Gaussian Processes in MAB algorithms

Exercise Lecture

Contents

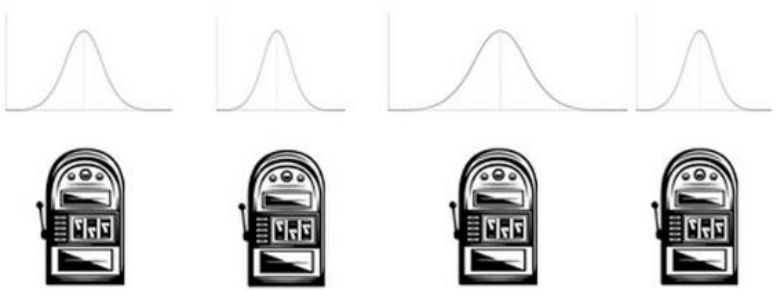
Why combining Gaussian Processes with MABs?

Standard Mab algorithms **perform poorly** in many complex real-world problems where

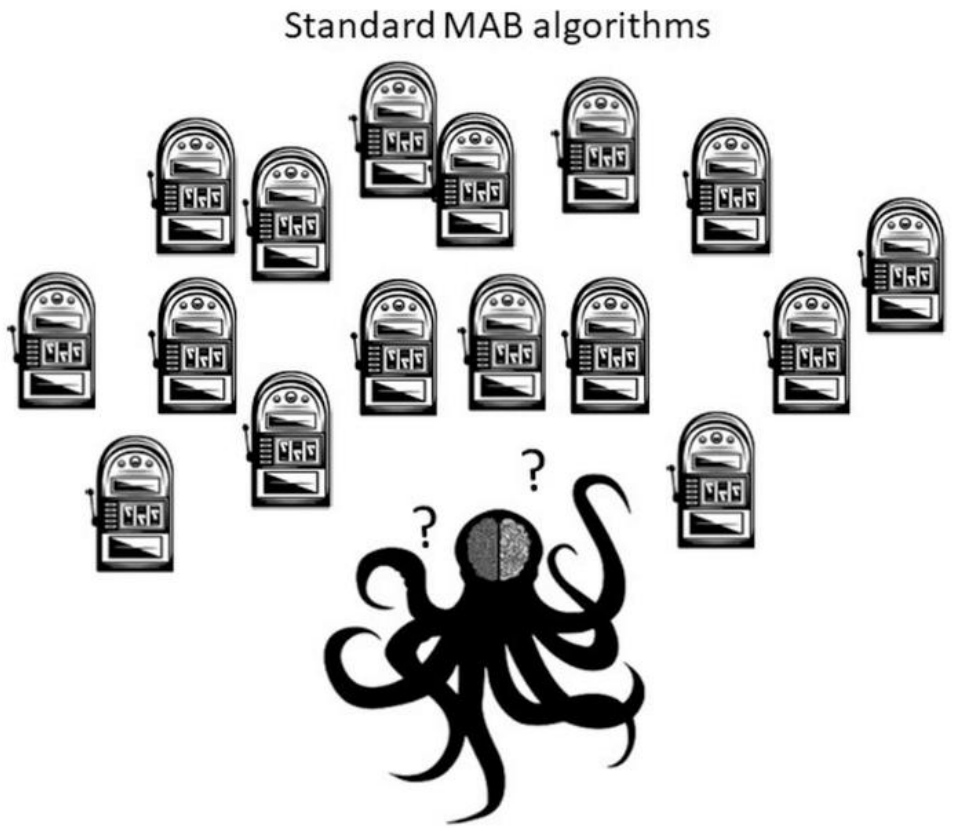
the **arms space is large**

Why combining Gaussian Processes with MABs?

Standard MAB algorithms

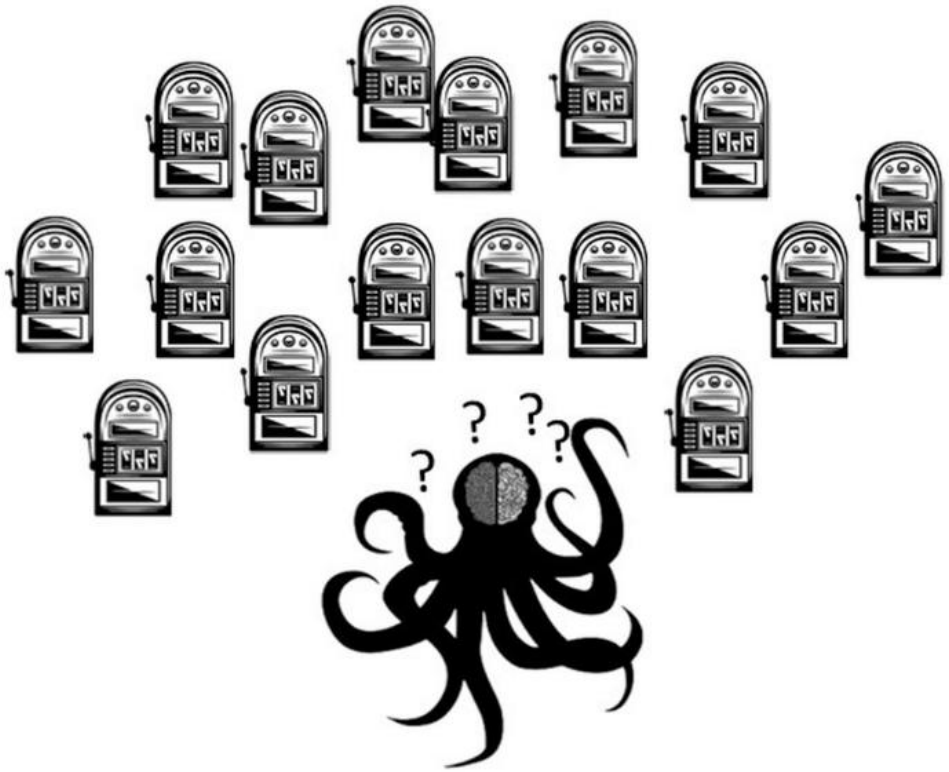


Why combining Gaussian Processes with MABs?

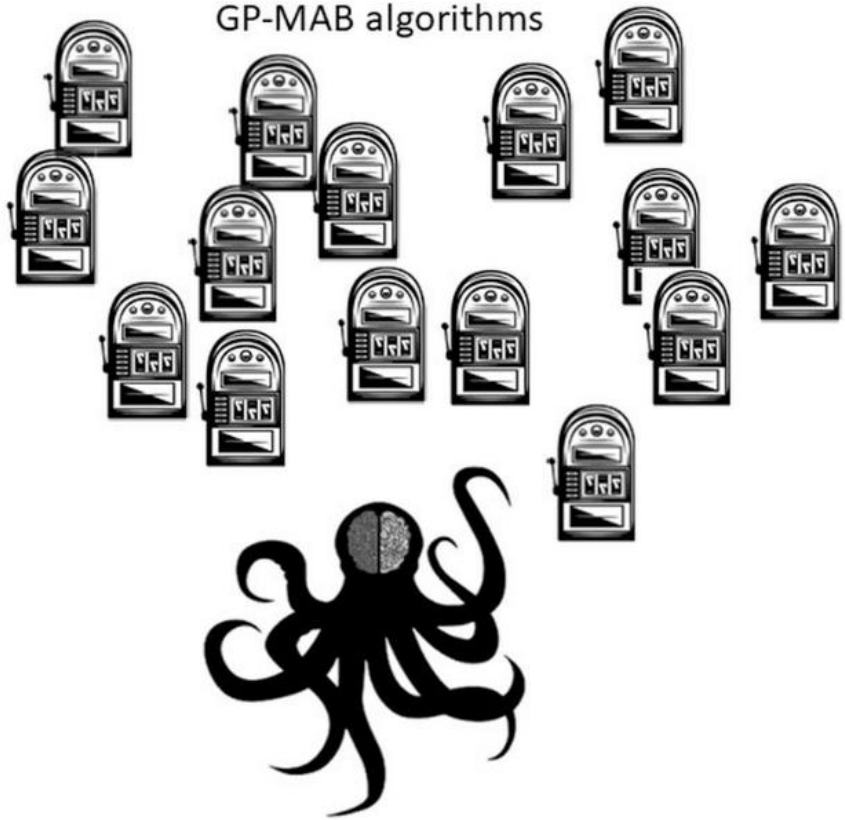


Why combining Gaussian Processes with MABs?

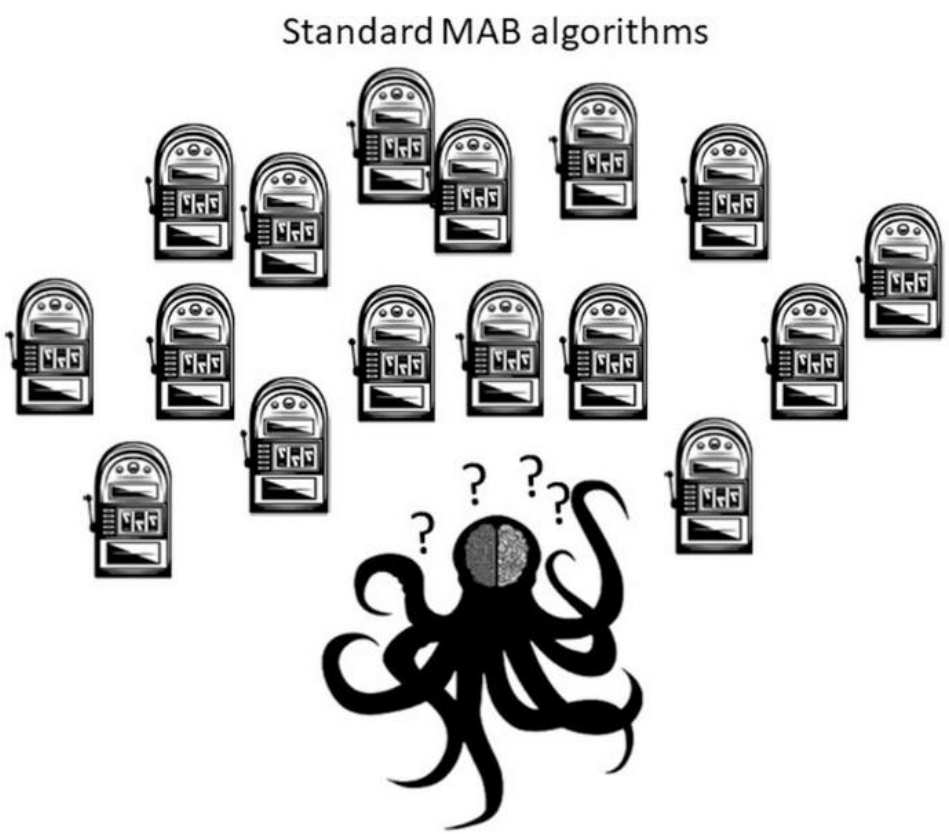
Standard MAB algorithms

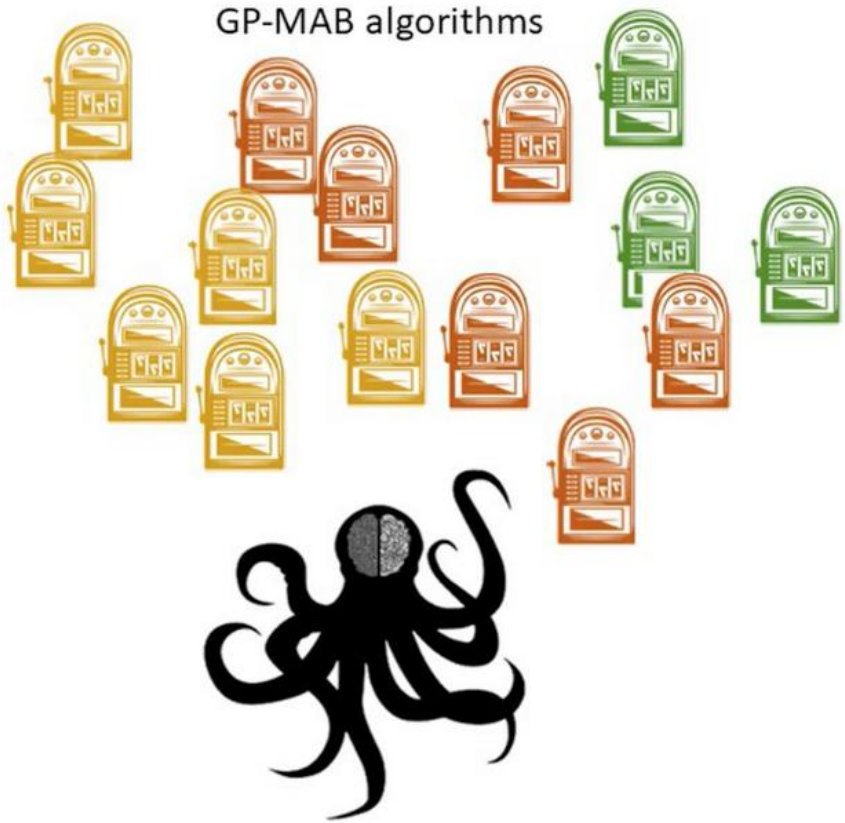


GP-MAB algorithms



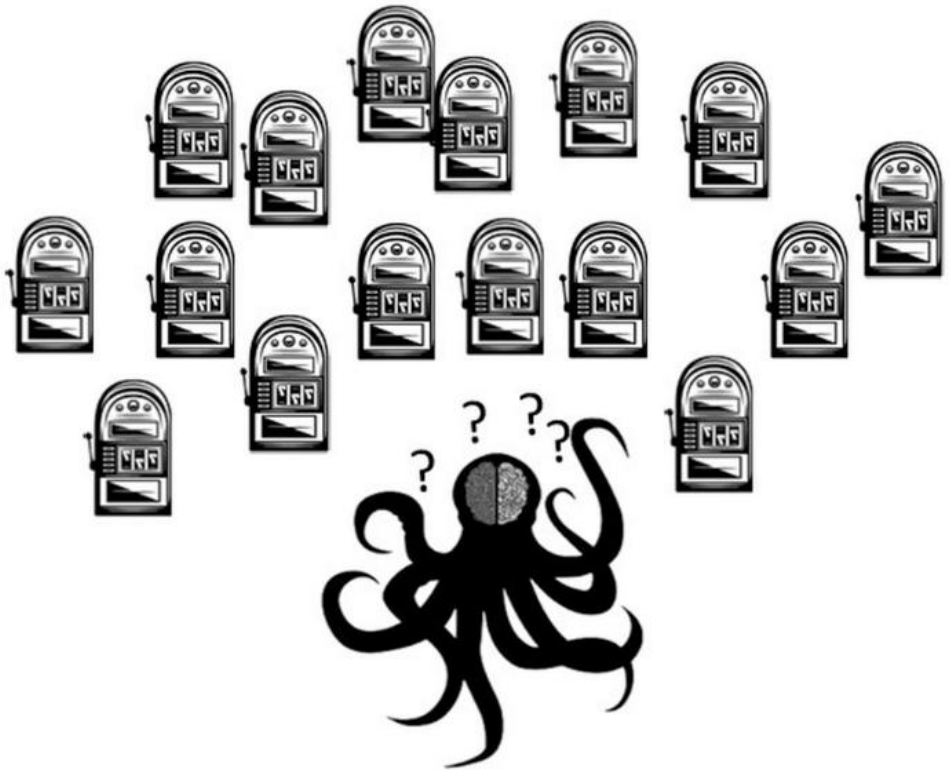
Why combining Gaussian Processes with MABs?



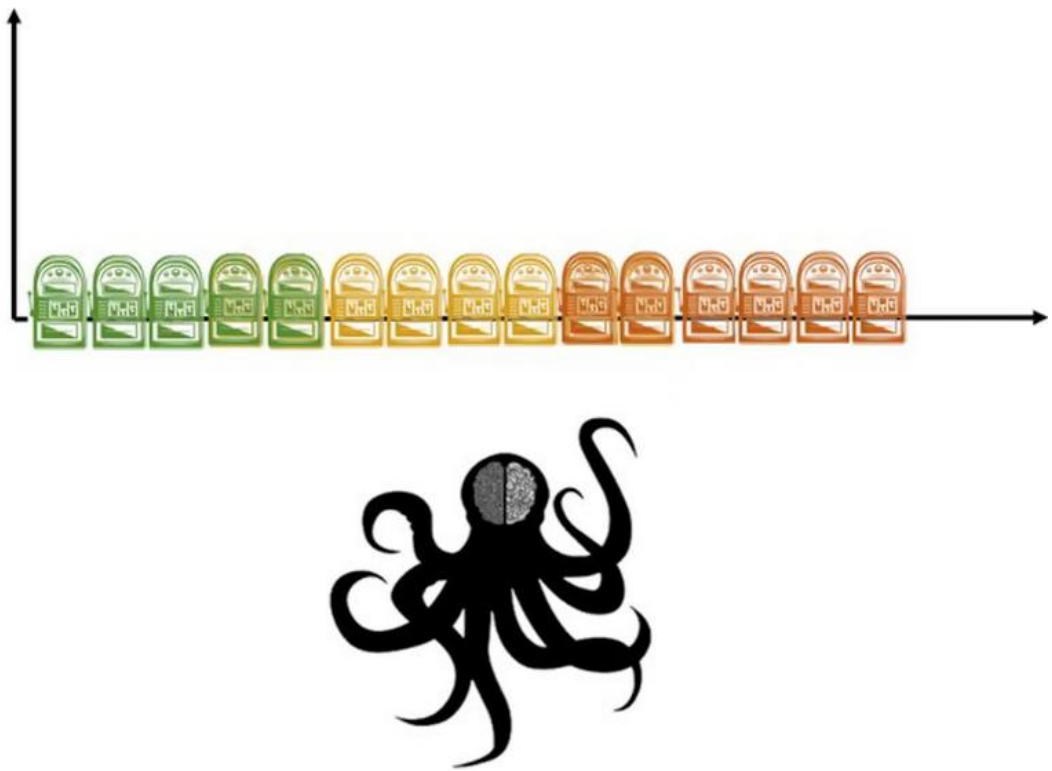


Why combining Gaussian Processes with MABs?

