What Do Students Know?

Titus Winters Tom Payne

University of California, Riverside

ICER Conference, 2005

Thoughts

- Quantitative Assessment and Continuous Improvement
  - Program assessment is hard – have you seen a good answer?
- Data mining is a vibrant field
  - But it requires more input data than most courses produce

Thoughts

- Quantitative Assessment and Continuous Improvement
  - Program assessment is hard – have you seen a good answer?
- Data mining is a vibrant field
  - But it requires more input data than most courses produce

Thoughts

- Quantitative Assessment and Continuous Improvement
  - Program assessment is hard – have you seen a good answer?
- Data mining is a vibrant field
  - But it requires more input data than most courses produce

Thoughts

- Quantitative Assessment and Continuous Improvement
  - Program assessment is hard – have you seen a good answer?
- Data mining is a vibrant field
  - But it requires more input data than most courses produce

UCR's Plan

Develop a process to

- Enable easy collection of detailed score data
- Require minimum input from instructors
- Provide numeric assessment results
- Make the process easy to use

UCR's Plan

Develop a process to

- Enable easy collection of detailed score data
- Require minimum input from instructors
- Provide numeric assessment results
- Make the process easy to use

UCR's Plan

Develop a process to

- Enable easy collection of detailed score data
- Require minimum input from instructors
- Provide numeric assessment results
- Make the process easy to use

UCR's Plan

Develop a process to

- Enable easy collection of detailed score data
- Require minimum input from instructors
- Provide numeric assessment results
- Make the process easy to use

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – "program outcomes"
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – “program outcomes”
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – "program outcomes"
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – "program outcomes"
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – "program outcomes"
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – “program outcomes”
  - Course matrix C (topics $\times$ outcomes)

Process Overview

- Courses taught as normal, scores recorded at the question level
  - Score matrix S (students $\times$ items)
- Course topics / objectives identified
- Scores analyzed or annotated by instructor for question$\longrightarrow$ topic relevance
  - Relevance matrix R (items $\times$ topics)
- Topics related to curriculum-level objectives – “program outcomes”
  - Course matrix C (topics $\times$ outcomes)

Process Overview

$$S \times R \times C = O$$

- O is students $\times$ outcomes
- Average columns of O for numeric outputs
- Course matrix C done once
- S requires lots of data
- R is tedious, but *may* be automated

Data Gathering

- Two major tools, Agar and HOMR
- Score data stored in spreadsheets, not SQL
  - Minimal format requirements
  - Easy to adopt
  - Gnumeric/ssconvert very useful for automation
  - Integrating with Moodle

Data Gathering

- Two major tools, Agar and HOMR
- Score data stored in spreadsheets, not SQL
  - Minimal format requirements
  - Easy to adopt
  - Gnumeric/ssconvert very useful for automation
  - Integrating with Moodle

Data Gathering

- Two major tools, Agar and HOMR
- Score data stored in spreadsheets, not SQL
  - Minimal format requirements
  - Easy to adopt
  - Gnumeric/ssconvert very useful for automation
  - Integrating with Moodle

Agar

Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation

Agar

Not just another Automated Grader

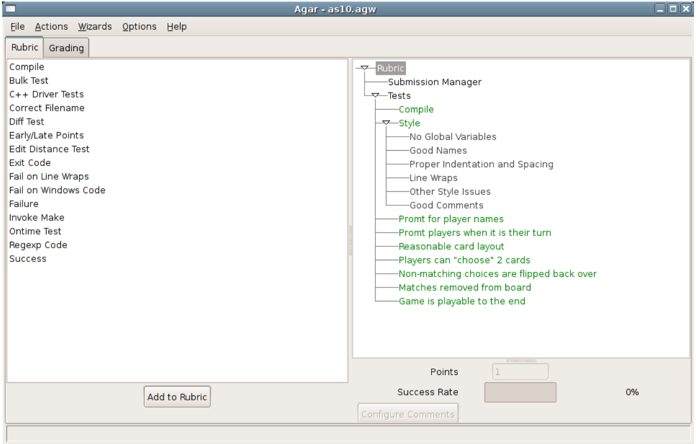
- CAA - Computer *Assisted* Assessment
- Rubric creation

