Action Analysis & Machine Learning Techniques

Claus Zinn
German Research Center for Artificial Intelligence (DFKI)

Original Author: Martin Mühlenbrock

Goals of Action Analysis

• Make use of available logs of learner actions
• Generate higher-level information
• Provide information or even support for students
• Provide information for teachers
• Provide tools for educational researchers
• Give feedback to content providers

What Makes ITS Data Analyzable?

• Multiple grain sizes
  – duration and frequency of student sessions
  – individual reading of words or sections
  – performance-related scores (exercises, mcqs, etc.)
• Data must be machine-readable
  – replacing freehand drawing with a limited palette of graphical objects and operations
  – replacing free-form responses with menu selections
• Timing
  – what students see, how long, and how often, e.g.
  – how long they take to read pages, sections, or sentences
  – response time to multiple choice questions to detect guessing
  – time spent on different activities

Related Fields

• Student Modeling
  – Assessing students’ knowledge (learner model) vs. analyzing students’ behavior (learner profile)
• Web Server Log Analysis
  – Focus on user navigation, no pedagogical concern
• Data mining
  – Technology for analyzing large databases, no pedagogical concern

Log Analysis as Data Mining

• Data collection
  – Server-side data collection
  – Client-side data collection
  – Integration of additional data, such as ontological information on learning material and user registration information
Log Analysis as Data Mining

• Data preparation
  - Fixed transformations e.g., timestamp conversions, learning material lookup, and extraction of URL parameters
  - User identification using heuristics if data collection does not provide explicit user identification
  - Session identification e.g., beginning and end of sessions, pauses
  - Flexible transformations e.g., cumulating information from single clicks to summarize session information
  - Data cleaning e.g., removing demo sessions, detect user name changes

• Data mining
  - Association analysis such as analyzing typical navigation paths
  - Sequence analysis e.g., for controlling these typical navigation paths for specific users or user groups over time
  - Cluster analysis e.g., for grouping users according to their behavior and their characteristics
  - Classification analysis i.e., in order to try to describe these clusters with classification rules such as decision trees

• Reporting facilities
  - Access statistics such as hits, page impressions, peak visit times, duration of sessions, average amount of pages seen
  - User statistics such as first time users, returning users, number of sessions per user, average time between user sessions
  - Session statistics ranging from number and duration of sessions up to information of referrers, entry points and exit points
  - More statistics provide information on effectiveness of hints, click through rates, or failure reports among others

• Result deployment
  - Profile generation such as profiles of specific users
  - Teacher reports such as generation of some high-level report
  - Personalization such as providing personalized links or contents to specific users or user groups

Example: ActiveMath

System Architecture
Sample Query (1/3)

select
  hour.datetime as hour, count(*) as hourcount
from eventtext
group by hour

How much user activity is at what time of the day?
Sample Query (2/3)

select page, count(*) as pagecount
from event
where page
order by pagecount desc

What are the most popular pages?

Sample Query (3/3)

select e1.user,
from_unixtime(left(e1.ts, 10)) as logintime,
(e2.ts - e1.ts)/60000 as duration
from event e1
left join event e2
using(session)
where e1.type = 'Logged In'
and e2.type = 'Logged Out'
and e2.ts - e1.ts > 0
order by duration

Sample School Experiment
Visualization of Query Results

The DFKI Student Inspector
- Diplomarbeit of Oliver Scheuer
- Requirements Analysis informed by questionnaire given to e-distance teachers
- Current support for ActiveMath log data
- Support for iClass log data in the working
- High usability, as testified in 2nd evaluation
- High potential for teachers and educational researchers
- Will inform design of corresponding tool for students
  - fostering meta-cognition
  - supporting the self-regulated learner
### Attributes for Further Analysis