Standard MAB algorithms

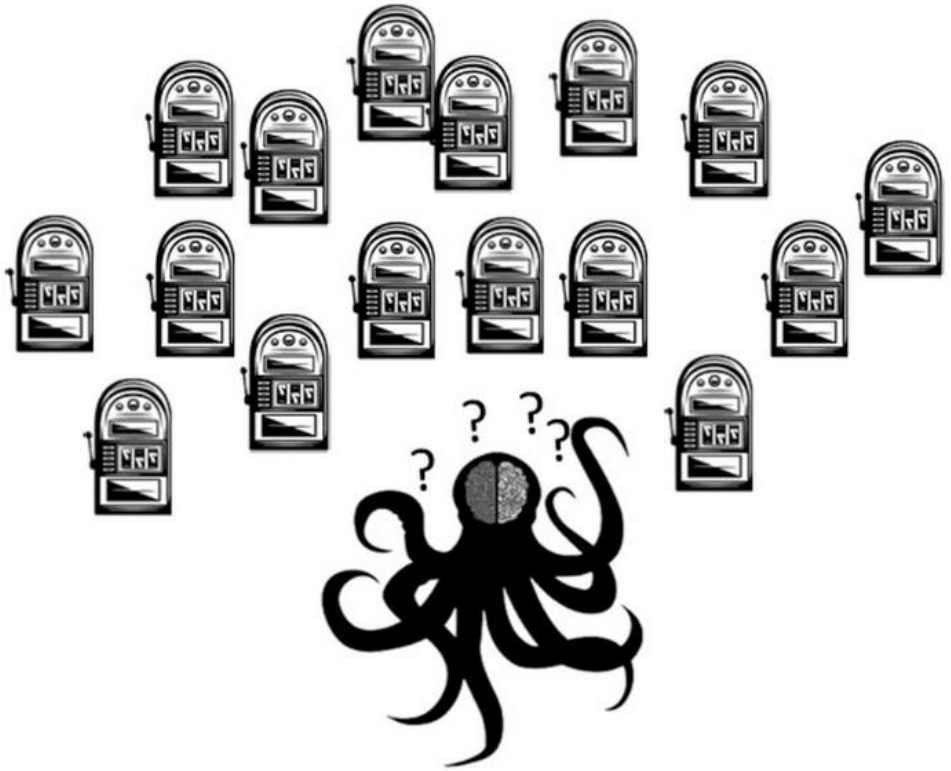


GP-MAB algorithms

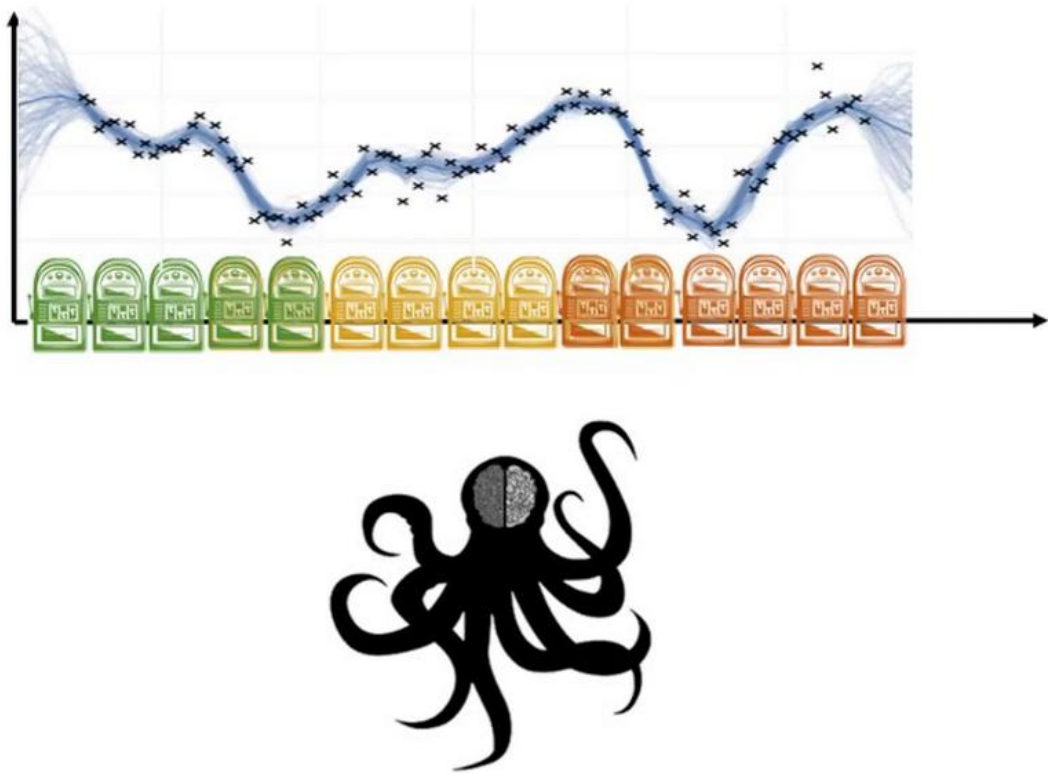


Why combining Gaussian Processes with MABs?

Standard MAB algorithms



GP-MAB algorithms



Why combining Gaussian Processes with MABs?

Standard Mab algorithms **perform poorly** in many complex real-world problems where the **arms space is large**

GP-Mab algorithms are suitable for **real-world problems** in which the **arms space is large** and the **arms are correlated**

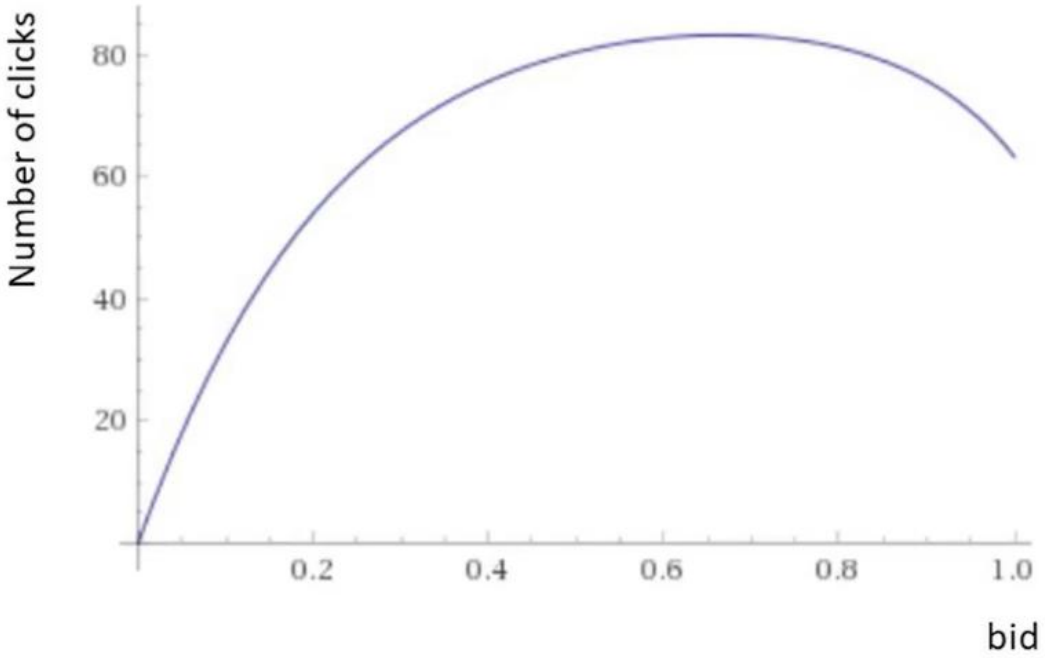
Example: Bidding Environment

Time horizon: 60 days

Fixed Daily budget

20 possible bids

Example: Bidding Environment



Bidding Environment Class

- The environment returns a stochastic reward (i.e., number of clicks) depending on the pulled arm (choosen bid)

- We have to specify a function that maps a bid value to the corresponding expected number of clicks

- The number of clicks is not deterministic

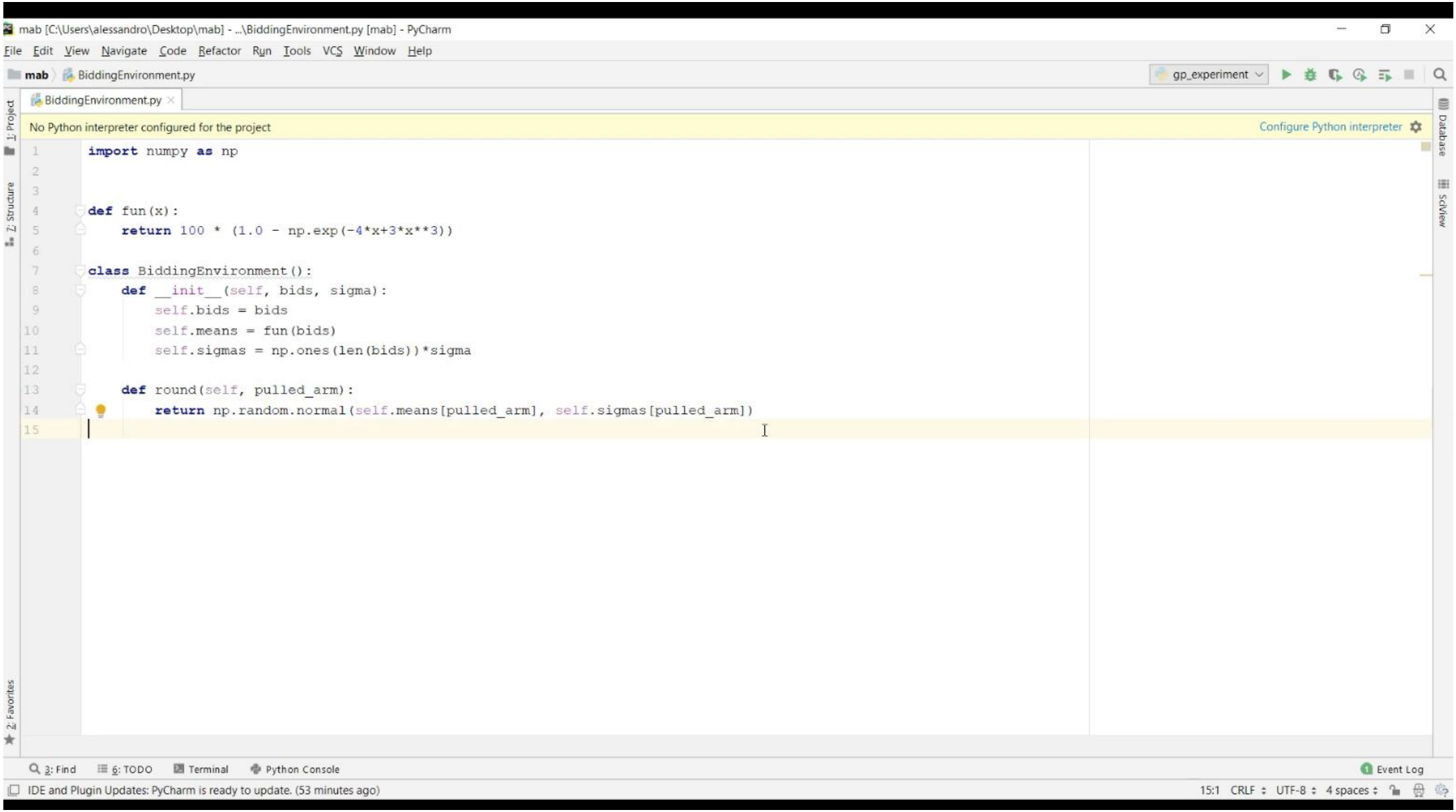
Let’s implement it!