Agar

Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation

Rubric


Agar - as10.agw
File Actions Wizards Options Help
Rubric
Grading
Compile
Bulk Test
C++ Driver Tests
Correct Filename
Diff Test
Early/Late Points
Edit Distance Test
Exit Code
Fail on Line Wraps
Fail on Windows Code
Failure
Invoke Make
Ontime Test
Regexp Code
Success
Add to Rubric
Rubric
Submission Manager
Tests
Compile
Style
No Global Variables
Good Names
Proper Indentation and Spacing
Line Wraps
Other Style Issues
Good Comments
Promt for player names
Promt players when it is their turn
Reasonable card layout
Players can "choose" 2 cards
Non-matching choices are flipped back over
Matches removed from board
Game is playable to the end
Points
1
Success Rate
0%
Configure Comments

Agar

Not just another Automated Grader

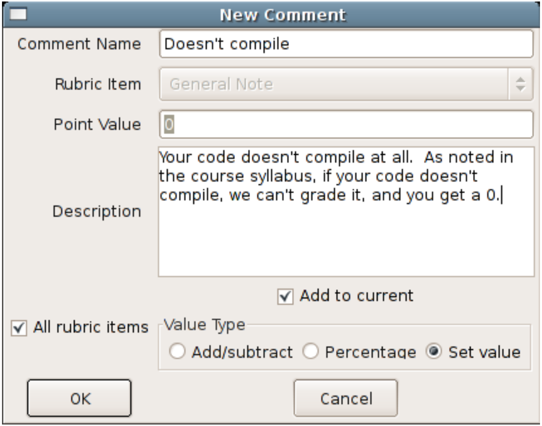
- CAA - Computer *Assisted* Assessment
- Rubric creation
- Write-once comments

Agar

Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation
- Write-once comments
- Easy viewing of submissions
- Annotation
- Automatic mailback of results and feedback

Comments


New Comment
Comment Name
Doesn't compile
Rubric Item
General Note
Point Value
0
Description
Your code doesn't compile at all. As noted in the course syllabus, if your code doesn't compile, we can't grade it, and you get a 0.
Add to current
All rubric items
Value Type
Add/subtract
Percentage
Set value
OK
Cancel

Agar

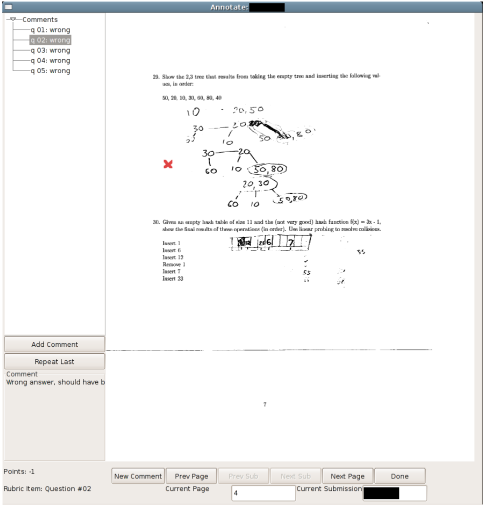
Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation
- Write-once comments
- Easy viewing of submissions
- Annotation

Agar

Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation
- Write-once comments
- Easy viewing of submissions
- Annotation


Annotate: [redacted]
Comments
q 01: wrong
q 02: wrong
q 03: wrong
q 04: wrong
q 05: wrong
Add Comment
Repeat Last
Comment
Wrong answer, should have b
29. Show the 2,3 tree that results from taking the empty tree and inserting the following values, in order:
50, 20, 10, 30, 60, 80, 40
30. Given an empty hash table of size 11 and the (not very good) hash function f(x) = 3x - 1, show the final results of these operations (in order). Use linear probing to resolve collisions.
Insert 1
Insert 6
Insert 12
Remove 1
Insert 7
Insert 23
7
Points: -1
Rubric Item: Question #02
New Comment
Prev Page
Prev Sub
Next Sub
Next Page
Done
Current Page
4
Current Submission [redacted]

Agar

Not just another Automated Grader

- CAA - Computer *Assisted* Assessment
- Rubric creation
- Write-once comments
- Easy viewing of submissions
- Annotation
- Automatic mailback of results and feedback

Remove redundant work, don't replace the human.

