

parameter_sensitivity

March 29, 2021

1 KPI sensitivity towards parameter choices

Danilo Lessa Bernardineli

This is a practical application of the ‘cadCAD_machine_search’ library

- `cadcad_machine_search.tools.sweep_cartesian_product` makes it easy to create large-scale simulations
- `cadcad_machine_search.visualizations.plot_kpi_sensitivity` allows to quickly get insights about what parameters matters the most while keeping track of the pathways about how they interact with the system

1.1 Dependences

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from cadCAD.configuration import Experiment
from cadCAD.configuration.utils import config_sim
from cadCAD.engine import ExecutionMode, ExecutionContext, Executor
```

1.2 Definitions

1.2.1 Initial conditions and parameters

```
[2]: initial_conditions = {
    'prey_population': 100,
    'predator_population': 15
}

# The sweep_cartesian_product makes it easy to create large-scale simulations
from cadcad_machine_search.tools import sweep_cartesian_product

PARAMS_TO_SWEEP = {
    "predator_death_const": np.linspace(0.9, 1.1, 5),
    "prey_death_const": np.linspace(0.02, 0.04, 5),
    "dt": np.linspace(0.04, 0.06, 4)
```

```

    }

sweep_params = sweep_cartesian_product(PARAMS_TO_SWEEP)

params = {
    "prey_birth_rate": [1.0],
    "predator_birth_rate": [0.01],
    **sweep_params
}

simulation_parameters = {
    'N': 5,
    'T': range(200),
    'M': params
}

```

1.2.2 Policies

```

[3]: def p_predator_births(params, step, sL, s):
    dt = params['dt']
    predator_population = s['predator_population']
    prey_population = s['prey_population']
    birth_fraction = params['predator_birth_rate'] + np.random.random() * 0.0002
    births = birth_fraction * prey_population * predator_population * dt
    return {'add_to_predator_population': births}

def p_prey_births(params, step, sL, s):
    dt = params['dt']
    population = s['prey_population']
    birth_fraction = params['prey_birth_rate'] + np.random.random() * 0.1
    births = birth_fraction * population * dt
    return {'add_to_prey_population': births}

def p_predator_deaths(params, step, sL, s):
    dt = params['dt']
    population = s['predator_population']
    death_rate = params['predator_death_const'] + np.random.random() * 0.005
    deaths = death_rate * population * dt
    return {'add_to_predator_population': -1.0 * deaths}

def p_prey_deaths(params, step, sL, s):
    dt = params['dt']
    death_rate = params['prey_death_const'] + np.random.random() * 0.1

```

```

prey_population = s['prey_population']
predator_population = s['predator_population']
deaths = death_rate * prey_population * predator_population * dt
return {'add_to_prepopulation': -1.0 * deaths}

```

1.2.3 State update functions

```

[4]: def s_prepopulation(params, step, sL, s, _input):
    y = 'prey_population'
    x = s['prey_population'] + _input['add_to_prepopulation']
    return (y, x)

def s_predator_population(params, step, sL, s, _input):
    y = 'predator_population'
    x = s['predator_population'] + _input['add_to_predator_population']
    return (y, x)

```

1.2.4 State update blocks

```

[5]: partial_state_update_blocks = [
    {
        'policies': {
            'predator_births': p_predator_births,
            'prey_births': p_prepopulation_births,
            'predator_deaths': p_predator_deaths,
            'prey_deaths': p_prepopulation_deaths,
        },
        'variables': {
            'predator_population': s_prepopulation,
            'prey_population': s_predator_population
        }
    }
]

```

1.2.5 Configuration and Execution

```

[6]: sim_config = config_sim(simulation_parameters)

exp = Experiment()
exp.append_configs(sim_configs=sim_config,
                  initial_state=initial_conditions,
                  partial_state_update_blocks=partial_state_update_blocks)

from cadCAD import configs

```

```

exec_mode = ExecutionMode()
exec_context = ExecutionContext(exec_mode.local_mode)
executor = Executor(exec_context=exec_context, configs=configs)
(records, tensor_field, _) = executor.execute()

```

