

Moodle CAT (R) Adaptive Quiz Plugin

Moodle 4.x Moodle >3.8

Description

This Plugin provides an adaptive quiz. The calculation of the CAT logic is done on a spearate Server and can there be updated via a R-Script without touching the Moodle Plugin. The communication between this Plugin and the R-Server is done via a [REST-API](#).

Setup/Installation

This Plugin can be installed via zip upload on your Moodle installation.

Usage

Testsettings

pers_est

The property **pers_est** has following short values:

Short	Description
MAP	Maximum-A-Posteriori
EAP	Expected-A-Posteriori
WLE	Weighted Likelihood Estimation
ML	Maximum Likelihood Estimation

criteria_adaptive

The property **criteria_adaptive** has following short values:

Short	Description
MI	Maximum Information
MEPV	Minimum Expected Posterior Variance
MEI	Maximum Expected Information
IKL	Integration-based

Itempool

The items in the itempool need to have certain tags to be recognized by the Plugin. The tags are:

Tag	Beispiel	Beschreibung
adpq_1	adpq_1	Marker, dass Frage für adaptiven Itempool markiert ist
enemey_[array with ItemID]	enemy_[ItemID1;ItemID456]	ItemID von Enemy-Items, also Items die nicht zusammen mit dem jeweiligen Item in einem Test vorkommen sollen. Mehrere Werte getrennt durch Semikolon. (<i>Tag weglassen, wenn es keine Enemy-Items gibt.</i>)
ca_[array with strings]	ca_[Leseverstehen;Verständnis;Märchen;MC]	Dieser Tag kann dafür genutzt werden verschiedene Inhaltsbereiche, die durch das Item abgedeckt werden, zu spezifizieren. Er kann zudem auch für andere, für die Itemauswahl relevanten Merkmale genutzt werden (z.B. Itemtyp, kognitives Anforderungsniveau). Mehrere Werte getrennt durch Semikolon. Die Anteile der verschiedenen hier spezifizierten Inhaltsbereiche am Gesamttest können in den Testeinstellungen angegeben werden.
diff_[array with floats]	diff_[1.2] für dichotome Items diff_[1.2;2.34;5.54] für polytome Items	Schwierigkeitsparameter. Kann mehrere Gleitkommawerte haben. Ein Wert bei dichotomen Items oder mehrere Werte bei polytomen Items, getrennt durch Semikolon.

Tag	Beispiel	Beschreibung
disc_[array with numbers]	disc_[1.4] für eindimensionale Tests disc_[1.9;0;1.4] für dreidimensionale Tests	Diskriminationsparameter. Kann mehrere Gleitkommawerte haben. Ein Wert bei eindimensionalen Tests oder mehrere Werte (spezifiziert Ladungsmuster) bei mehrdimensionalen Tests, getrennt durch Semikolon.
cluster_CLUSTERNAME	cluster_A, cluster_C1, cluster_L2	Cluster i.S.d. kontinuierlichen Kalibrierungsstrategie, dem das Item zugehörig ist. Dient der kontinuierlichen Kalibrierungsstrategie. Unterschieden werden adaptive Cluster (cluster_A), Kalibrierungscluster (cluster_C#) und Linkingcluster (cluster_L#)

Extension

Testsettings

Files	Beschreibung
/db/install.xml	SQL-Statements für die Datenbanktabelle hinzufügen, dass Einstellungen in Moodle gespeichert werden
/mod_form.php	UI für Testeinstellungen, ggf. Validierung einfügen
/attempt.php	\$data_for_r_server erweitern
/classes/local/attempt/attempt.php	Url zum R-Server in public function call_r_server ändern
/classes/local/attempt/attempt.php	Tagverarbeitung um an R-Server geschickt zu werden: public static function distribute_used_tags(\$tags, \$itemsArray) erweitern
/lang/de/catadaptivequiz.php /lang/en/catadaptivequiz.php ...	Übersetzungen/Texte für Testeinstellungen hinzufügen

R-Server

Setup/Files

The R-Server is setup via Docker. [rstudio/plumber](#) is used as image for the R-Server.

Dockerfile:

```
FROM rstudio/plumber

RUN apt-get update -qq && apt-get install -y \
    libssl-dev \
    libcurl4-gnutls-dev \
    r-base \
    r-base-core \
    r-base-dev \
    r-base-html \
    r-cran-boot \
    r-cran-class \
    r-cran-cluster \
    r-cran-codetools \
    r-cran-foreign \
    r-cran-kernsmooth \
    r-cran-lattice \
    r-cran-mass \
    r-cran-matrix \
    r-cran-mgcv \
    r-cran-nlme \
    r-cran-nnet \
    r-cran-rpart \
    r-cran-spatial \
    r-cran-survival \
    r-doc-html \
    r-recommended \
    r-cran-dplyr \
    r-cran-rjson \
    r-cran-jsonlite

    # Install plyr
RUN R -e "install.packages('plyr')" > /install_plyr.log

    # Install mirtCAT
RUN R -e "install.packages('mirtCAT')" > /install_mirtCAT.log
RUN R -e "install.packages('dplyr')" > /install_dplyr.log
RUN R -e "install.packages('rjson')" > /install_rjson.log
RUN R -e "install.packages('jsonlite')" > /install_jsonlite.log

    # Expose port 80 for the web server
EXPOSE 80
```

A simple R-Script is used to start the webserver:

```
library(plumber)
print(paste("getwd: ", getwd()))
pr <- plumber::plumb("./main-api.R")
pr$run(host = "0.0.0.0", port = 80)
```

In the main-api.R file the API is defined:

```
# Another endpoint '/doCAT' is defined here, which executes the CAT
# functionality
# and returns a JSON object
#* @post /doCAT
function(data) {
  tryCatch({
    input_object <- data
    # do some checks/preprocessings with the input_object
    # ...

    # Call the test_script function with the input object
    json <- test_script(input_object)
    return (json)
  }, error = function(e) {
    # Handle error here
    message("An error occurred: ", e$message)
    return(NULL)
  })
}
```

Restrictions

The R-Server is not able to access the Moodle Database. Therefore, the R-Server needs to get all necessary information via the REST-API.

R is a single-threaded language. Therefore, the R-Server can only handle one request at a time. This can be a bottleneck if many users are using the Plugin at the same time.

This can be solved by using a load balancer in front of the R-Server or by using a more powerful server or by using a different webserver that can handle multiple requests at the same time.

REST-API

This section describes the REST-API of the R-Server. The R-Server is responsible for the calculation of the CAT-Logic. The Plugin communicates with the R-Server via a REST-API. The R-Server is a separate Server and can be updated without touching the Plugin.

Input Data

```
{
  "courseID": [
    "2"
  ],
  "testID": [
    "2"
  ],
  "itempool": {
    "items": [
      {
        "diff": [
          "0.57"
        ],
        "content_area": [
          "I",
          "9",
          "25",
          "A",
          "KI"
        ],
        "disc": [
          "1.52"
        ],
        "cluster": "A",
        "enemys": [],
        "ID": "I0925AKa",
        "dbID": "67"
      },
      {
        "diff": [
          "-1.25"
        ],
        "content_area": [
          "I",
          "9",
          "25",
          "W",
          "HI"
        ],
        "disc": [
          "1.08"
        ],
        "cluster": "A",
        "enemys": [],
        "ID": "I0925WHa",
        "dbID": "69"
      },
      ...
    ]
  },
  "settings": {
    "maxItems": [
      "12"
    ]
  }
}
```

```

    ],
    "minItems": [
        "1"
    ],
    "minStdError": [
        "0.00000"
    ],
    "criteria_not_adaptive": [
        "random"
    ],
    "ncl_calib": [
        "2"
    ],
    "ncl_link": [
        "2"
    ],
    "ncl_adaptive": [
        "2"
    ],
    "pers_est": [
        "MAP"
    ],
    "criteria_adaptive": [
        "MI"
    ],
    ],
    "exposure": {
        "enabled": [
            1
        ],
        "nitems_exposure": [
            "4"
        ]
    },
    "content_areas": {
        "enabled": [
            "1"
        ],
        "distribution": [
            "catname8:0.5;catname2:0.3;catname3:0.5"
        ]
    }
},
"person": {
    "personID": [
        "2"
    ]
},
"test": {
    "itemID": [
        "U0206VHa",
        "U0206AHa"
    ],
    "item": [ // indices of the items in the itempool
        27,

```

```
        26
      ],
      "scoredResponse": [
        0,
        0
      ],
      "itemtime": [
        21,
        4
      ],
      "timeout": [
        false
      ]
    ]
  }
}
```

Output Data

```
{
  "personID":["2"], // user id
  "terminated":[false], // is quiz terminated
  "theta":[-1.3127], // ability measure
  "SE":[0.3918], // standarderror
  "nextItem":["B0720VHb"] // next question id
}
```