CrowdGrader

A Tool for CrowdSourcing the Evaluation of Homework Assignments

Luca de Alfaro		Michael Shavlovsky

UC Santa Cruz

SIGCSE 2014

How did we get into this?

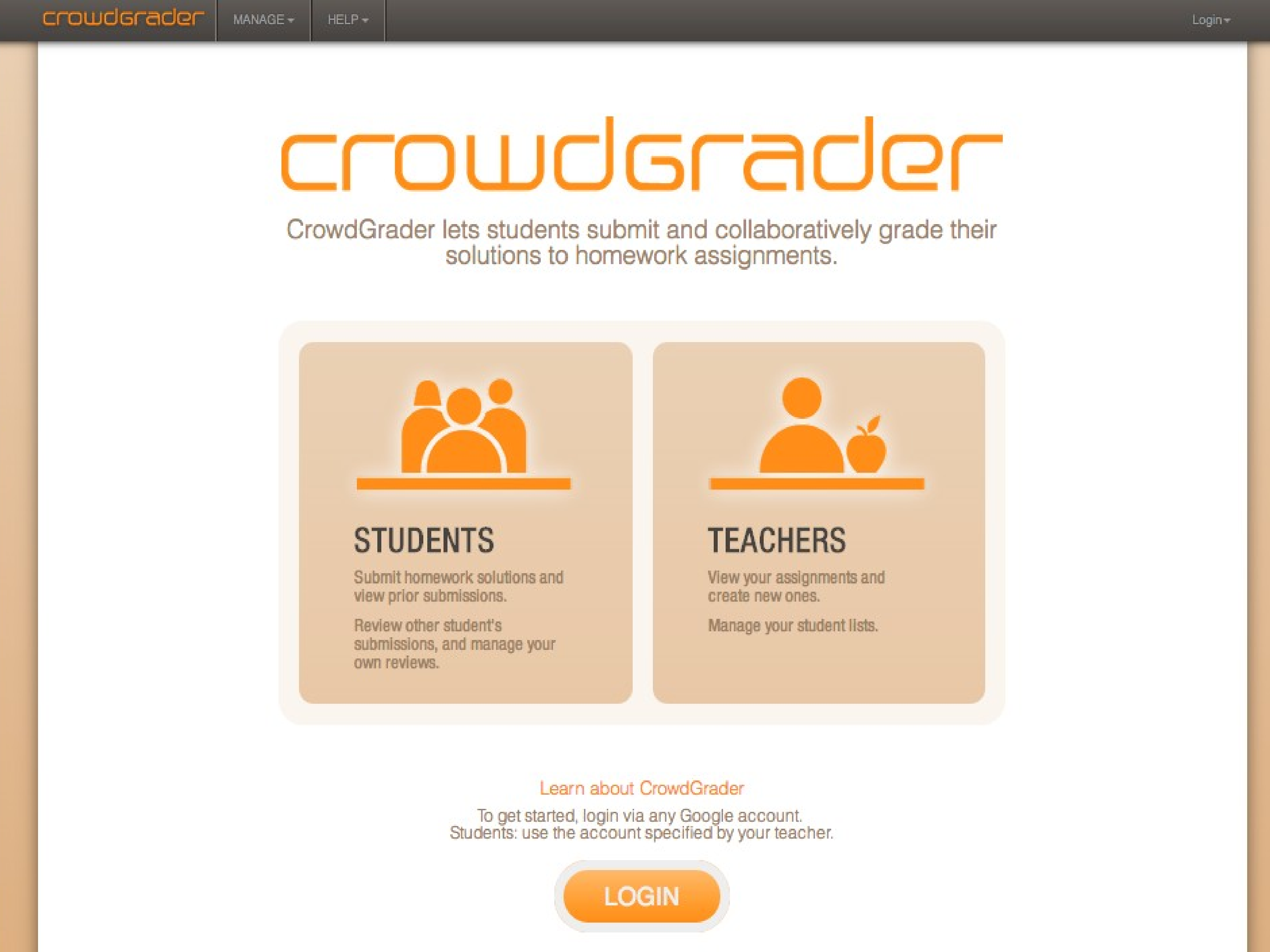
I started a class on Android development

- Very popular, 80+ students
- Little TA support (1 TA, which does not know Android)
- Simply loading an Android app to grade it takes minutes

I am supposed to know about CrowdSourcing

- I had often worked on reputation systems, crowdsourcing, related topics.
- I am supposed to be a good developer!