```

      -----
   ----- / / ----- | / --- \
 / --- / --- / --- / / | | / / / /
/ --- / / / / / / / --- | / / / /
\ --- / \ --- / \ --- / / | / / --- /
by cadCAD

```

Execution Mode: local_proc

Configuration Count: 100

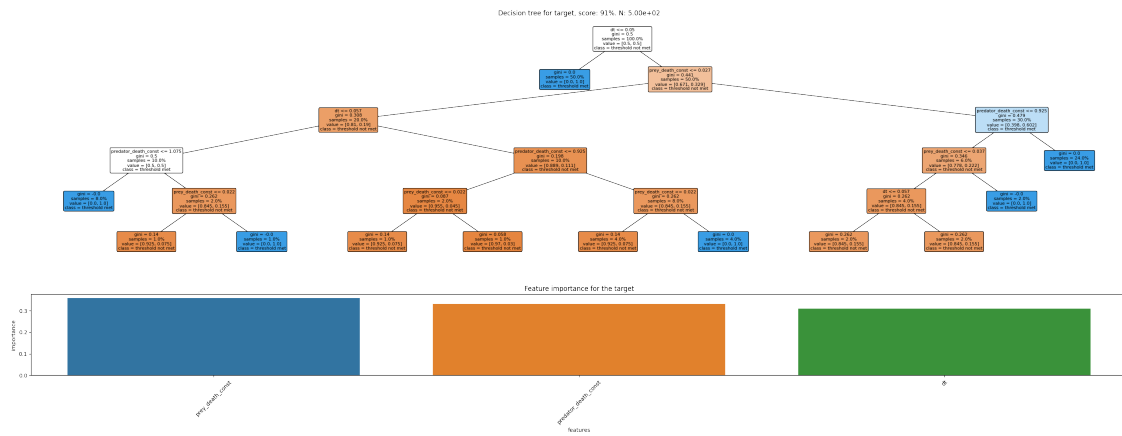
Dimensions of the first simulation: (Timesteps, Params, Runs, Vars) = (200, 5, 5, 2)

Execution Method: local_simulations

SimIDs : [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 12, 12, 12, 12, 12, 13, 13, 13, 13, 13, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 17, 17, 17, 17, 17, 18, 18, 18, 18, 18, 19, 19, 19, 19, 19, 20, 20, 20, 20, 20, 21, 21, 21, 21, 21, 22, 22, 22, 22, 22, 23, 23, 23, 23, 23, 24, 24, 24, 24, 24, 25, 25, 25, 25, 25, 26, 26, 26, 26, 26, 27, 27, 27, 27, 27, 28, 28, 28, 28, 28, 29, 29, 29, 29, 29, 30, 30, 30, 30, 30, 31, 31, 31, 31, 31, 32, 32, 32, 32, 32, 33, 33, 33, 33, 33, 34, 34, 34, 34, 34, 35, 35, 35, 35, 35, 36, 36, 36, 36, 36, 37, 37, 37, 37, 37, 38, 38, 38, 38, 38, 39, 39, 39, 39, 39, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 42, 42, 42, 42, 42, 43, 43, 43, 43, 43, 44, 44, 44, 44, 44, 45, 45, 45, 45, 45, 46, 46, 46, 46, 46, 47, 47, 47, 47, 47, 48, 48, 48, 48, 48, 49, 49, 49, 49, 49, 50, 50, 50, 50, 50, 51, 51, 51, 51, 51, 52, 52, 52, 52, 52, 53, 53, 53, 53, 53, 54, 54, 54, 54, 54, 55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 57, 57, 57, 57, 57, 58, 58, 58, 58, 58, 59, 59, 59, 59, 59, 60, 60, 60, 60, 60, 61, 61, 61, 61, 61, 62, 62, 62, 62, 62, 63, 63, 63, 63, 63, 64, 64, 64, 64, 64, 65, 65, 65, 65, 65, 66, 66, 66, 66, 66, 67, 67, 67, 67, 67, 68, 68, 68, 68, 68, 69, 69, 69, 69, 69, 70, 70, 70, 70, 70, 71, 71, 71, 71, 71, 72, 72, 72, 72, 72, 73, 73, 73, 73, 73, 74, 74, 74, 74, 74, 75, 75, 75, 75, 75, 76, 76, 76, 76, 76, 77, 77, 77, 77, 77, 78, 78, 78, 78, 78, 79, 79, 79, 79, 79, 80, 80, 80, 80, 80, 81, 81, 81, 81, 81, 82, 82, 82, 82, 82, 83, 83, 83, 83, 83, 84, 84, 84, 84, 84, 85, 85, 85, 85, 85, 86, 86, 86, 86, 86, 87, 87, 87, 87, 87, 88, 88, 88, 88, 88, 89, 89, 89, 89, 89, 90, 90, 90, 90, 90, 91, 91, 91, 91, 91, 92, 92, 92, 92, 92, 93, 93, 93, 93, 93, 94, 94, 94, 94, 94, 95, 95, 95, 95, 95, 96, 96, 96, 96, 96, 97, 97, 97, 97, 97, 98, 98, 98, 98, 98, 99, 99, 99, 99, 99]

SubsetIDs: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 12, 12, 12, 12, 12, 13, 13, 13, 13, 13, 14, 14, 14, 14, 14, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 17, 17, 17, 17, 17,

Execution Mode: parallelized
Total execution time: 10.57s



[]: