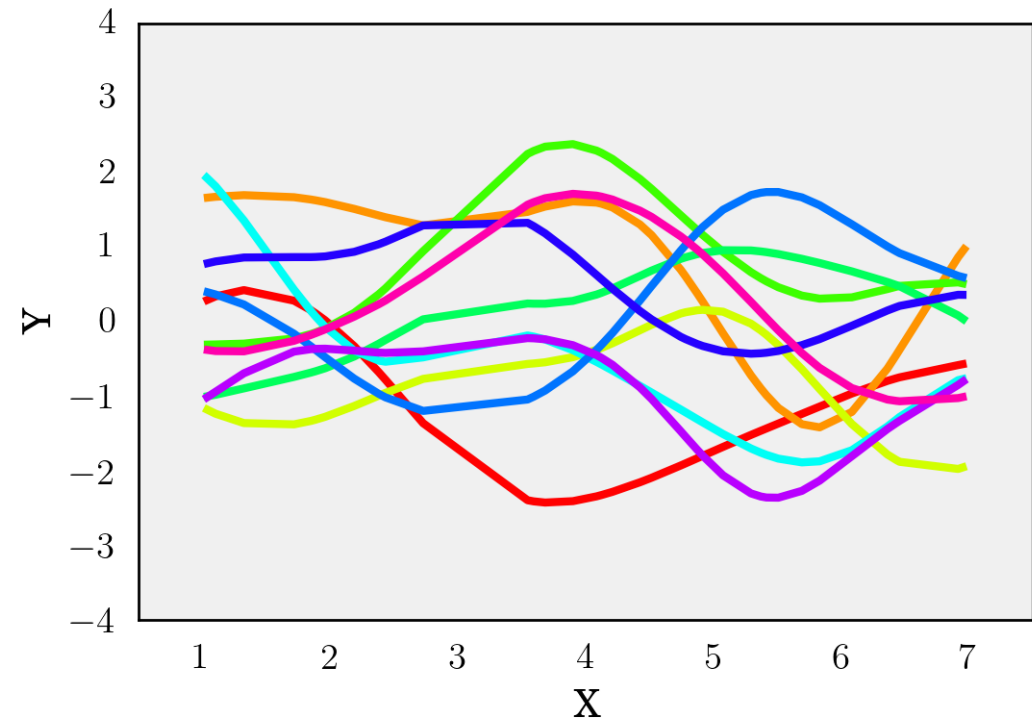
Всяка функция е смпъл от 40-мерна Гаусова дистрибуция

Вариране хиперпараметрите на кернела

$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v	Noise level	0
l	Horizontal lengthscale	1
σ_f	Vertical lengthscale	1



Вариране хиперпараметрите на кернела

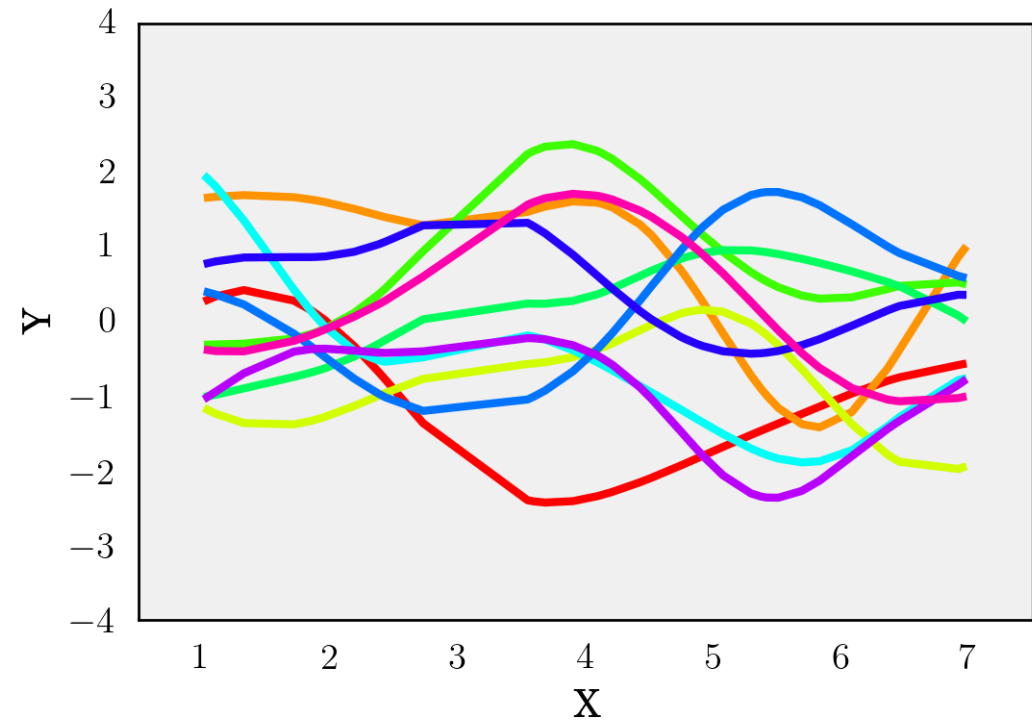
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

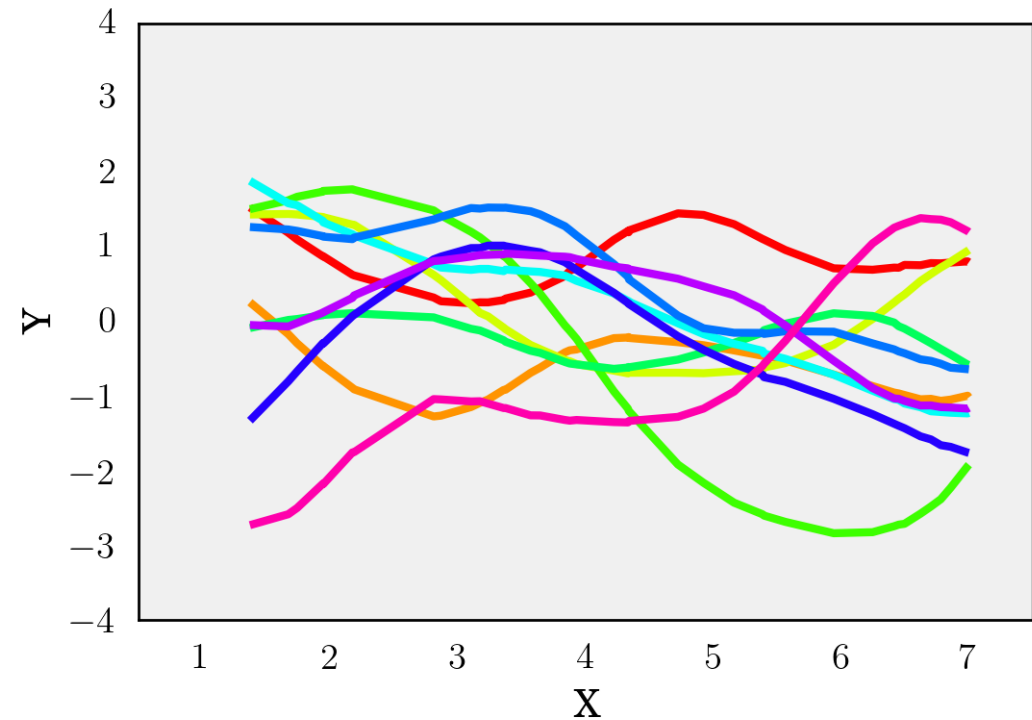
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0.01

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

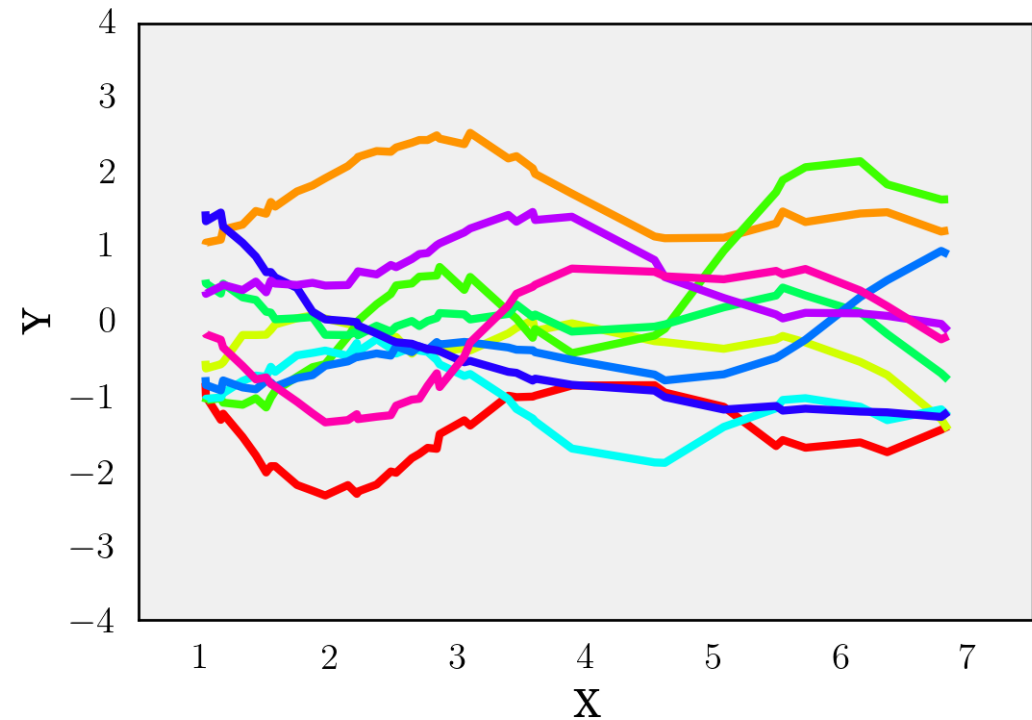
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0.05

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

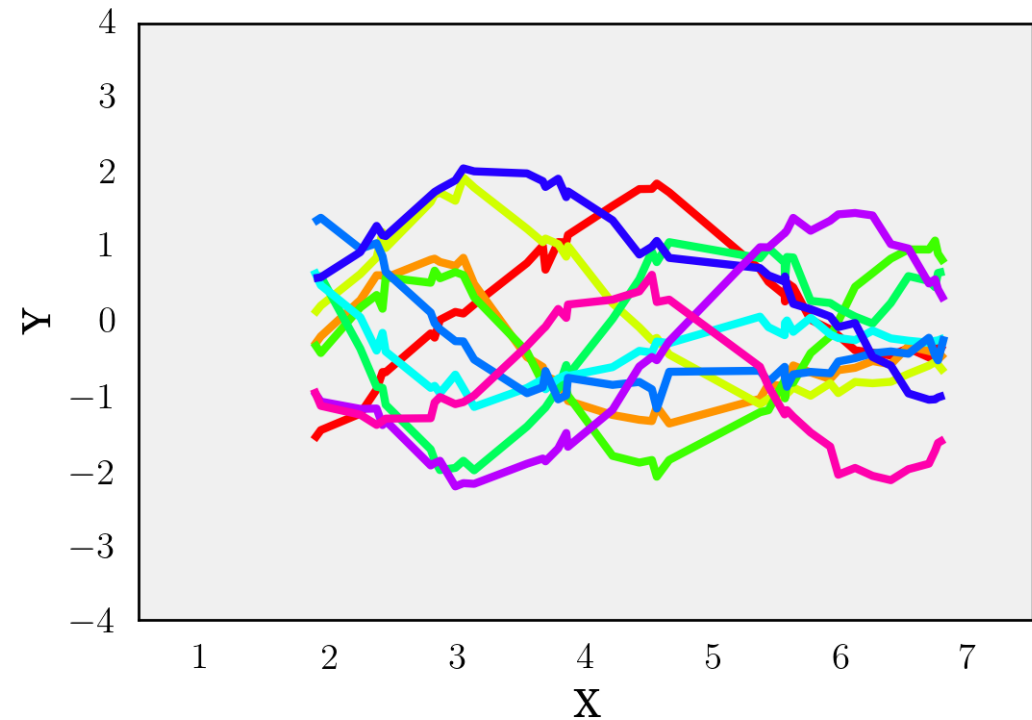
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0.1

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

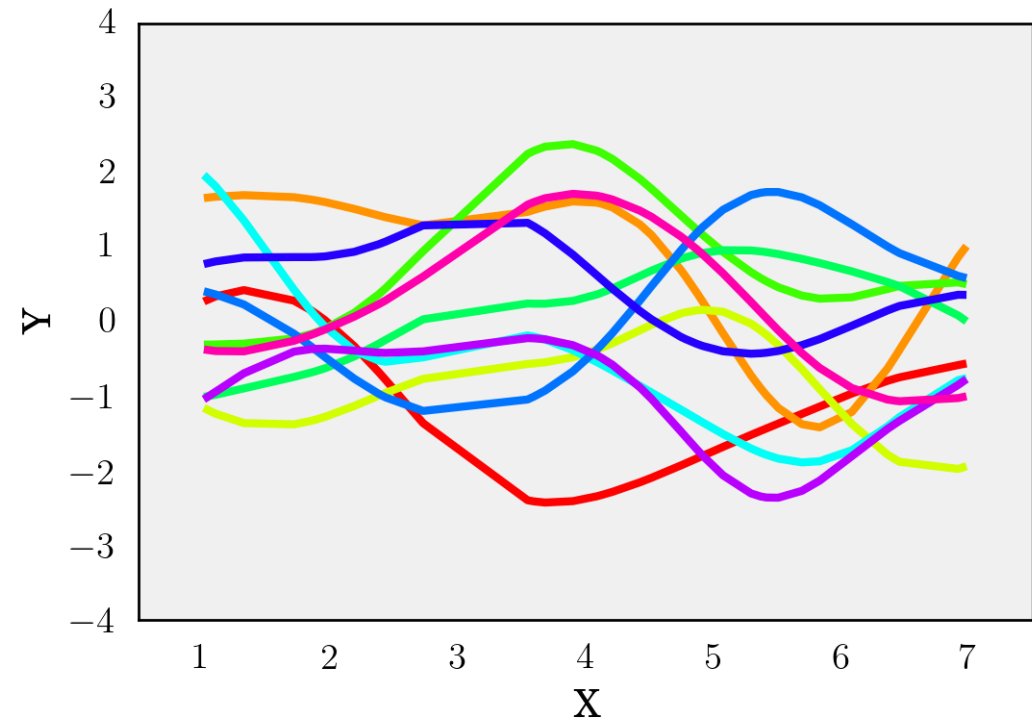
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1

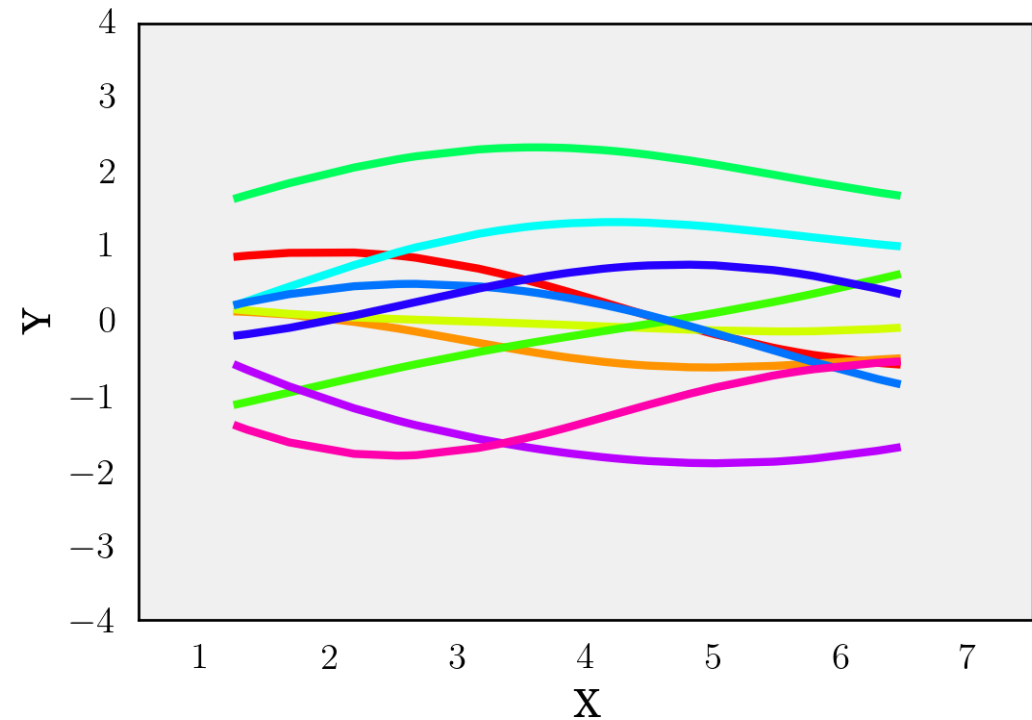


Вариране хиперпараметрите на кернела

$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v	Noise level	0
l	Horizontal lengthscale	3
σ_f	Vertical lengthscale	1

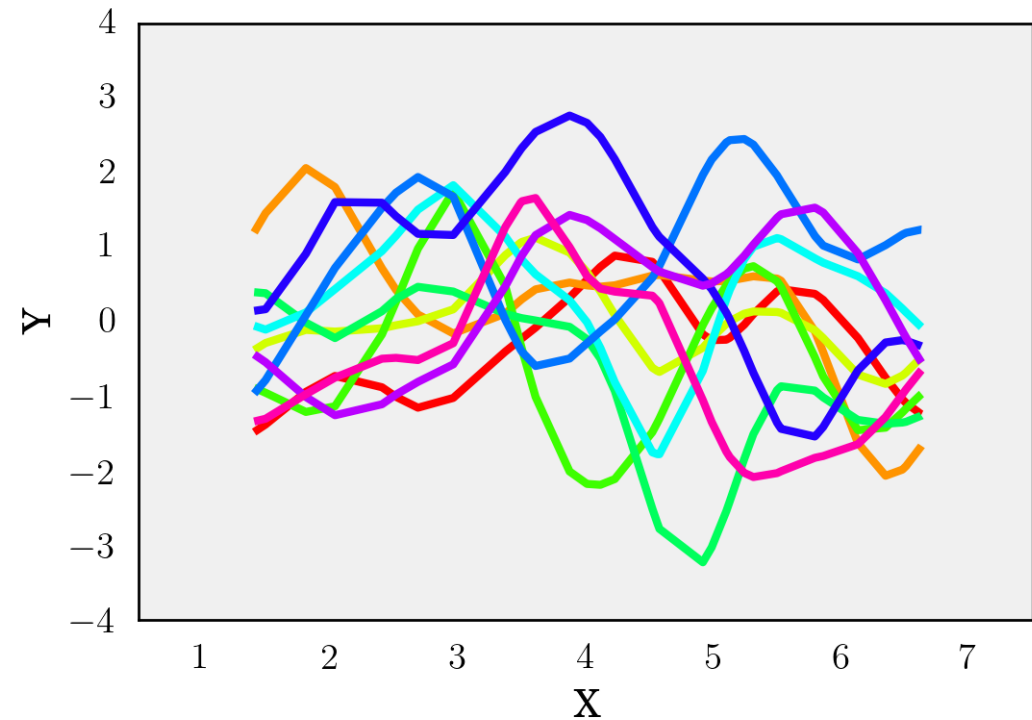


Вариране хиперпараметрите на кернела

$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v	Noise level	0
l	Horizontal lengthscale	0.5
σ_f	Vertical lengthscale	1



Вариране хиперпараметрите на кернела

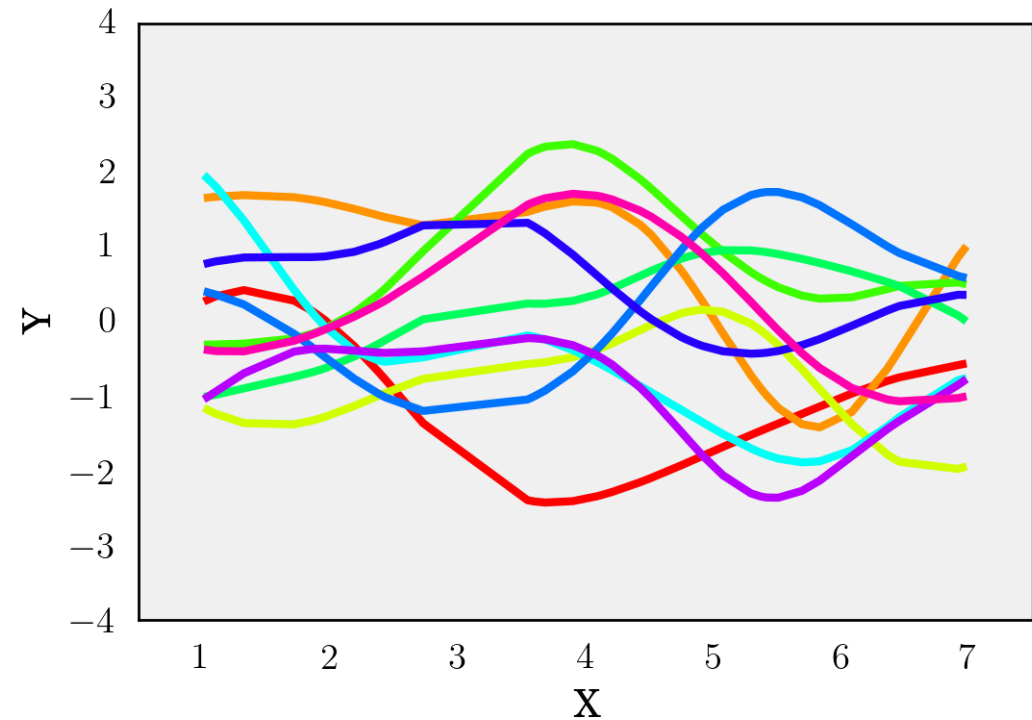
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