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate of successful exercises to all finished exercises</td>
<td>Automatic Exercise Success Rate</td>
</tr>
<tr>
<td>Rate of finished exercises to all started exercises</td>
<td>Automatic Exercise Finished Rate</td>
</tr>
<tr>
<td>Whether the student accessed the learning environment beyond lesson hours, e.g., from home or during free periods</td>
<td>Automatic Worked Off Time</td>
</tr>
<tr>
<td>Whether the student used the dictionary for searching information</td>
<td>Automatic Dictionary Use</td>
</tr>
<tr>
<td>Average number of solving actions in a session</td>
<td>Automatic Average Solving</td>
</tr>
<tr>
<td>Average number of reading actions in a session</td>
<td>Automatic Average Reading</td>
</tr>
<tr>
<td>Number of successful exercises</td>
<td>Automatic Number Successes</td>
</tr>
<tr>
<td>Number of exercises finished</td>
<td>Automatic Number Finished</td>
</tr>
<tr>
<td>Number of exercises started</td>
<td>Automatic Number Started</td>
</tr>
<tr>
<td>Results in the post test done in writing (binned into low, medium, and high for the decision tree learning)</td>
<td>Manual Post Test Result</td>
</tr>
<tr>
<td>Whether the student is handicapped</td>
<td>Manual Integration Pupil</td>
</tr>
<tr>
<td>Male or female</td>
<td>Manual Gender</td>
</tr>
<tr>
<td>Each class has been split into two subgroups, with each being taught by another teacher (not used for the decision tree learning)</td>
<td>Manual Teacher</td>
</tr>
<tr>
<td>Course, each comprised of about 20 students (not used for the decision tree learning)</td>
<td>Manual Class</td>
</tr>
<tr>
<td>User name (not used for the decision tree learning)</td>
<td>Manual User</td>
</tr>
<tr>
<td>Comment</td>
<td>Generation Attribute</td>
</tr>
</tbody>
</table>

### Data Mining / Machine Learning Methods

- **Classification Learning**
  - Given a set of classified examples, find a "way" of classifying unseen examples
- **Association Learning**
  - Find any association between the features/attributes of the example set
- **Clustering**
  - Find groups of examples that belong together
- **Numeric Prediction**
  - Predicted outcome is not a discrete class but a numeric quantity

### Classification Learning

- **Goals**
  - Represent interrelations and dependencies in the data (characterize)
  - Often: Provide explicit and intelligible description
  - Classify new data (prediction)
- **Main ingredients**
  - Training set: Data used for machine learning
  - Classifier: Indicator for class membership
  - Test set: Data used for evaluating the quality of the classifier
  - (Positive/Negative) Example: Element of the training or test set characterized by the (binary) classifier

### Classification Analysis

1. Learn classifier
   - One interesting (hidden) attribute
2. Use classifier
   - Set of observed (logged) attributes

### Classifier: Decision Trees

- **Decision Tree**
  - Play (yes/no)
  - Windy (true/false)
  - Temperature (hot/mild/cold)
  - Outlook (sunny/overcast/rainy)
  - Humidity (high/normal)

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(slides adapted and examples taken from Witten & Frank, 2000)
Probabilities for weather data

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2</td>
<td>3</td>
<td>Hot</td>
<td>2</td>
</tr>
<tr>
<td>Overcast</td>
<td>4</td>
<td>0</td>
<td>Mild</td>
<td>4</td>
</tr>
<tr>
<td>Rainy</td>
<td>3</td>
<td>2</td>
<td>Cold</td>
<td>3</td>
</tr>
</tbody>
</table>

- Sunny: 2/3, Hot: 2/4, False: 6/2, True: 9/5
- Overcast: 4/4, Mild: 4/2, Normal: 8/1, True: 3/2
- Rainy: 3/2, Cold: 3/1

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2/3</td>
<td>3</td>
<td>Hot</td>
<td>2</td>
</tr>
<tr>
<td>Overcast</td>
<td>4/4</td>
<td>0</td>
<td>Mild</td>
<td>4</td>
</tr>
<tr>
<td>Rainy</td>
<td>3/3</td>
<td>2</td>
<td>Cold</td>
<td>3</td>
</tr>
</tbody>
</table>

- Sunny: 2/3, Hot: 2/5, False: 6/2, True: 9/5
- Overcast: 4/5, Mild: 4/2, Normal: 6/1, True: 3/2
- Rainy: 3/2, Cold: 3/1

A new day:

- Outlook: Sunny
- Temperature: High
- Humidity: Normal
- Windy: False
- Play: ?

Example: Decision Trees

- Outlook: sunny, overcast, rainy
- Temperature: hot, windy: true, false
- Humidity: normal

Constructing decision trees

- Normal procedure: top-down in recursive divide-and-conquer fashion
- First: attribute is selected for root node and branch is created for each possible attribute value
- Then: instances are split into subsets (one for each branch extending from the node)
- Finally: procedure is repeated recursively for each branch, using only instances that reach the branch
- Process stops if all instances have the same class

Which attribute to select?

