Machine Teaching

Laura de Oliveira F. Moraes

Aug 01, 2019

CONTENTS:

CHAPTER

ONE

BKT MODULE

class bkt.**BKT**(*hmm_folder='hmm-scalable-818d905234a8600a8e3a65bb0f7aa4cf06423f1a'*, *git_commit='818d905234a8600a8e3a65bb0f7aa4cf06423f1a'*)

Bases: `object`

download()

This implementation is a wrapper around the HMM-scalable tool (http://yudelson.info/hmm-scalable). This function will download and install the original implementation.

Returns self

Return type object

Notes

This is a wrapper around the HMM-scalable tool (http://yudelson.info/hmm-scalable).

fit(*data*, *q_matrix*, *solver='bw'*, *iterations=200*)

Fit BKT model to data. As of July 2019, just default parameters are allowed.

Parameters

- **data** (*{array-like}, shape (n_steps, 3)*) – Sequence of students steps. Each of the three dimensions are: Observed outcome: 0 for fail and 1 for success Student id: student unique identifier Question id: question id in q_matrix
- **q_matrix** (*matrix, shape (n_questions, n_concepts)*) – Each row is a question and each column a concept. If the concept is present in the question, the correspondent cell should contain 1, otherwise, 0.
- **solver** (*string, optional*) – Algorithm used to fit the BKT model. Available solvers are: 'bw': Baum-Welch (default) 'gd': Gradient Descent 'cgd_pr': Conjugate Gradient Descent (Polak-Ribiere) 'cgd_fr': Conjugate Gradient Descent (Fletcher–Reeves) 'cgd_hs': Conjugate Gradient Descent (Hestenes-Stiefel)
- **iterations** (*integer, optional*) – Maximum number of iterations

Returns self

Return type object

Notes

This is a wrapper around the HMM-scalable tool (http://yudelson.info/hmm-scalable)

get_params()

Get fitted params.

Returns **params**

Return type list. List containing the prior, transition and emission values for each skill.

`predict` (*data*, *q_matrix*, *model_file=None*)

Predict student outcomes based on trained model. This is just the hard-assigment (highest probability) of the outcome probabilities.

Parameters **`data`** (*`{array-like}, shape (n_steps, 3)`*) – Sequence of students steps. Each of the three dimensions are: Observed outcome: 0 for fail and 1 for success Student id: student unique identifier Question id: question id in q_matrix

Returns **outcome** – Outcomes for steps in data. Outcome 0 is incorrect and outcome 1 is correct.

Return type {array-like}, shape (n_steps,)

Notes

This is a wrapper around the HMM-scalable tool (http://yudelson.info/hmm-scalable)

`predict_proba` (*data*, *q_matrix*, *model_file=None*)

Predict student outcome probabilities based on trained model.

Parameters **`data`** (*`{array-like}, shape (n_steps, 3)`*) – Sequence of students steps. Each of the three dimensions are: Observed outcome: 0 for fail and 1 for success Student id: student unique identifier Question id: question id in q_matrix

Returns **outcome** – Outcome probabilites for steps in data. Column 0 corresponds to outcome 0 (incorrect) and column 1 to outcome 1 (correct)

Return type {array-like}, shape (n_steps, 2)

Notes

This is a wrapper around the HMM-scalable tool (http://yudelson.info/hmm-scalable)

`set_params` (*params*)

Set model params. No validation is done for this function. Make sure the params variable is in the expected format.

Returns **self**

Return type object

CHAPTER TWO

SIMULATE_STUDENT MODULE

class simulate_student.**SimulateStudent**(*pi*, *A*, *B*)

Bases: `object`

random_MN_draw(*n*, *probs*)

get X random draws from the multinomial distribution whose probability is given by 'probs'

simulate(*nSteps*)

given an HMM = (A, B, pi), simulate state and observation sequences

CHAPTER THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX