
Package "crp": Chinese Restaurant Process

Ruben Abbout & Jack Potrykus

Mar 14, 2021

CONTENTS:

1	Indices and tables	3
	Python Module Index	5
	Index	7


```

class chinese_restaurant_process.ChineseRestaurantProcess(alpha)
    Initializes and provides utilities to draw from Chinese Restaurant Process

    alpha
        concentration parameter
        Type float

    tables
        keys are table #s (ints), values are arrays, indicating which n resulted in a customer sitting at that table
        Type dict

    history
        list of all previous states of the Process
        Type list

    n
        number of states simulated, plus 1
        Type int

    animate()
        Animate the progress of the CRT by looping through self.history and producing a bar plot at each interval

    get_table_dict()
        Get a dictionary of pairs from get_table_names() and get_table_sizes()
        Returns keys = table ids (int), values = # of customers at each table (int)
        Return type dict

    get_table_names()
        Get an array of the names of each table
        Returns array of the table names
        Return type np.array

    get_table_sizes()
        Get an array of the number of people at each table
        Returns array of number of people at each table
        Return type np.array

    iter(niter)
        Advance the process niter times.
        Parameters niter(int) – number of draws to make from the Process
        Returns Returns self, so you can call it inline with assignment, i.e. crp = ChineseRestaurant-
            Process(alpha=1).iter(100)
        Return type ChineseRestaurantProcess

    to_pandas()
        Produce a pd.DataFrame object summarizing results of simulation

    visualize()
        Visualize the final state of the Process as a bar plot
        Returns Bar plot as described
        Return type plt.figure

```

class `chinese_restaurant_mixture.ChineseRestaurantMixture` (*alpha*, *param_prior*, *sampler*)

Inherits from `ChineseRestaurantProcess` class, with the extra utility provided simply being a wrapper around the `Process`.

alpha
concentration parameter
Type float

tables
keys are table #s (ints), values are arrays, indicating which n resulted in a customer sitting at that table
Type dict

history
list of all previous states of the `Process`
Type list

n
number of states simulated, plus 1
Type int

param_prior
since the number of tables is theoretically infinite, this is a function which maps table # (int) to a int or float (parameter value). That is, `param_prior(k)` should yield a draw from the prior distribution of θ_k
Type int -> (int or float)

sampler
function which takes a parameter value from `param_prior` and samples from a certain distribution with given parameter value
Type (int or float) -> (int or float)

datapoints
array of the sampled values from the mixture
Type np.ndarray (int or float)

animate (*clear=False*)
Animate the progress of the CRT by looping through `self.history` and producing a bar plot at each interval

reset ()
Reset the `ChineseRestaurantMixture` by clearing `self.datapoints`

sample (*sample_size*, *reset=False*)
Sample `sample_size` points from a Chinese Restaurant Mixture process.

Parameters

- **sample_size** (*int*) – number of points to sample
- **reset** (*bool*) – whether or not to reset `self.datapoints` before proceeding

visualize (*first_n=None*, *clear=False*)
Visualize the final state of the Mixture Process as a kernel density estimate

Parameters **first_n** (*int*) – (optional) visualize the first `first_n` datapoints (to be used in `animate()` method). Default behavior is to use all datapoints

Returns Kernel density estimate of datapoints

Return type plt.figure

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

C

chinese_restaurant_mixture, [1](#)
chinese_restaurant_process, [??](#)

INDEX

A

alpha (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

alpha (*chinese_restaurant_process.ChineseRestaurantProcess*
attribute), 1

animate () (*chinese_restaurant_mixture.ChineseRestaurantMixture*
method), 2

animate () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

C

chinese_restaurant_mixture
module, 1

chinese_restaurant_process
module, 1

ChineseRestaurantMixture (class in *chinese_restaurant_mixture*), 1

ChineseRestaurantProcess (class in *chinese_restaurant_process*), 1

D

datapoints (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

G

get_table_dict () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

get_table_names () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

get_table_sizes () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

H

history (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

history (*chinese_restaurant_process.ChineseRestaurantProcess*
attribute), 1

I

infer () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

M

module (*chinese_restaurant_mixture*, 1
chinese_restaurant_process, 1)

N

n (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

n (*chinese_restaurant_process.ChineseRestaurantProcess*
attribute), 1

P

param_prior (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

R

reset () (*chinese_restaurant_mixture.ChineseRestaurantMixture*
method), 2

S

sample () (*chinese_restaurant_mixture.ChineseRestaurantMixture*
method), 2

sampler (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

T

tables (*chinese_restaurant_mixture.ChineseRestaurantMixture*
attribute), 2

tables (*chinese_restaurant_process.ChineseRestaurantProcess*
attribute), 1

to_pandas () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1

V

visualize () (*chinese_restaurant_mixture.ChineseRestaurantMixture*
method), 2

visualize () (*chinese_restaurant_process.ChineseRestaurantProcess*
method), 1