Support Existing Paradigms

- Annotation interface for red-pen grading.
- Merge features for breaking up workload among multiple graders

Support Existing Paradigms

- Annotation interface for red-pen grading.
- Merge features for breaking up workload among multiple graders

HOMR

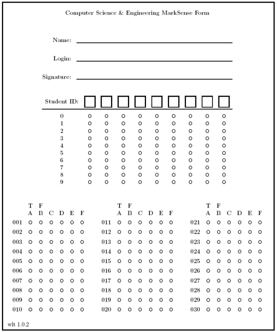
Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces "0" or "1" to stdout for each bubble

HOMR

Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces "0" or "1" to stdout for each bubble


Computer Science & Engineering MarkSense Form

Name:

Login:

Signature:

Student ID:

0 1 2 3 4 5 6 7 8 9

	T A	F B	C	D	E	F
001						
002						
003						
004						
005						
006						
007						
008						
009						
010						

	T A	F B	C	D	E	F
011						
012						
013						
014						
015						
016						
017						
018						
019						
020						

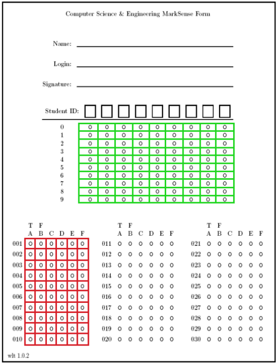
	T A	F B	C	D	E	F
021						
022						
023						
024						
025						
026						
027						
028						
029						
030						

wlt 1.0.2

HOMR

Homebrew Optical Mark Recognition

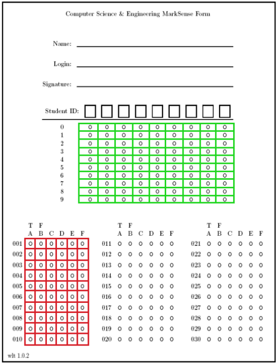
- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces “0” or “1” to stdout for each bubble



HOMR

Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces "0" or "1" to stdout for each bubble



Computer Science & Engineering MarkSense Form

Name:

Login:

Signature:

Student ID:

0
1
2
3
4
5
6
7
8
9

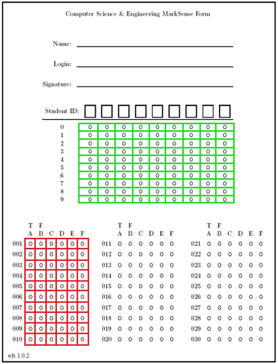
T F A B C D E F		T F A B C D E F		T F A B C D E F
001		011		021
002		012		022
003		013		023
004		014		024
005		015		025
006		016		026
007		017		027
008		018		028
009		019		029
010		020		030

wlt 1.0.2

HOMR

Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces “0” or “1” to stdout for each bubble


Computer Science & Engineering MarkSense Form

Name:

Login:

Signature:

Student ID:

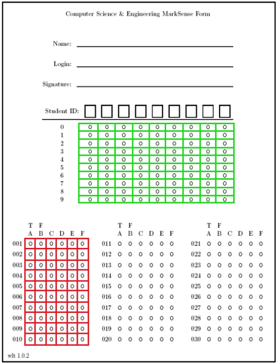
	T A	F B	C	D	E	F
001	○	○	○	○	○	○
002	○	○	○	○	○	○
003	○	○	○	○	○	○
004	○	○	○	○	○	○
005	○	○	○	○	○	○
006	○	○	○	○	○	○
007	○	○	○	○	○	○
008	○	○	○	○	○	○
009	○	○	○	○	○	○
010	○	○	○	○	○	○
011	○	○	○	○	○	○
012	○	○	○	○	○	○
013	○	○	○	○	○	○
014	○	○	○	○	○	○
015	○	○	○	○	○	○
016	○	○	○	○	○	○
017	○	○	○	○	○	○
018	○	○	○	○	○	○
019	○	○	○	○	○	○
020	○	○	○	○	○	○
021	○	○	○	○	○	○
022	○	○	○	○	○	○
023	○	○	○	○	○	○
024	○	○	○	○	○	○
025	○	○	○	○	○	○
026	○	○	○	○	○	○
027	○	○	○	○	○	○
028	○	○	○	○	○	○
029	○	○	○	○	○	○
030	○	○	○	○	○	○

wlt 1.0.2

HOMR

Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces "0" or "1" to stdout for each bubble