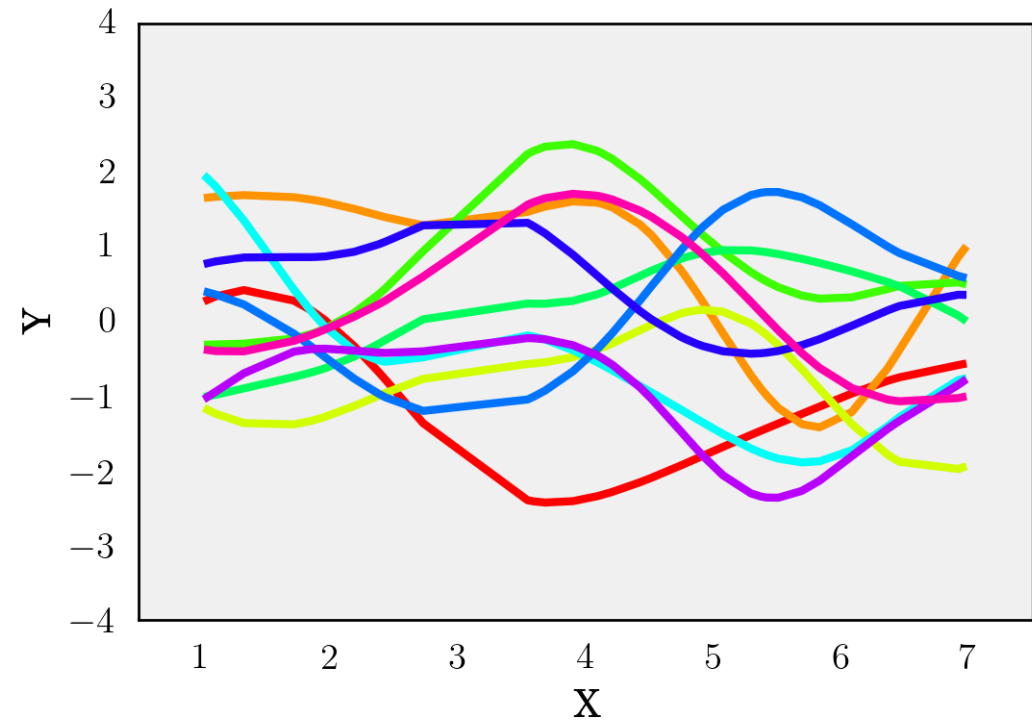
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0

l Horizontal lengthscale 1

σ_f Vertical lengthscale 1



Вариране хиперпараметрите на кернела

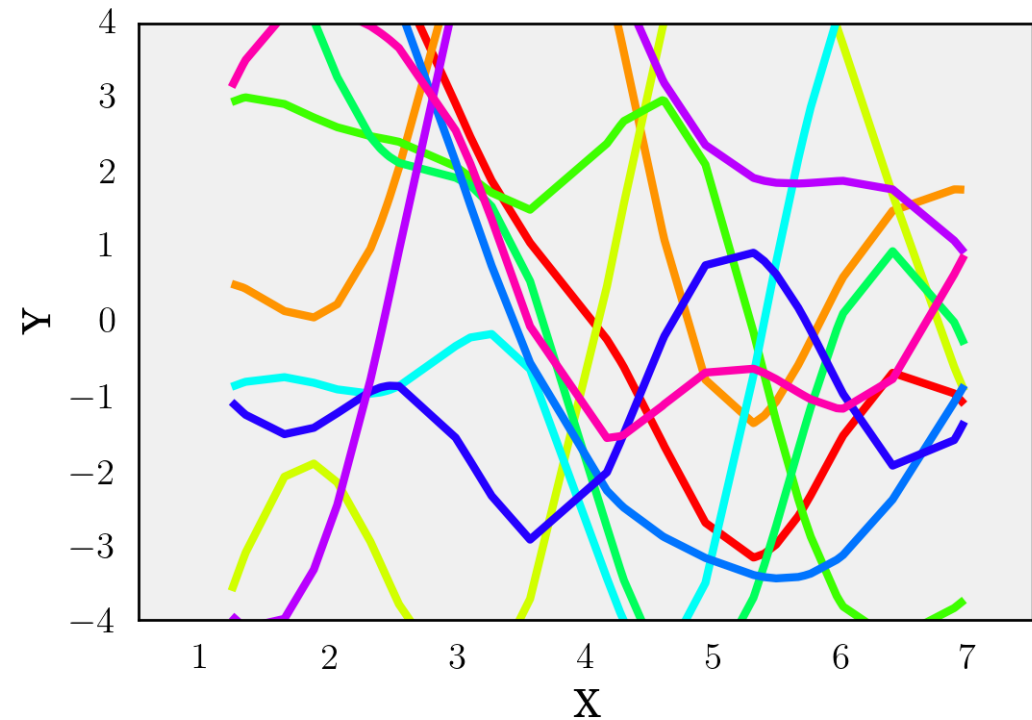
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

σ_v Noise level 0

l Horizontal lengthscale 1

σ_f Vertical lengthscale 3

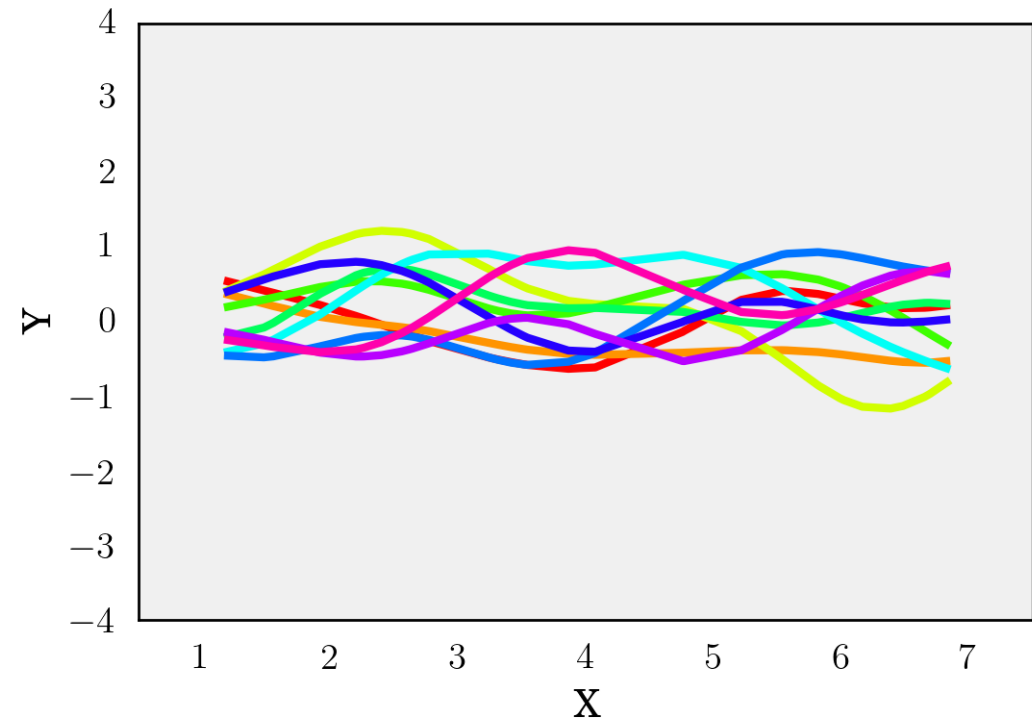


Вариране хиперпараметрите на кернела

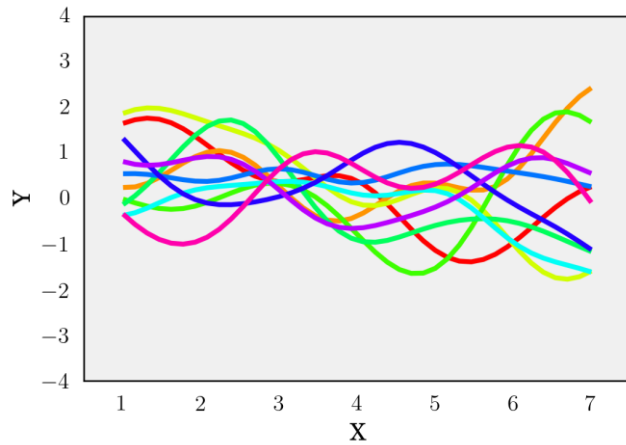
$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$

$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x_n - x_{n'})^2\right)$$

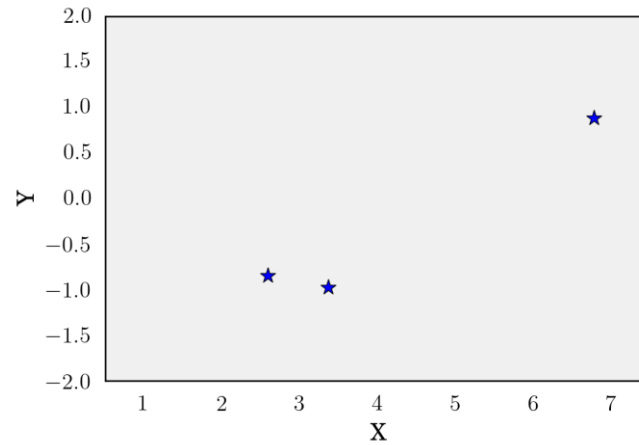
σ_v	Noise level	0
l	Horizontal lengthscale	1
σ_f	Vertical lengthscale	0.5



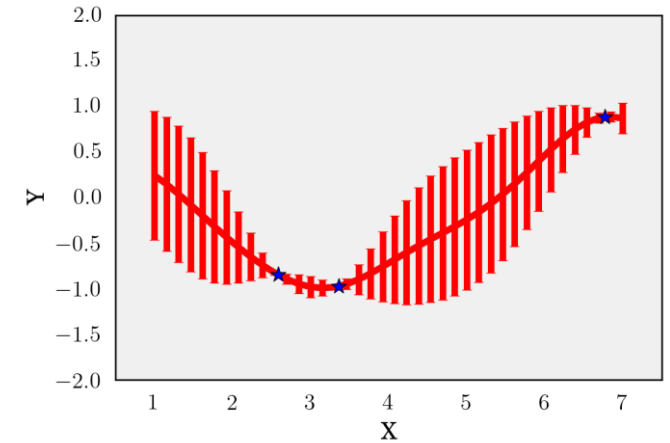
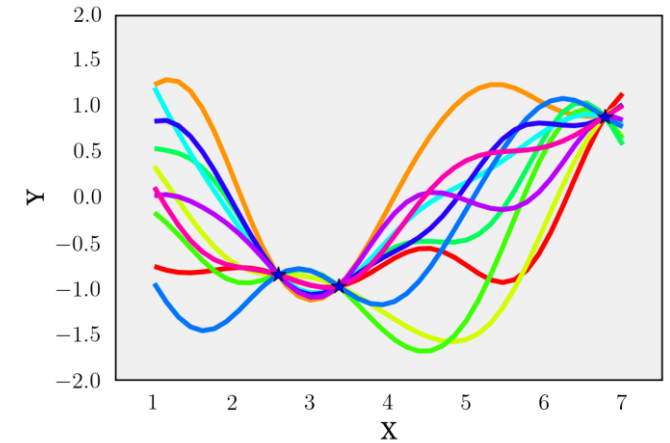
Използване на данни



$$f_* \sim N(0, \Sigma(X_*, X_*))$$



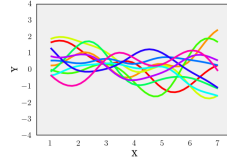
$$\{X, f\}$$



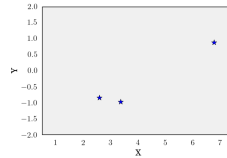
$$f_* | X_*, X, f \sim N(\mu_{f_*}, \Sigma_{f_*})$$