Gaussian TS-Learner Class

It extends the class Learner we implemented in the First Lecture

The GTS-Learner, at each round, updates the parameters of a **normal distribution**

associated to the pulled arm

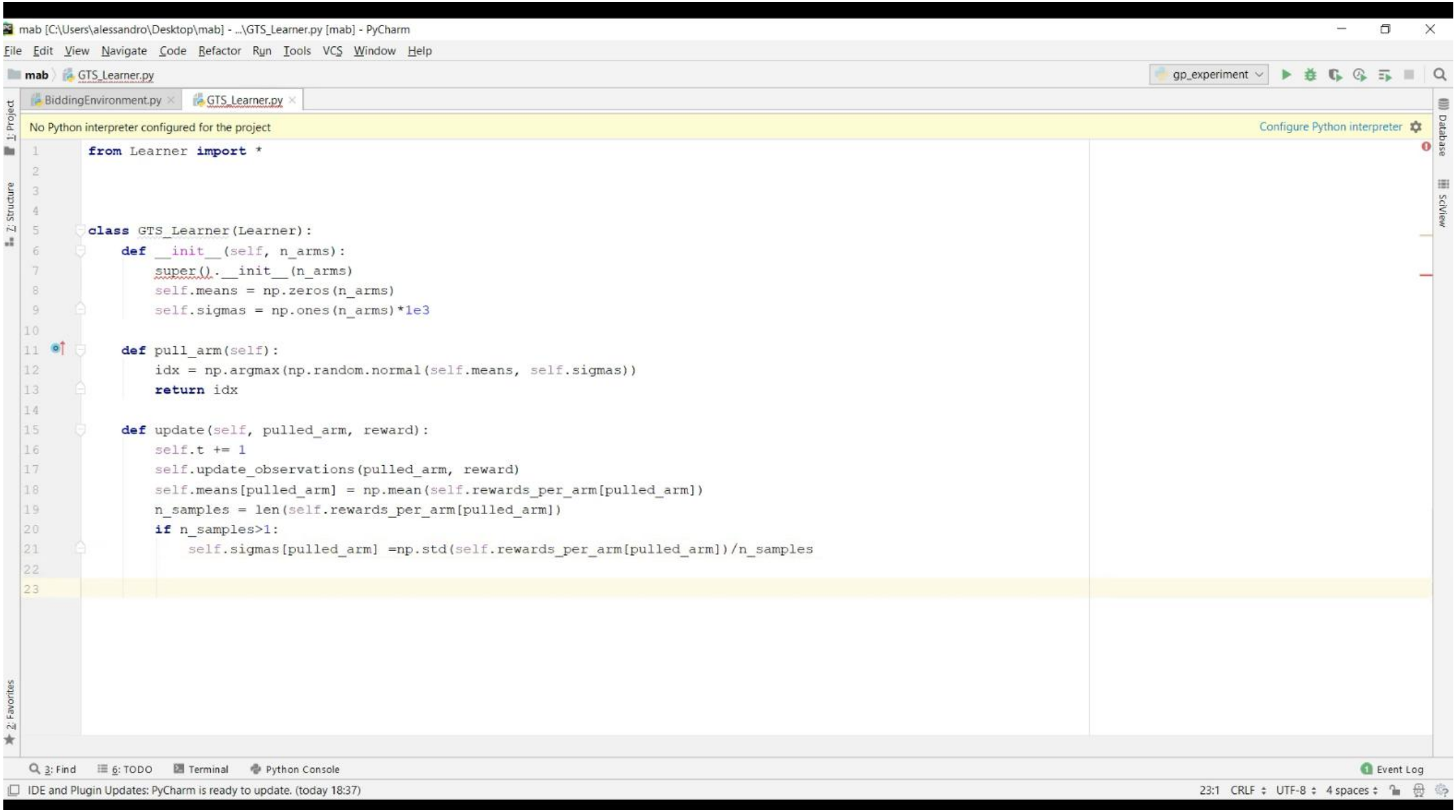

mab [C:\Users\alessandro\Desktop\mab] - ...\BiddingEnvironment.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab BiddingEnvironment.py
gp_experiment
BiddingEnvironment.py
No Python interpreter configured for the project
Configure Python interpreter
import numpy as np
def fun(x):
return 100 * (1.0 - np.exp(-4*x+3*x**3))
class BiddingEnvironment():
def __init__(self, bids, sigma):
self.bids = bids
self.means = fun(bids)
self.sigmas = np.ones(len(bids))*sigma
def round(self, pulled_arm):
return np.random.normal(self.means[pulled_arm], self.sigmas[pulled_arm])
1: Project
2: Favorites
7: Structure
Database
SciView
3: Find
6: TODO
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (53 minutes ago)
15:1 CRLF UTF-8 4 spaces

Gaussian TS-Learner Class

It extends the class Learner we implemented in the First Lecture

The GTS-Learner, at each round, updates the parameters of a **normal distribution**

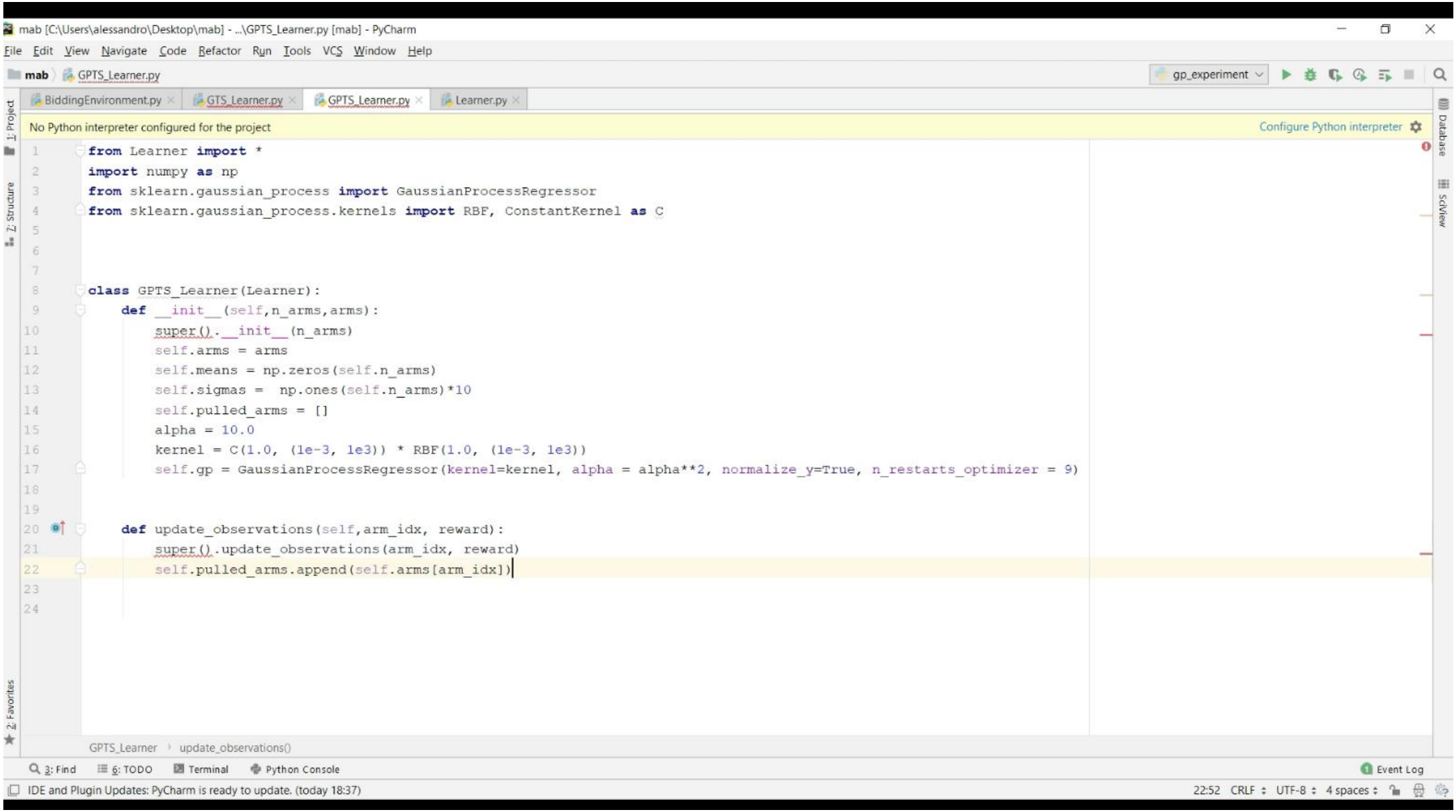
associated to the pulled arm

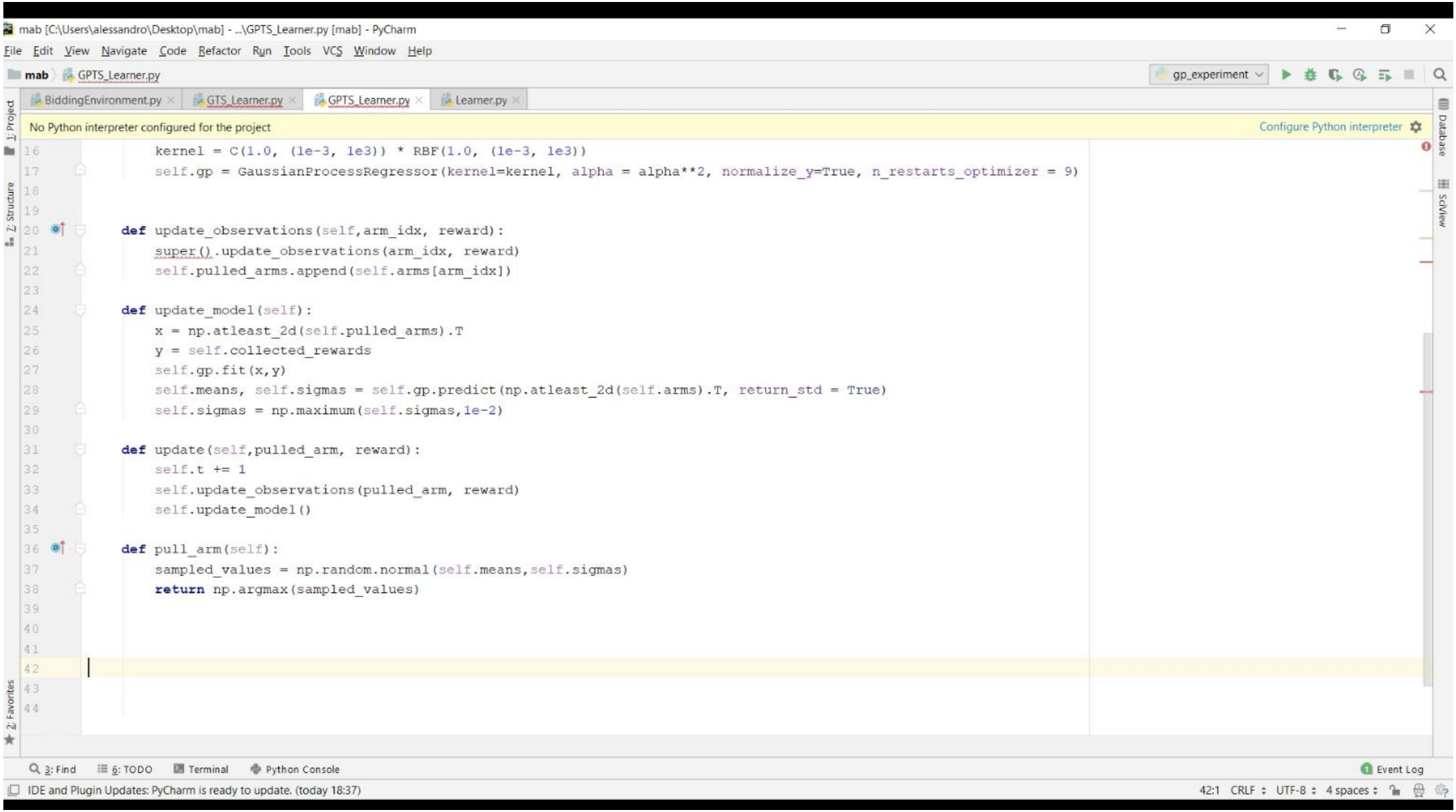

mab [C:\Users\alessandro\Desktop\mab] - ...\GTS_Learner.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab GTS_Learner.py
gp_experiment
BiddingEnvironment.py
GTS_Learner.py
No Python interpreter configured for the project
Configure Python interpreter
from Learner import *
class GTS_Learner(Learner):
def __init__(self, n_arms):
super().__init__(n_arms)
self.means = np.zeros(n_arms)
self.sigmas = np.ones(n_arms)*1e3
def pull_arm(self):
idx = np.argmax(np.random.normal(self.means, self.sigmas))
return idx
def update(self, pulled_arm, reward):
self.t += 1
self.update_observations(pulled_arm, reward)
self.means[pulled_arm] = np.mean(self.rewards_per_arm[pulled_arm])
n_samples = len(self.rewards_per_arm[pulled_arm])
if n_samples>1:
self.sigmas[pulled_arm] =np.std(self.rewards_per_arm[pulled_arm])/n_samples
1: Project
7: Structure
2: Favorites
Database
SciView
3: Find
6: TODO
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (today 18:37)
23:1 CRLF UTF-8 4 spaces

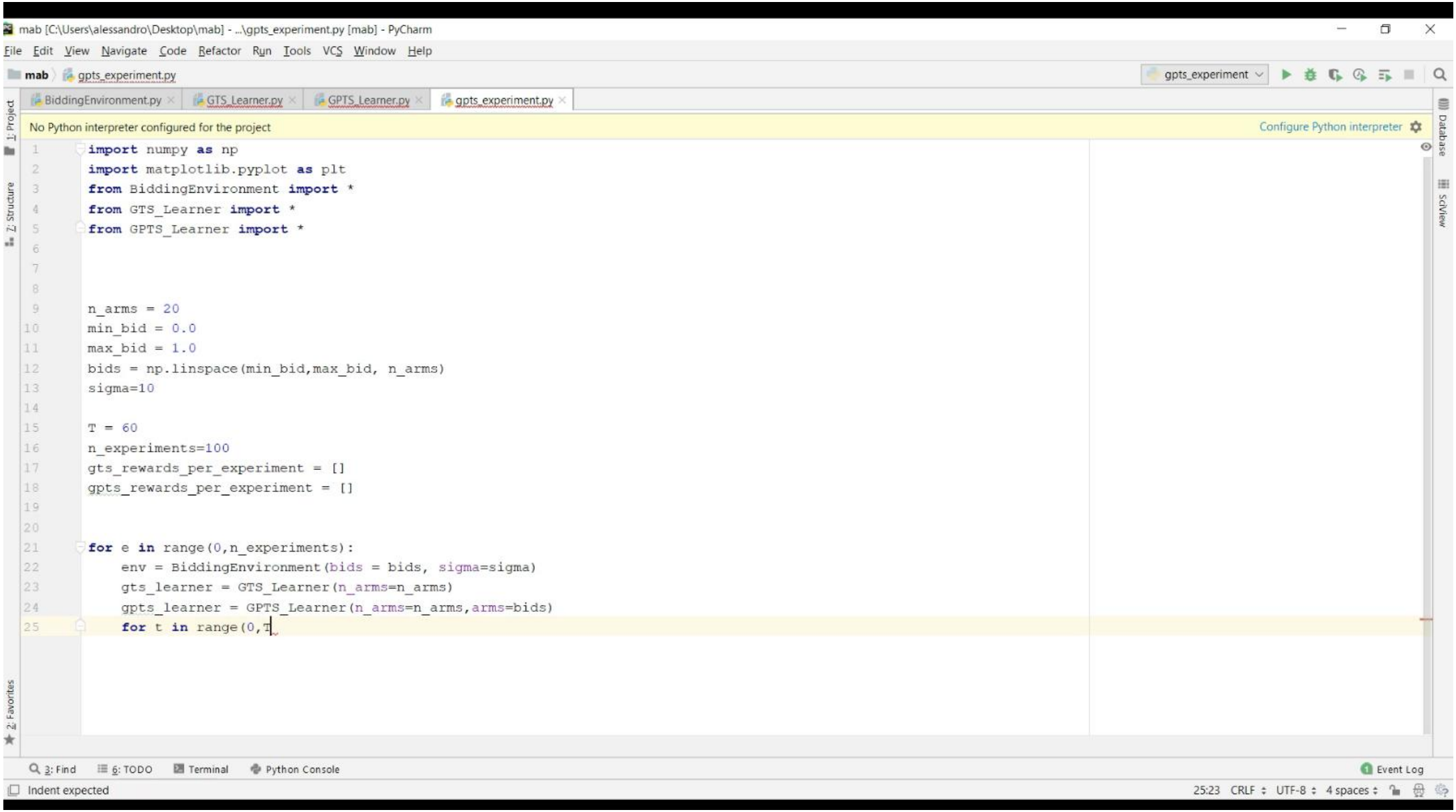
GPTS-Learner Class

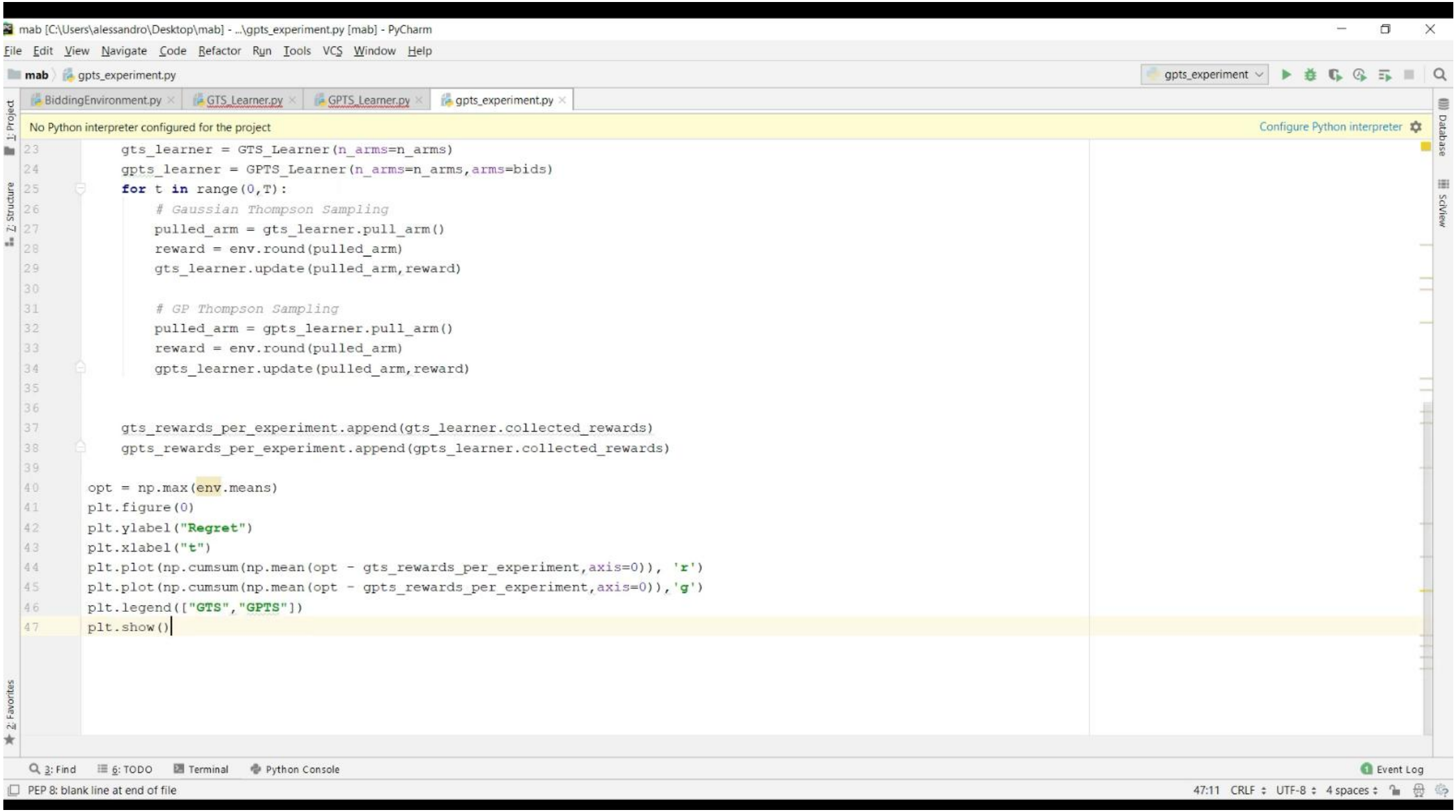
At each round, the GPTS-Learner has to fit a GP with the chosen bids as inputs and the observed number of clicks as targets