Computer Science & Engineering MarkSense Form

Name:

Login:

Signature:

Student ID:

0 1 2 3 4 5 6 7 8 9

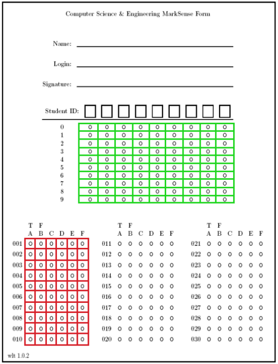
	T A	F B	C	D	E	F
001						
002						
003						
004						
005						
006						
007						
008						
009						
010						
011						
012						
013						
014						
015						
016						
017						
018						
019						
020						
021						
022						
023						
024						
025						
026						
027						
028						
029						
030						

wlt 1.0.2

HOMR

Homebrew Optical Mark Recognition

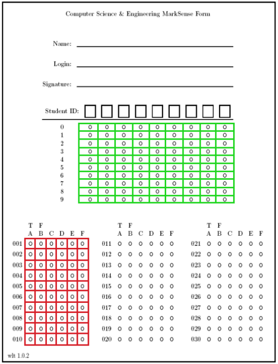
- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces "0" or "1" to stdout for each bubble



HOMR

Homebrew Optical Mark Recognition

- C++ program reads in
  - Page configuration file
  - Number of bubbles to extract
  - Image filename
- Computer vision to normalize page
- AdaBoost classifier to classify each bubble
- Produces “0” or “1” to stdout for each bubble



HOMR Features

- Use from any language that can make shell calls
- Highly accurate: 99.99%
- Tolerant: crossed out shapes are “0”

HOMR Features

- Use from any language that can make shell calls
- Highly accurate: 99.99%
- Tolerant: crossed out shapes are “0”

HOMR Features

- Use from any language that can make shell calls
- Highly accurate: 99.99%
  Compare with US Census: 99.6%
- Tolerant: crossed out shapes are "0"

HOMR Features

- Use from any language that can make shell calls
- Highly accurate: 99.99%
  Compare with US Census: 99.6%
- Tolerant: crossed out shapes are “0”

Data Analysis

Two major techniques

- Evaluate individual items
- Determine which items correspond to the same topic.

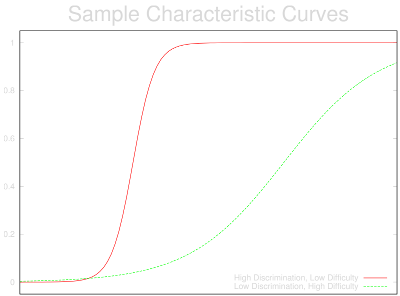
Data Analysis

Two major techniques

- Evaluate individual items
- Determine which items correspond to the same topic.

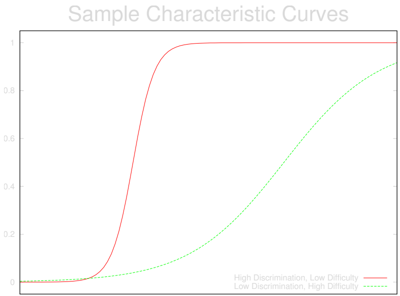
Item Evaluation

- Based on IRT
- Identify difficulty β / discrimination α
- Given β, α is percent correct (below is wrong, above is right).
- β is the split that maximizes α



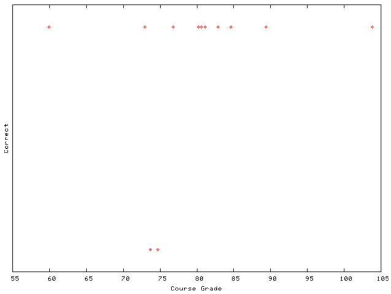
Item Evaluation

- Based on IRT
- Identify difficulty β / discrimination α
- Given β, α is percent correct (below is wrong, above is right).
- β is the split that maximizes α



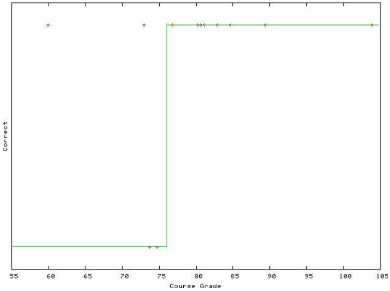
Item Evaluation

- Based on IRT
- Identify difficulty β / discrimination α
- Given β, α is percent correct (below is wrong, above is right).
- β is the split that maximizes α



Item Evaluation

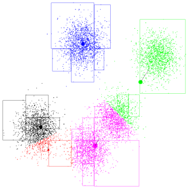
- Based on IRT
- Identify difficulty β / discrimination α
- Given β, α is percent correct (below is wrong, above is right).
- β is the split that maximizes α



Topic Identification

Unsupervised or semi-supervised learning provides many possible techniques:

- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF



Topic Identification

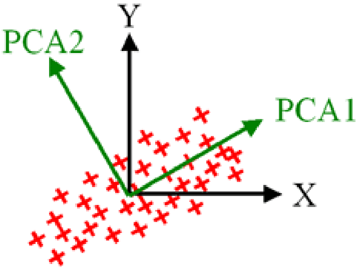
Unsupervised or semi-supervised learning provides many possible techniques:

- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF

Topic Identification

Unsupervised or semi-supervised learning provides many possible techniques:

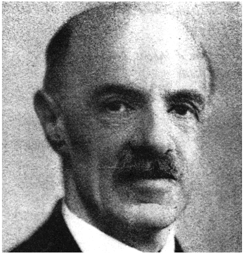
- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF



Topic Identification

Unsupervised or semi-supervised learning provides many possible techniques:

- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF



Charles Spearman

Topic Identification

Unsupervised or semi-supervised learning provides many possible techniques:

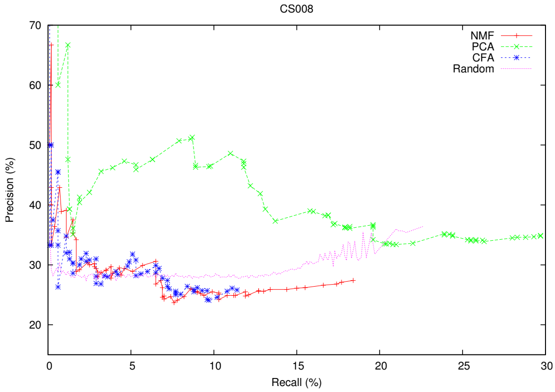
- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF

Topic Identification

Unsupervised or semi-supervised learning provides many possible techniques:

- Clustering
- Hierarchical clustering
- PCA
- CFA
- NMF

Results


CS008
NMF
PCA
CFA
Random
Precision (%)
Recall (%)
70
60
50
40
30
20
0
5
10
15
20
25
30

Conclusions

- Still room for instructional tool development
- Some analysis is easy
- Some analysis is hard
- Score data is **noisy**
- Must be mindful of concepts from educational statistics

Conclusions

- Still room for instructional tool development
- Some analysis is easy
- Some analysis is hard
- Score data is **noisy**
- Must be mindful of concepts from educational statistics

Conclusions

- Still room for instructional tool development
- Some analysis is easy
- Some analysis is hard
- Score data is **noisy**
- Must be mindful of concepts from educational statistics

Conclusions

- Still room for instructional tool development
- Some analysis is easy
- Some analysis is hard
- Score data is **noisy**
- Must be mindful of concepts from educational statistics

Conclusions

- Still room for instructional tool development
- Some analysis is easy
- Some analysis is hard
- Score data is **noisy**
- Must be mindful of concepts from educational statistics

Questions?