Използване на данни

$$f_* \sim N(0, \Sigma(X_*, X_*))$$

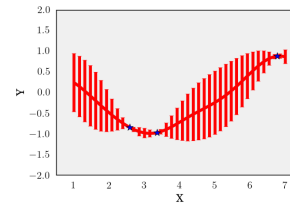
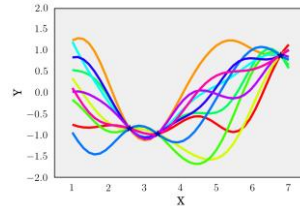


$$\{x_n, y_n\}_{n=1}^N \quad \{X, f\}$$



$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \Sigma(X, X) & \Sigma(X, X_*) \\ \Sigma(X_*, X) & \Sigma(X_*, X_*) \end{bmatrix}\right)$$

$$f_* | X_*, X, f \sim N(\mu_{f_*}, \Sigma_{f_*})$$

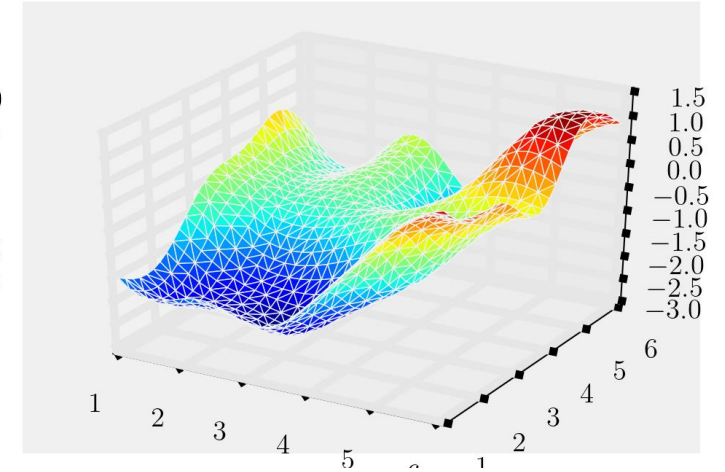
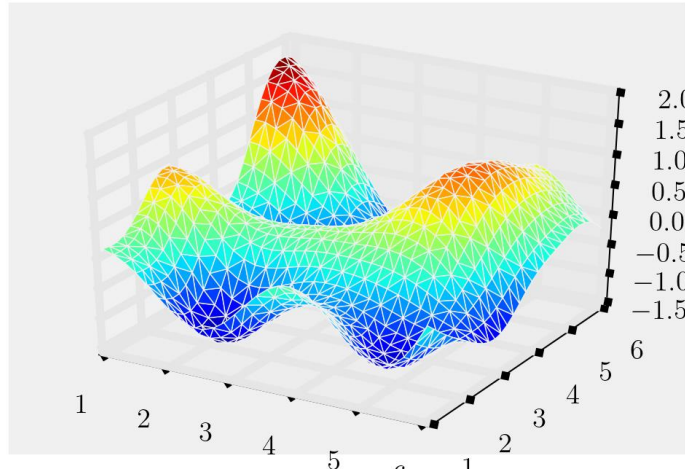


$$\Sigma_{f_*} = \Sigma(X_*, X_*) - \Sigma(X_*, X)\Sigma(X, X)^{-1}\Sigma(X, X_*)$$

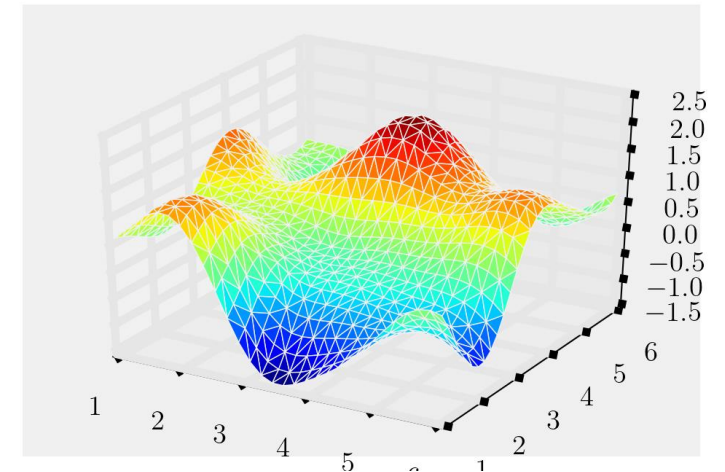
$$\mu_{f_*} = \Sigma(X_*, X)\Sigma(X, X)^{-1}f$$

Двуизмерни данни

$$\text{cov}(y_n, y_{n'}) = k(x_n, x_{n'}) + \sigma_v^2 \delta_{nn'}$$



$$k(x_n, x_{n'}) = \sigma_f^2 \exp\left(-\sum_{d=1}^D \frac{1}{2l^2} (x_{dn} - x_{dn'})^2\right)$$



Моделиране с Гаусови процеси

Да кажем, че искаме да направим **модел на времето в България**

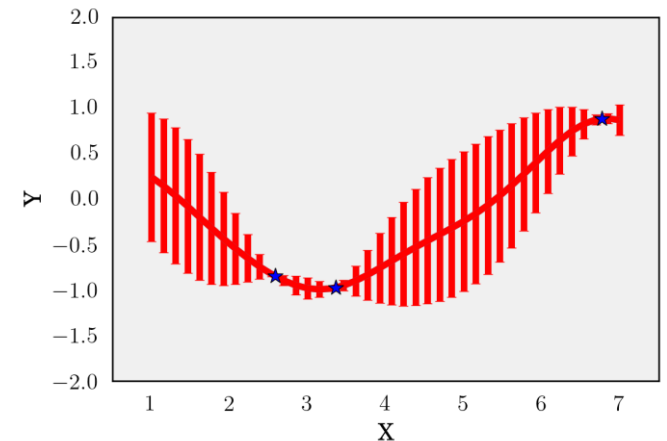
Лимитации:

Може да направим **много малко измервания**

От къде да съберем измервания за да направим най-точната прогноза?

След като имаме измервания, как правим самата прогноза?

Колко сме сигурни, че нашата прогноза е вярна?



Гаусовите процеси има следните желани свойства:

- **Полезен bias** от кернел функцията (близки локции имат подобна температура)
- **Измерване на несигурността** на предположенията (колко точно е нашето предположение за дадено място)
- **Избиране на следваща точка за измерване** (къде да сложим нов сензор за измервания)
- **Мързелива оценка** (измерване на температурата на едно място, обновяване на нашия модел, избиране на „най-несигурното“ място, измерване, обновяване, избиране на ново място и т.н.)

Примери

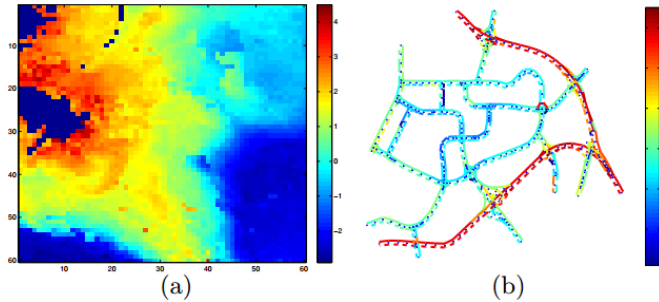


Figure 1: Real-world non-stationary environmental phenomena: (a) Plankton density (chl-a) phenomenon (measured in mg/m^3) in log-scale in Gulf of Mexico, and (b) traffic (road speeds) phenomenon (measured in km/h) over an urban road network.

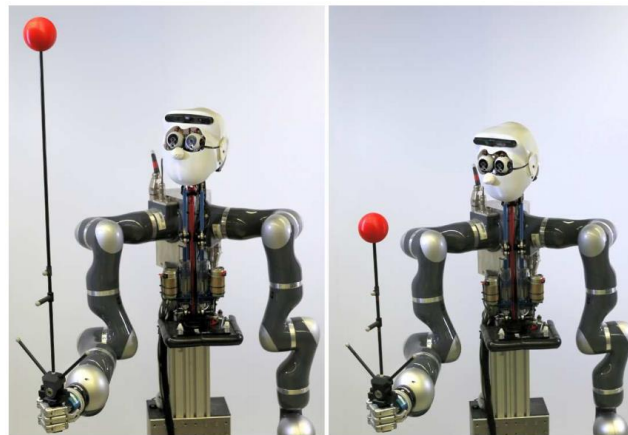
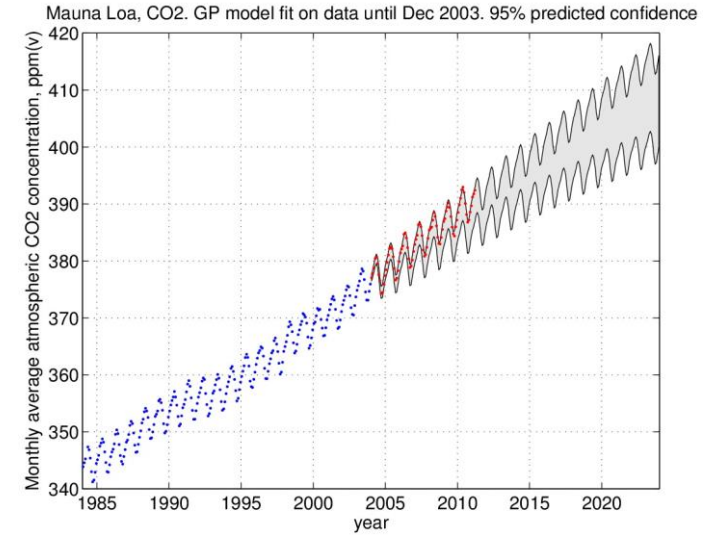
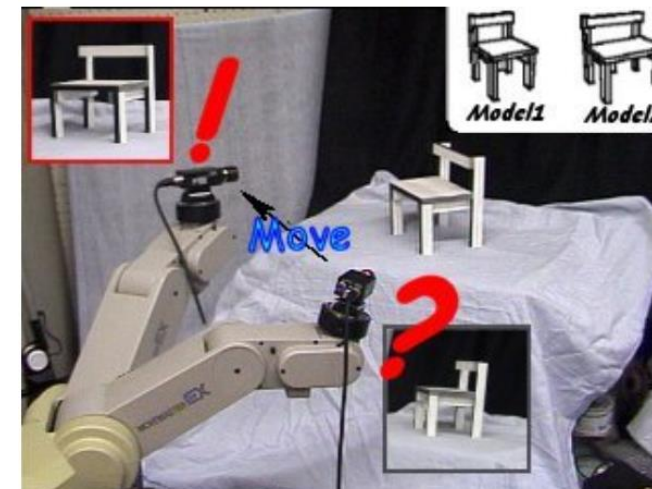
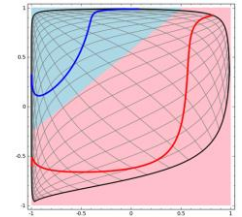
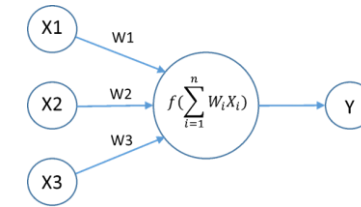
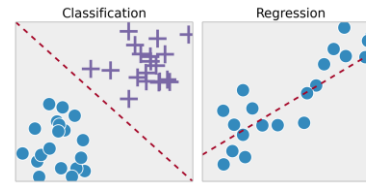


FIGURE 1.1: Robot Apollo balancing two inverted poles. These experimental setups are used as demonstrators of the automatic tuning framework.