It uses the GP to predict the means and variances of the distributions associated to each arm (bid value)


mab [C:\Users\alessandro\Desktop\mab] - ...\GPTS_Learner.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab GPTS_Learner.py
gp_experiment
BiddingEnvironment.py
GTS_Learner.py
GPTS_Learner.py
Learner.py
No Python interpreter configured for the project
Configure Python interpreter
from Learner import *
import numpy as np
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF, ConstantKernel as C
class GPTS_Learner(Learner):
def __init__(self,n_arms,arms):
super().__init__(n_arms)
self.arms = arms
self.means = np.zeros(self.n_arms)
self.sigmas = np.ones(self.n_arms)*10
self.pulled_arms = []
alpha = 10.0
kernel = C(1.0, (1e-3, 1e3)) * RBF(1.0, (1e-3, 1e3))
self.gp = GaussianProcessRegressor(kernel=kernel, alpha = alpha**2, normalize_y=True, n_restarts_optimizer = 9)
def update_observations(self,arm_idx, reward):
super().update_observations(arm_idx, reward)
self.pulled_arms.append(self.arms[arm_idx])
GPTS_Learner › update_observations()
3: Find
6: TODO
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (today 18:37)
22:52 CRLF UTF-8 4 spaces
1: Project
7: Structure
2: Favorites
Database
SciView


mab [C:\Users\alessandro\Desktop\mab] - ...\GPTS_Learner.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab GPTS_Learner.py
gp_experiment
BiddingEnvironment.py
GTS_Learner.py
GPTS_Learner.py
Learner.py
No Python interpreter configured for the project
Configure Python interpreter
kernel = C(1.0, (1e-3, 1e3)) * RBF(1.0, (1e-3, 1e3))
self.gp = GaussianProcessRegressor(kernel=kernel, alpha = alpha**2, normalize_y=True, n_restarts_optimizer = 9)
def update_observations(self,arm_idx, reward):
super().update_observations(arm_idx, reward)
self.pulled_arms.append(self.arms[arm_idx])
def update_model(self):
x = np.atleast_2d(self.pulled_arms).T
y = self.collected_rewards
self.gp.fit(x,y)
self.means, self.sigmas = self.gp.predict(np.atleast_2d(self.arms).T, return_std = True)
self.sigmas = np.maximum(self.sigmas,1e-2)
def update(self,pulled_arm, reward):
self.t += 1
self.update_observations(pulled_arm, reward)
self.update_model()
def pull_arm(self):
sampled_values = np.random.normal(self.means,self.sigmas)
return np.argmax(sampled_values)
3: Find
6: TODO
Terminal
Python Console
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (today 18:37)
42:1 CRLF UTF-8 4 spaces


mab [C:\Users\alessandro\Desktop\mab] - ...\gpts_experiment.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab gpts_experiment.py
gpts_experiment
BiddingEnvironment.py
GTS_Learner.py
GPTS_Learner.py
gpts_experiment.py
No Python interpreter configured for the project
Configure Python interpreter
import numpy as np
import matplotlib.pyplot as plt
from BiddingEnvironment import *
from GTS_Learner import *
from GPTS_Learner import *
n_arms = 20
min_bid = 0.0
max_bid = 1.0
bids = np.linspace(min_bid,max_bid, n_arms)
sigma=10
T = 60
n_experiments=100
gts_rewards_per_experiment = []
gpts_rewards_per_experiment = []
for e in range(0,n_experiments):
env = BiddingEnvironment(bids = bids, sigma=sigma)
gts_learner = GTS_Learner(n_arms=n_arms)
gpts_learner = GPTS_Learner(n_arms=n_arms,arms=bids)
for t in range(0,T
1: Project
2: Structure
2: Favorites
Database
SciView
3: Find
6: TODO
Terminal
Python Console
Event Log
Indent expected
25:23 CRLF UTF-8 4 spaces


mab [C:\Users\alessandro\Desktop\mab] - ...\gpts_experiment.py [mab] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mab gpts_experiment.py
gpts_experiment
BiddingEnvironment.py
GTS_Learner.py
GPTS_Learner.py
gpts_experiment.py
No Python interpreter configured for the project
Configure Python interpreter
gts_learner = GTS_Learner(n_arms=n_arms)
gpts_learner = GPTS_Learner(n_arms=n_arms,arms=bids)
for t in range(0,T):
# Gaussian Thompson Sampling
pulled_arm = gts_learner.pull_arm()
reward = env.round(pulled_arm)
gts_learner.update(pulled_arm,reward)
# GP Thompson Sampling
pulled_arm = gpts_learner.pull_arm()
reward = env.round(pulled_arm)
gpts_learner.update(pulled_arm,reward)
gts_rewards_per_experiment.append(gts_learner.collected_rewards)
gpts_rewards_per_experiment.append(gpts_learner.collected_rewards)
opt = np.max(env.means)
plt.figure(0)
plt.ylabel("Regret")
plt.xlabel("t")
plt.plot(np.cumsum(np.mean(opt - gts_rewards_per_experiment,axis=0)), 'r')
plt.plot(np.cumsum(np.mean(opt - gpts_rewards_per_experiment,axis=0)),'g')
plt.legend(["GTS","GPTS"])
plt.show()
1: Project
2: Favorites
7: Structure
Database
SciView
3: Find
6: TODO
Terminal
Python Console
Event Log
PEP 8: blank line at end of file
47:11 CRLF UTF-8 4 spaces