- Which is the best attribute?
- The one which will result in the smallest tree
- Heuristic: choose the attribute that produces the "purest" nodes
- Popular impurity criterion: information gain
- Information gain increases with the average purity of the subsets that an attribute produces
- Strategy: choose attribute that results in greatest information gain
Computing information

- Information is measured in bits
- Given a probability distribution, the info required to predict an event is the distribution’s entropy
- Entropy gives the information required in bits (this can involve fractions of bits)
- Formula for computing the entropy:
  \[ \text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_n \log_2 p_n \]

Example: attribute “Outlook”

- “Outlook” = “Sunny”:
  \[ \text{info}(2.3) = \text{entropy}(2/3, 1/3) = -2/3 \log_2(2/3) - 1/3 \log_2(1/3) = 0.971 \text{ bits} \]
- “Outlook” = “Overcast”:
  \[ \text{info}(4.0) = \text{entropy}(1.0) = -1 \log_2(1) = 0 \text{ bits} \]
- “Outlook” = “Rainy”:
  \[ \text{info}(3.2) = \text{entropy}(3/5, 2/5) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.971 \text{ bits} \]

- Expected information for attribute:
  \[ \text{info}(3.2, 4.0, 3.2) = (3/5) \times 0.971 + (4/14) \times 0.971 = 0.693 \text{ bits} \]

Computing the information gain

- Information gain: information before splitting – information after splitting
  \[ \text{gain}(\text{Outlook}) = \text{info}(9.5) - \text{info}(3.2, 4.0, 3.2) = 0.540 - 0.693 = 0.247 \text{ bits} \]
- Information gain for attributes from weather data:
  \[ \text{gain}(\text{Outlook}) = 0.247 \text{ bits} \]
  \[ \text{gain}(\text{Temperature}) = 0.029 \text{ bits} \]
  \[ \text{gain}(\text{Humidity}) = 0.152 \text{ bits} \]
  \[ \text{gain}(\text{Windy}) = 0.048 \text{ bits} \]

Continuing to split

- \text{gain}(\text{Temperature}) = 0.571 \text{ bits} \]
  \[ \text{gain}(\text{Humidity}) = 0.971 \text{ bits} \]
  \[ \text{gain}(\text{Windy}) = 0.020 \text{ bits} \]

The final decision tree

- Note: not all leaves need to be pure; sometimes identical instances have different classes
- Splitting stops when data can’t be split any further

Attributes for Further Analysis

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Operation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Manual</td>
<td>The user's role for the decision tree learning.</td>
</tr>
<tr>
<td>Class</td>
<td>Manual</td>
<td>Order: initial class of about 20 students, last used for the decision tree training.</td>
</tr>
<tr>
<td>Teacher</td>
<td>Manual</td>
<td>Initial class of about 20 students, last used for the decision tree training.</td>
</tr>
<tr>
<td>Status</td>
<td>Automatic</td>
<td>Whether the student is handicapped.</td>
</tr>
<tr>
<td>Test result</td>
<td>Manual</td>
<td>Status: in the first class, between 15 and 20, in the second, between 20 and 30, and high for the decision tree training.</td>
</tr>
<tr>
<td>Sex</td>
<td>Automatic</td>
<td>Whether the student is male or female.</td>
</tr>
<tr>
<td>H. Student</td>
<td>Automatic</td>
<td>Whether the student is a habitat.</td>
</tr>
<tr>
<td>No. receipt</td>
<td>Automatic</td>
<td>Number of incorrect responses.</td>
</tr>
<tr>
<td>Avg. reading</td>
<td>Automatic</td>
<td>Average number of incorrect statements in a session.</td>
</tr>
<tr>
<td>Avg. writing</td>
<td>Automatic</td>
<td>Average number of incorrect statements in a session.</td>
</tr>
<tr>
<td>Avg. math</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. science</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. history</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. dom</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. foreign</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. science</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. math</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. reading</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. writing</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. history</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. dom</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. foreign</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. science</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. math</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. reading</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. writing</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. history</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. dom</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. foreign</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
<tr>
<td>Avg. science</td>
<td>Automatic</td>
<td>Whether the student was for the decision tree training.</td>
</tr>
</tbody>
</table>
### Decision Tree for Post Test Result

- **ex_success_rate = 0.76 & ex_finished_rate = 0.93 & avg_reading = 56**
- **post test result = ?**