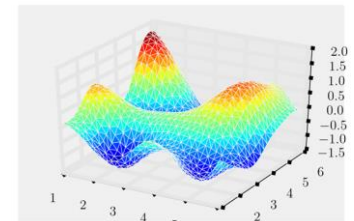
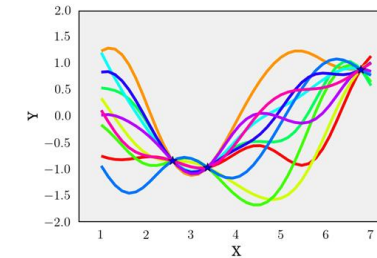
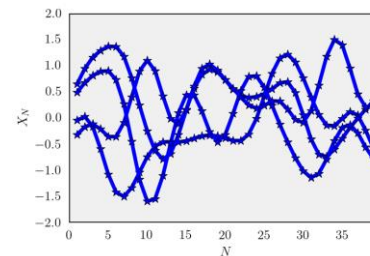
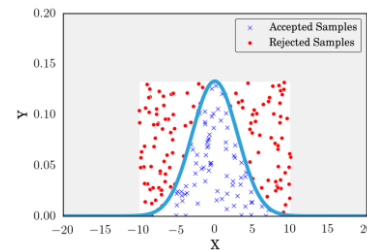


План за лекцията

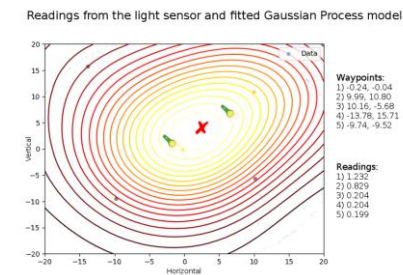
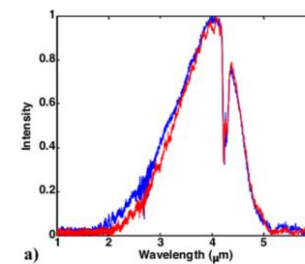
Машинно самообучение и невронни мрежи



Гаусови процеси



Активно семплиране с дрон използвайки Гаусови процеси за намиране на опасни газове



Заклучение

Структура на проекта

Heriot-Watt University



Marius Rutkauskas
Ph.D. Nonlinear microscopy



Luke Maidment*
Eng.D. (Dstl)



Ultrafast Optics Group

Ultrafast lasers and instrumentation for microscopy, metrology and spectroscopy

Track record of successful CDE projects in mid-IR stand-off chemical detection

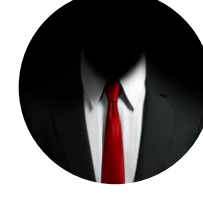
Edinburgh University



Ram Ramamoorthy
Group Head



Martin Asenov
PhD. UAV-based
active sensing
(EPSRC CDT)



Carson Vogt
PhD. student

Robust Autonomy and Decisions Group

Motion synthesis and robot learning in dynamic environments

Software libraries and knowhow for reactive motion planning and active model learning

Мотивация

- **Активно семплиране с дрон**
 - Използване на дрон за събиране на информация – полезен инструмент за различни науки/индустрии като енергийна, опазване на околната среда и т.н.
 - Използване на машинно самообучение за по-ефективно събиране на информация и моделиране на желания феномен
 - Дронове, батерии и бързи портативни компютри стават все по-популярни технологии и широко достъпни
 - „Повторна употреба“ на модела – използване на различни сензори

Мотивация прод.

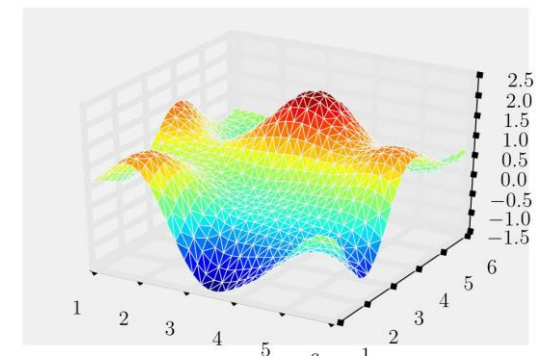
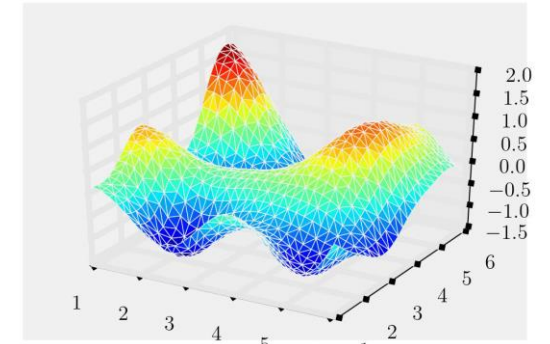
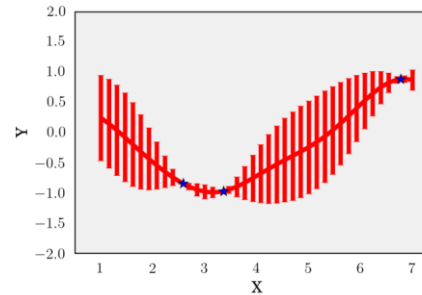
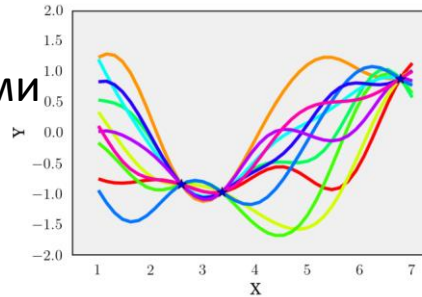
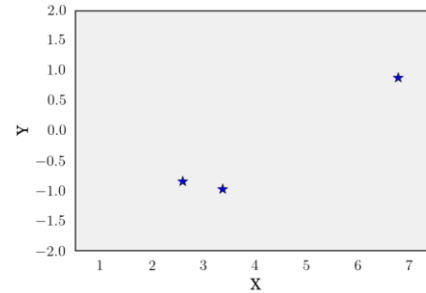
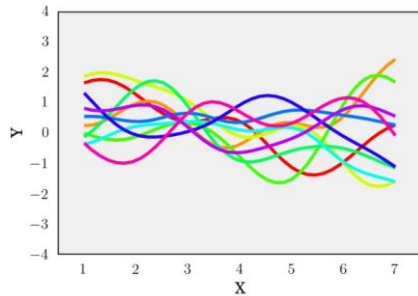
- **Примерен проект - намиране на опасни газове**
 - Намирането на източник на експлозивни/отровни газове е трудно
 - Опасно е да се изпращат хора
 - Важно е да се намери източника възможно най-бързо
 - Превенция на по-сериозно разпространение



Гаусови процеси (обобщение)

- **Дефиниция и полезни свойства**

- Използвайки mean и covariance функция -> дистрибуция от функции
- Използване на данни – запазване свойствата на функцията, докато те минават през нашите точки от данните
- Измерване несигурността на модела
- Непараметричен модел
- Може да се използва за многомерни проблеми



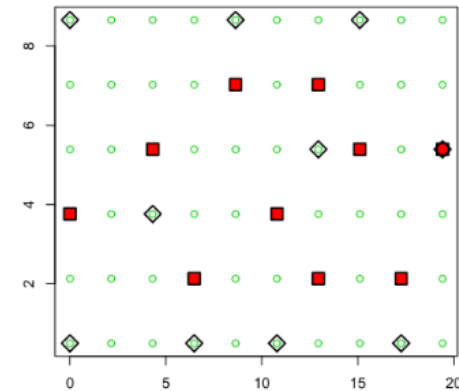
Моделиране с Гаусови Процеси

- Регресия с Гаусови процеси – селектиране на модел
 - Избиране на кернел функция
 - Избиране на хиперпараметри за кернел функцията

Обикновено разчитаме на информация, което имаме за проблема. Например, газовете имат бавно променяща се концентрация като функция на време и пространство, затова може да използваме squared exponential кернел функция.

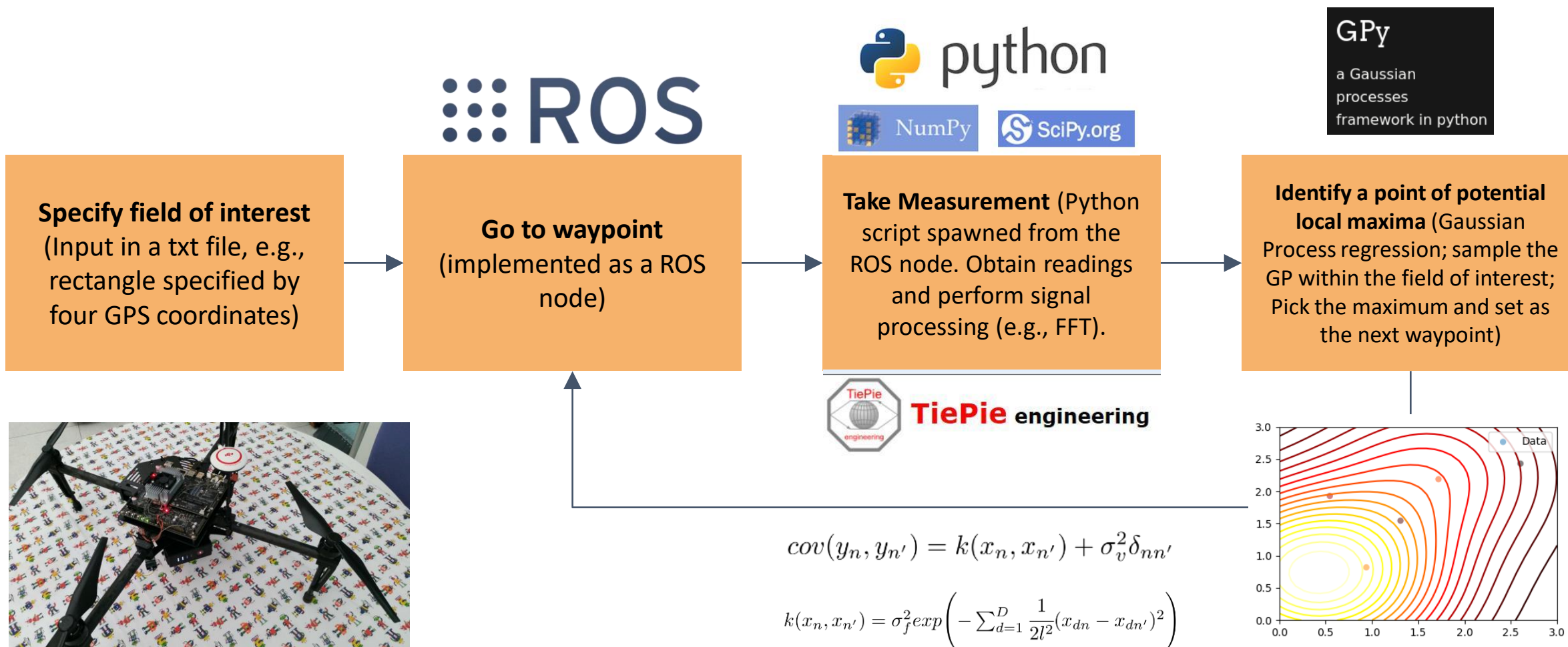
Може също така да направим малък експеримент и така да изберем нашите параметри.

- Избиране на следваща точка за измервания
 - Максимизиране на ентропията на всички измервания
 - Максимизиране на взаимната информация между измерванията
 - „Трудност“ при правене на измерване*

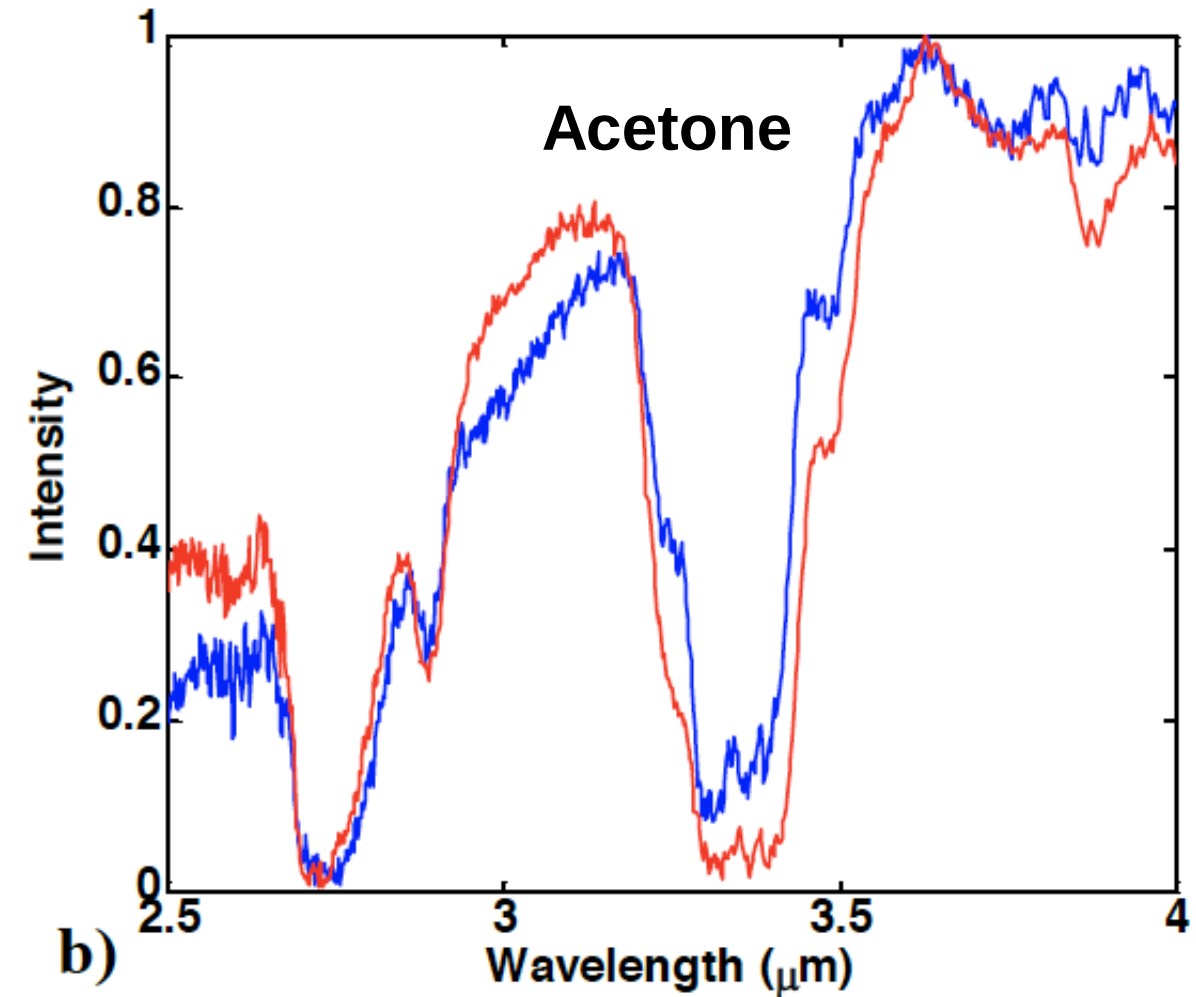
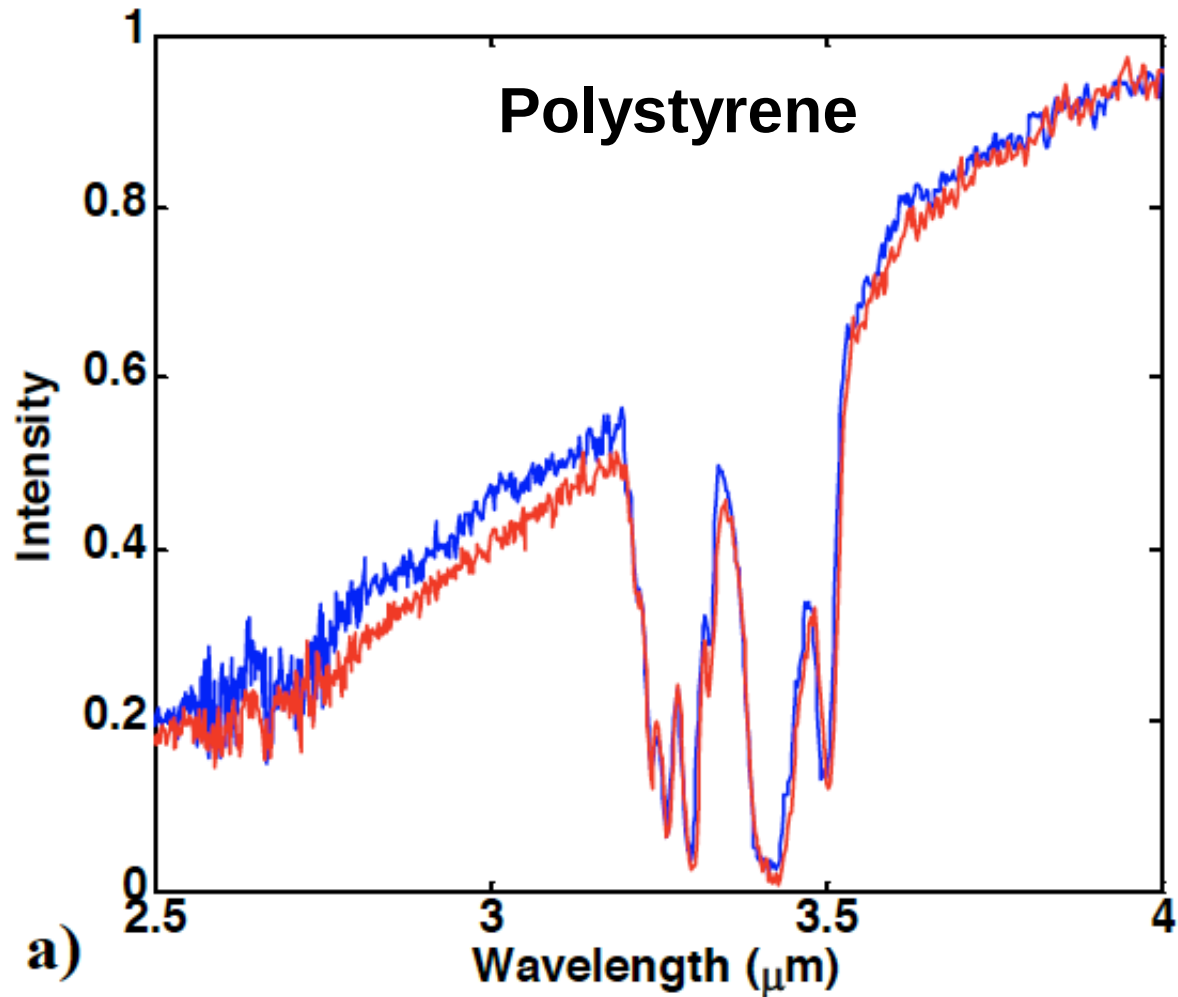