### Confusion Matrix

- **Predicted class**
  - Low
  - Medium
  - High

- **Actual class**
  - Training set
  - Test set

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Confusion Matrix (cont’d)

- **Accuracy (AC)** is proportion of total number of predictions that were correct:
  \[
  AC = \frac{a + d}{a + b + c + d}
  \]

- **Recall or true positive rate (TP)** is proportion of positive cases correctly identified:
  \[
  TP = \frac{d}{c + d}
  \]

- **False positive rate (FP)** is proportion of negatives cases incorrectly classified as positive:
  \[
  FP = \frac{b}{a + b}
  \]

- **True negative rate (TN)** is proportion of negatives cases classified correctly:
  \[
  TN = \frac{a}{a + b}
  \]

- **False negative rate (FN)** is proportion of positives cases incorrectly classified as negative:
  \[
  FN = \frac{c}{c + d}
  \]

- **Precision (P)** is proportion of the predicted positive cases that were correct:
  \[
  P = \frac{d}{b + d}
  \]

### Bayesian Approach

(Arroyo, Murray, & Woolf, 2004)

### Conditional Probability Table

<table>
<thead>
<tr>
<th>Serious Attitude</th>
<th>Trained</th>
<th>Learned</th>
<th>Time per Problem</th>
<th>Cases</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Low</td>
<td>1</td>
<td>0.51</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>High</td>
<td>2</td>
<td>0.67</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>High</td>
<td>0</td>
<td>0.24</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>High</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>Low</td>
<td>Low</td>
<td>24</td>
<td>0.43</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>High</td>
<td>Low</td>
<td>29</td>
<td>0.53</td>
</tr>
</tbody>
</table>

### Sample Inference

<table>
<thead>
<tr>
<th>Observed “test” nodes</th>
<th>Observed Value</th>
<th>Inferred P📺E📺 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning factor</td>
<td>High</td>
<td>Learning factor? 0.352</td>
</tr>
<tr>
<td>Help per test grade</td>
<td>Low</td>
<td>Help per test grade? 0.0057</td>
</tr>
<tr>
<td>Std. Items per course</td>
<td>High</td>
<td>Std. Items per course? 0.5160</td>
</tr>
<tr>
<td>Time by problem</td>
<td>High</td>
<td>Time by problem? 0.0254</td>
</tr>
<tr>
<td>Other Approaches</td>
<td>High</td>
<td>Other Approaches? 0.7072</td>
</tr>
<tr>
<td>Get Over With</td>
<td>High</td>
<td>Get Over With? 0.3184</td>
</tr>
</tbody>
</table>
Current & Further Work

• ActiveMath group works on
  – ActiveMath usage to inform teachers, learners and educational researchers
  – Student Inspector targets teachers
  – using action analysis to detect learning styles
  – instead of complex questionnaires to be filled out by learners,
  – use interaction patterns to detect learning style automatically
    – Active vs. Reflective Learners
    – Sensing vs. Intuitive Learners
    – Visual vs. Verbal Learners
    – Sequential vs. Global Learners
  – exploit learning styles to personalise selection and presentation
    of learning material

Interesting DFKI Project: ARGUNAUT

• Educational Data Mining - the ARGUNAUT Project
  – Use of Machine Learning Techniques to learn collaborative /
    argumentative behaviors of students
  – Use the resulting classifiers to support the moderator of an online
    collaborative discussion
  – Looking for an enthusiastic student / HIWI to work
  on this project
    – Learn to use modern Machine Learning tools
    – Integrating data and text mining
    – Work with a team of learning scientists and PhD students
    – Academic publications!
  – Talk to me after the lecture or send email to
    bmclaren@dfki.de

Interesting DFKI Project: iClass

• Learner Profiling within the iClass Project
  – Use of Machine Learning Techniques to detect user’s preferences, needs,
    cognitive styles, e.g.,
    – deductive – intuitive
    – visual – auditory – verbal – combined
    – individual - social
  – Use the resulting classifiers to support meta-cognitive activities (with
    learner and teacher)
  – Looking for an enthusiastic student / HIWI to work on this project
    • Extend existing Java application (the Student Inspector) to cater for new
      incoming requirements
    – Learn to use modern Machine Learning tools (Weka)
    – Work with a team of learning scientists and PhD students
    – Academic publications!
  – Talk to me after the lecture or send email to
    zinn@dfki.de