*това е по-голям проблем при наземни работи, не толкова при дронове

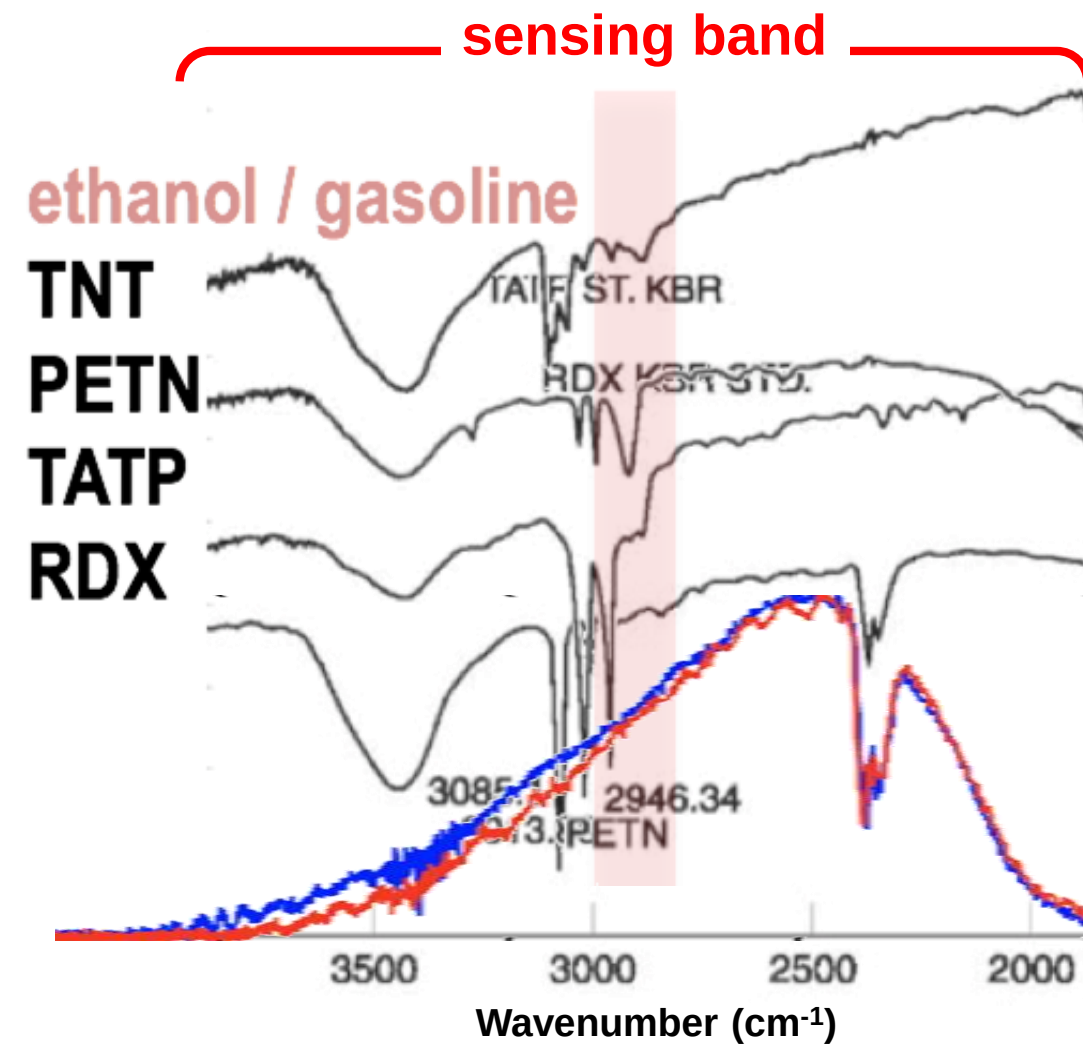
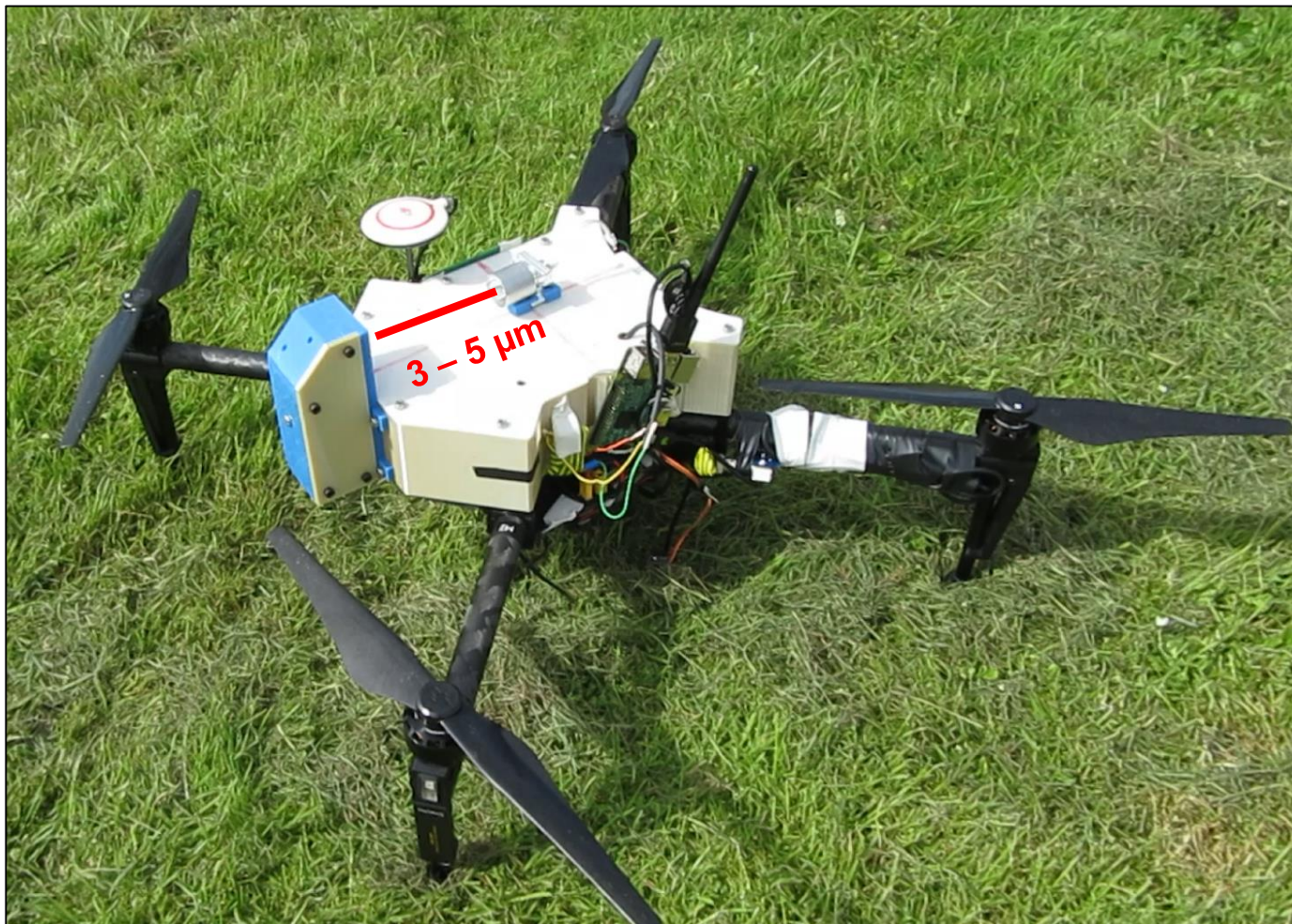
Софтуерна архитектура на проекта



Spectroscopy

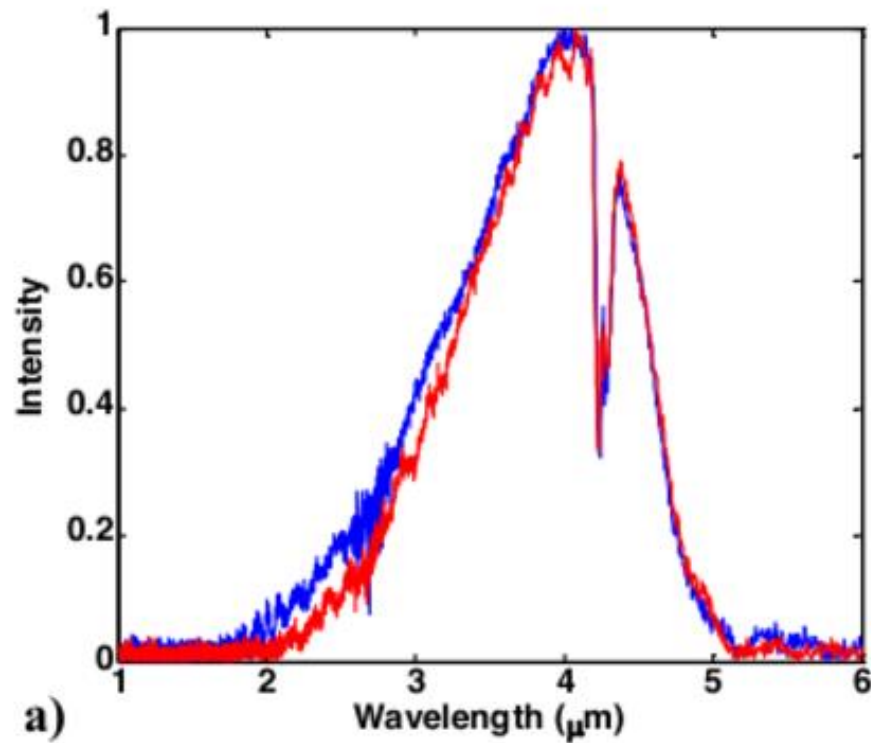


Концепция: Mid-IR spectrometer върху дрон



Експерименти с химическия сензор

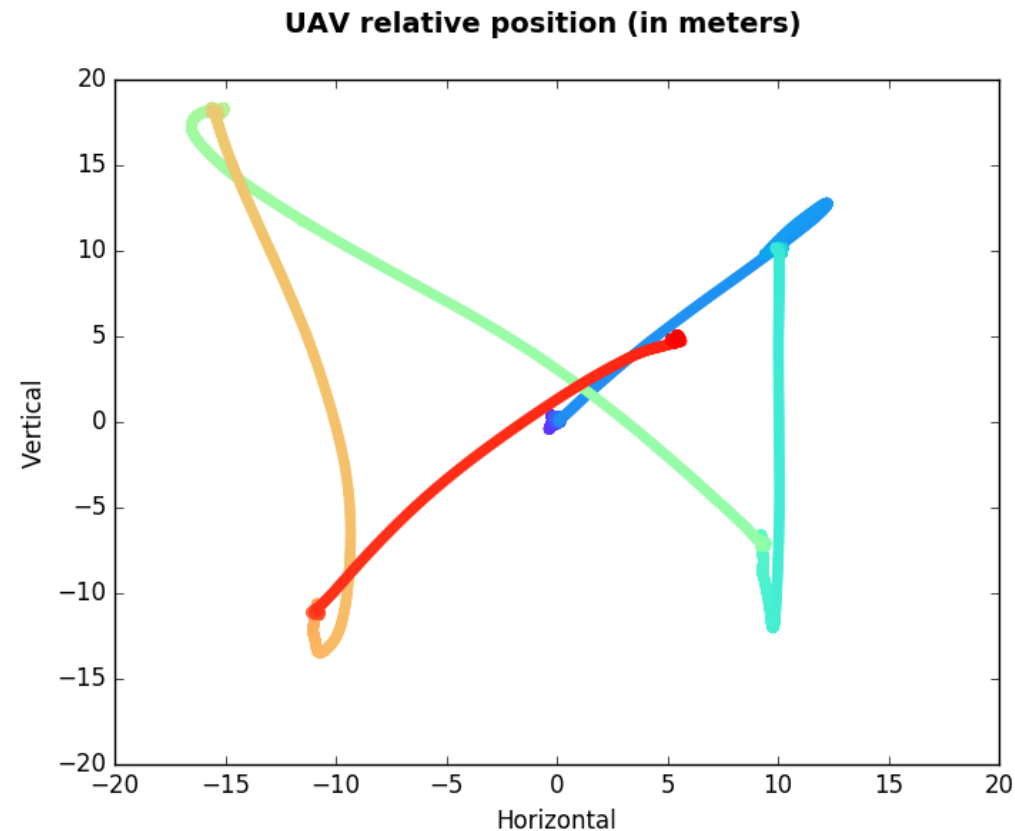
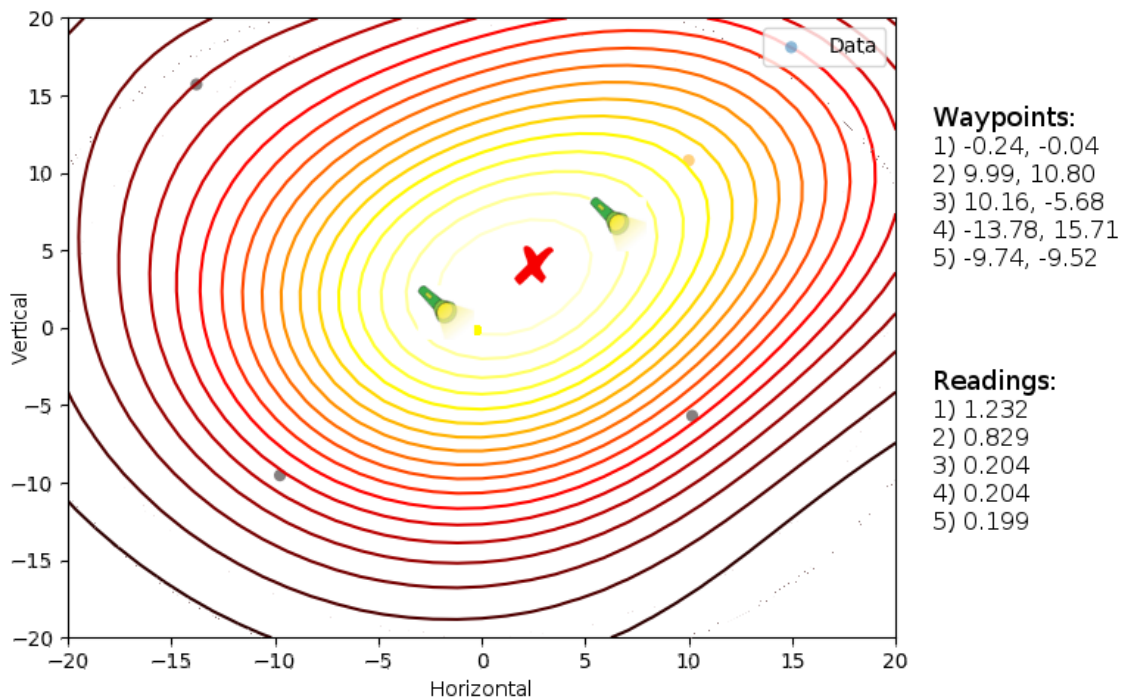
[Video](#) [Video2](#) [Video3](#)



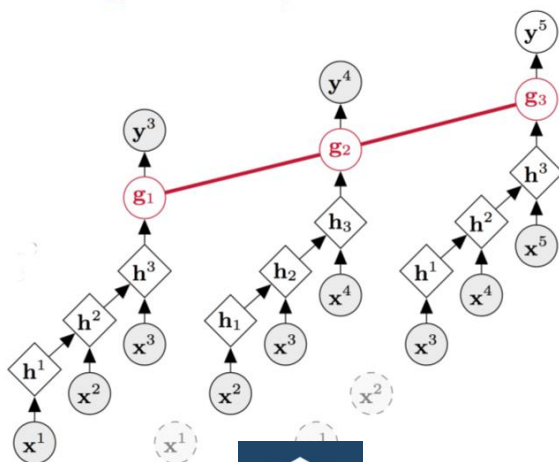
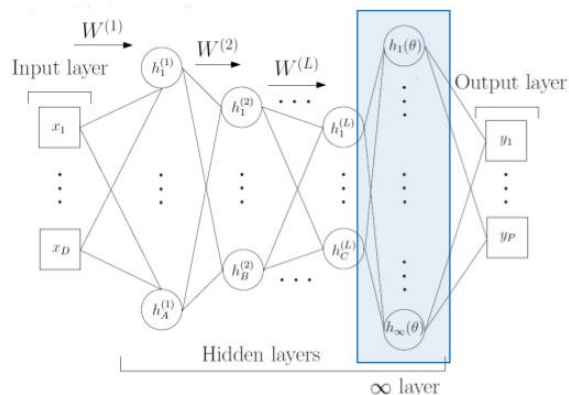
Експерименти със сензор за светлина

[Video](#)

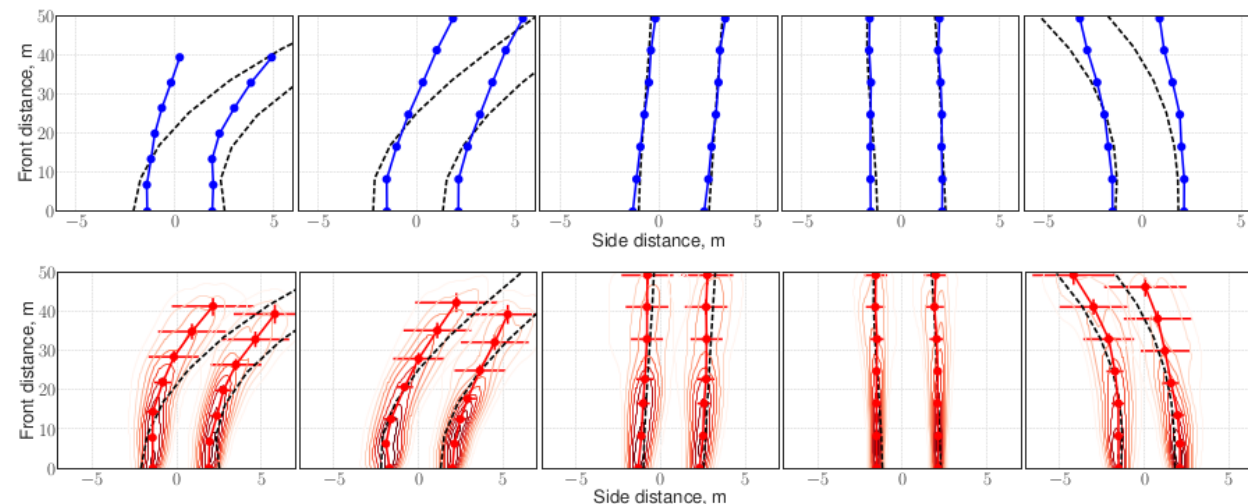
Readings from the light sensor and fitted Gaussian Process model



Комбиниране на Гаусови процеси и невронни процеси за следене на пътно платно



Lane prediction: LSTM vs GP-LSTM



Кода за Гаусови процеси е в Github!

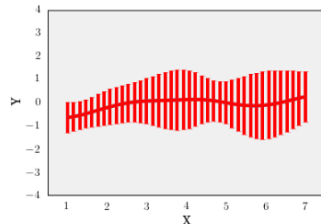
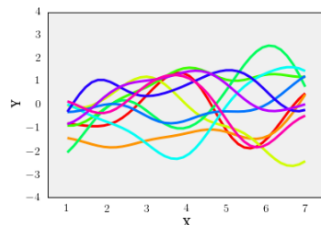
- Different visualizations
- Inference from data
- 1D and 2D regression

```
samples = 10
ndim = 40
length = 7
sigma_v = 0
l=1
sigma_f=1
uniform=False

dots=False
usecolors=True
lw=3
xname="X"
yname="Y"
ylim=4
filename="presentation5"

x,y,cov = genGP(samples=samples, ndim=ndim, length=length, sigma_v=sigma_v, sigma_f=sigma_f, usecolors=usecolors, lw=lw, xname=xname, yname=yname, ylim=ylim, filename=filename)

/home/martin/Dropbox/MSc/ASR/report/gaussian.py:169: RuntimeWarning: covariance is not positive-semidefinite.
y=np.random.multivariate_normal(mu, cov, samples).T
```

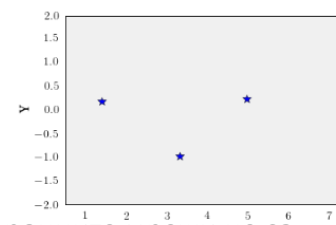
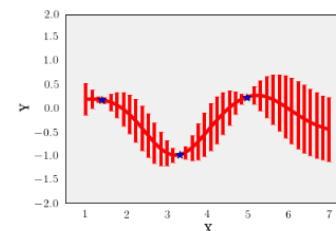
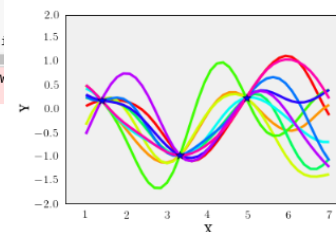


```
mu,cov = inference(sample_x, sample_y, x, cov, ndim=ndim, length=length, sigma_v=sigma_v, l=l, sigma_f=sigma_f)
```

```
sample_x.shape
sample_y.shape
data = np.array((sample_x,sample_y))

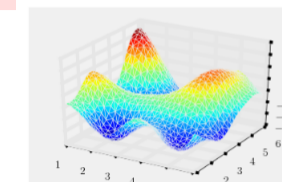
y=np.random.multivariate_normal(mu, cov, samples).T
plotGP(x,y,xlim=length,ylim=2,data=data,filename="presen

/usr/lib/python3.5/site-packages/ipykernel/_main_.py:1
if __name__ == '__main__':
```



```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_trisurf(x.flatten(),y.flatten(),z.flatten(), cmap=cm.jet)
plt.savefig('../FinalReport/Pictures/Introduction/2dregression.png', dpi=200,bbox_inches='tight')
fig.show()

/usr/lib/python3.5/site-packages/matplotlib/figure.py:397: UserWarning: matplotlib is currently using a non-GUI backen
d, so cannot show the figure
"matplotlib is currently using a non-GUI backend, "
```

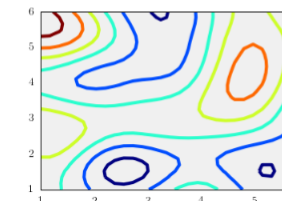


```
from matplotlib import cm

samples=1
ndim=30
cov,x,y = sigma(ndim,uniform=False)
mu=np.zeros(ndim*ndim)
z=np.random.multivariate_normal(mu, cov, samples).T
z = z.reshape(ndim,ndim).T
x = x.reshape(ndim,ndim).T
y = y.reshape(ndim,ndim).T

surf = plt.contour(x,y,z,cmap=cm.jet)
fig.colorbar(surf, shrink=0.5, aspect=5)
plt.savefig('../FinalReport/Pictures/Introduction/2dregression.png', dpi=200,bbox_inches='tight')
plt.show()
```

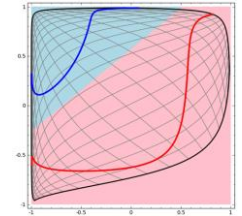
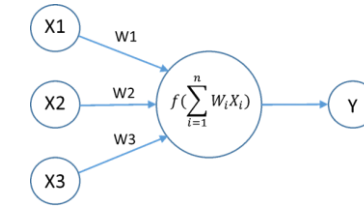
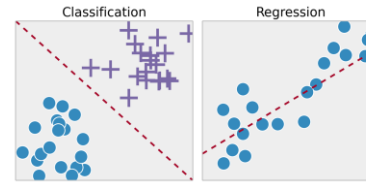
```
/usr/lib/python3.5/site-packages/ipykernel/_main_.py:7: RuntimeWarning: covariance is not positive-semidefinite.
```



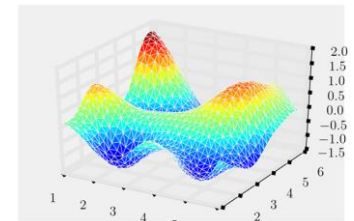
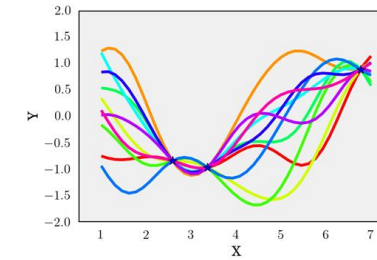
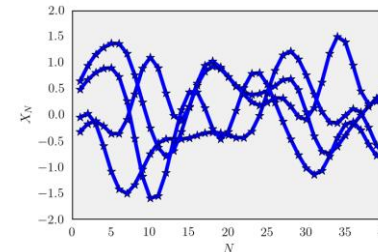
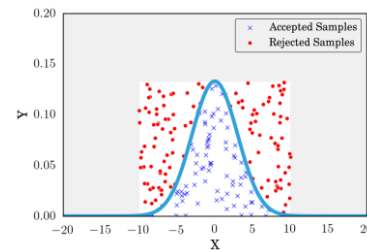
<https://github.com/masenov/gaussian-processes-introduction>

Заклучение

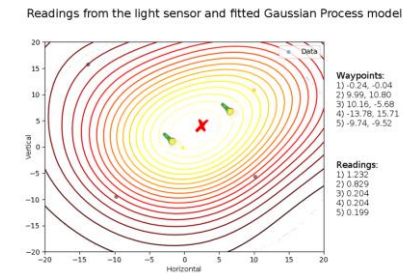
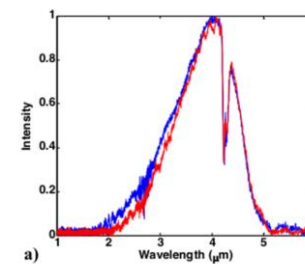
Машинно самообучение и невронни мрежи



Гаусови процеси

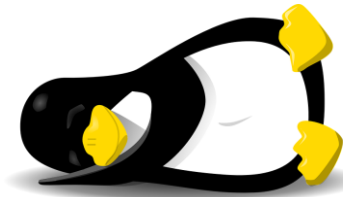


Активно семплиране с дрон използвайки Гаусови процеси за намиране на опасни газове



Благодаря за вниманието!

Някакви въпроси?



За контакти:

m.a.asenov@gmail.com

<https://masenov.github.io>

Код:

<https://github.com/masenov/gaussian-processes-introduction>