Why not try to help myself?



crowdgrader

CrowdGrader lets students submit and collaboratively grade their solutions to homework assignments.

STUDENTS

Submit homework solutions and view prior submissions.

Review other student's submissions, and manage your own reviews.

TEACHERS

View your assignments and create new ones.

Manage your student lists.

Learn about CrowdGrader

To get started, login via any Google account.
Students: use the account specified by your teacher.

CrowdGrader

1. **Submission:** students submit their solutions (can submit in groups).
2. **Review:** Students are assigned submissions to review: for each submission, they enter a review, and a **grade/rank**.
3. **Crowd-grades:** CrowdGrader computes a **crowd-grade** that depends both on the quality of their submission, and on the quality of their reviewing work.
4. **Feedback:** students get feedback, and instructors can read all reviews.

CrowdGrader

Design Issues

- **Review assignment**
- **Grade assignment:** how to compute reliable grades
- **Incentive design**: how to motivate students to do good reviews?

How well does it work?

- **Participation in reviews**
- **Amount and quality of feedback**

Review Assignment

One at a time: each time students are done with a review, they get the next submission to review.

This ensures submissions receive a uniform number of reviews.

- CG implements a predictive algorithm that estimates the probability that reviews are completed.
- Opens the way to many experiments on optimally assigning submissions to reviewers.

[Also PG, from NCSU, does dynamic review assignment]

Evaluate via rank, or grades?

We initially asked students to rank the submissions they were reviewing in order of qualiy.

- Ranking requires only a relative, rather than an absolute judgement.
- Simpler.
- Many methods for online and offline rank aggregation.

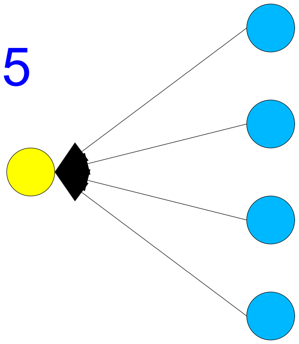
Ranking did not work well

Ranking was skipped 28% of the time!

- **Uneasiness** about ranking peers
- Considered a **coarse instrument**: students complained about having to arbitrarily rank submissions they considered “equivalent”
- **Lack of trust** that this would lead to an accurate ranking.

We moved to grades.

Optimal grade aggregation



Users

Grade

Average: 6.625

Item

6.0

5.0

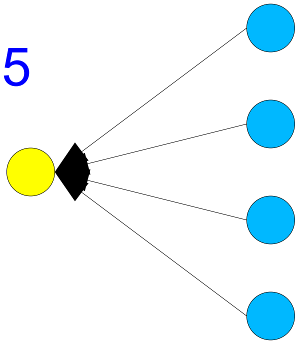
8.0

7.5

Optimal grade aggregation

Average: 6.625

Item



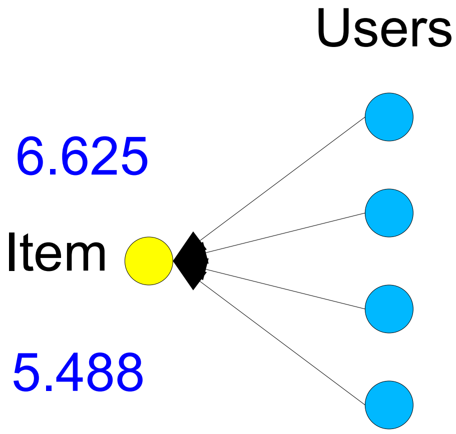
Users

Grade	Variance
6.0	1.0
5.0	0.2
8.0	2.0
7.5	3.0

Optimal grade aggregation

Average: 6.625

MVLE: 5.488



Grade	Variance
6.0	1.0
5.0	0.2
8.0	2.0
7.5	3.0

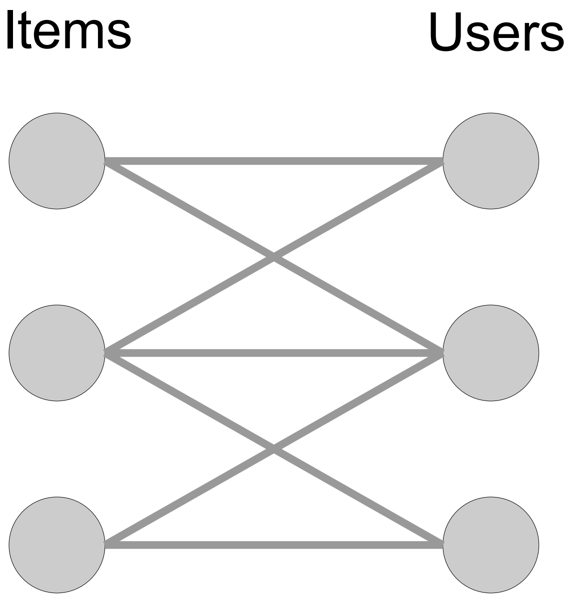
Minimum variance linear estimator:

Let X_1 , ..., X_n be uncorrelated random variables with mean x and variances v_1 , ..., v_n . The minimum-variance linear estimator of x can be obtained by:

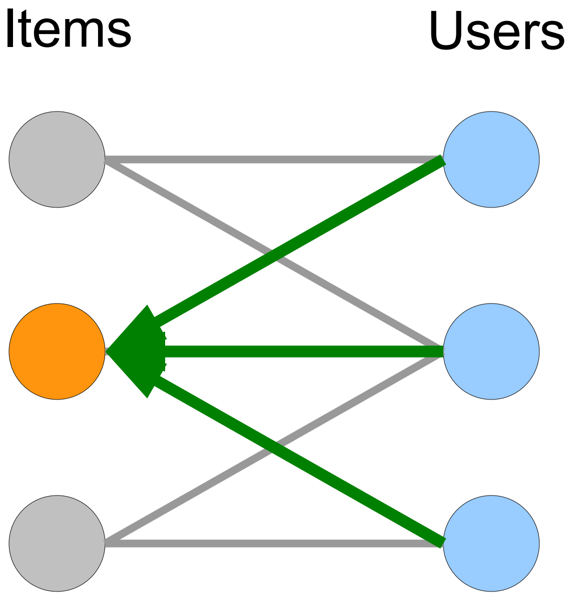
$$\frac{\sum_{i=1}^{n} X_i / v_i}{\sum_{i=1}^{n} 1 / v_i}$$

$$\text{Variance} = \left(\sum_{i=1}^{n} \frac{1}{v_i} \right)^{-1}$$

The Vancouver Algorithm

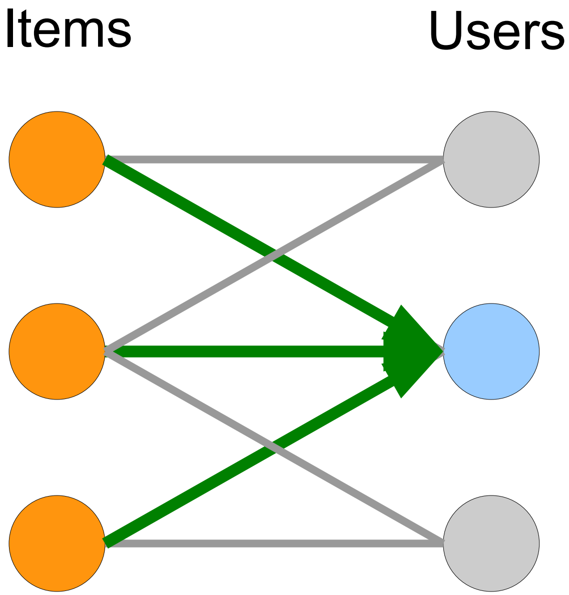


The Vancouver Algorithm



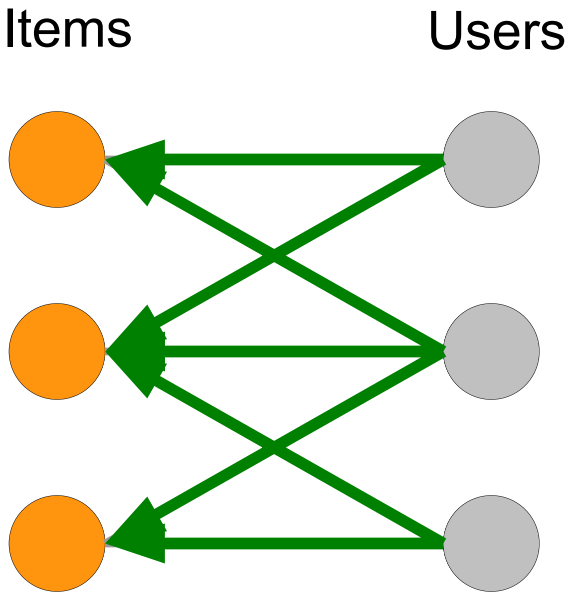
- Users send to items their grade, and their esimated variance.
- We compute grade and variance for the items.

The Vancouver Algorithm



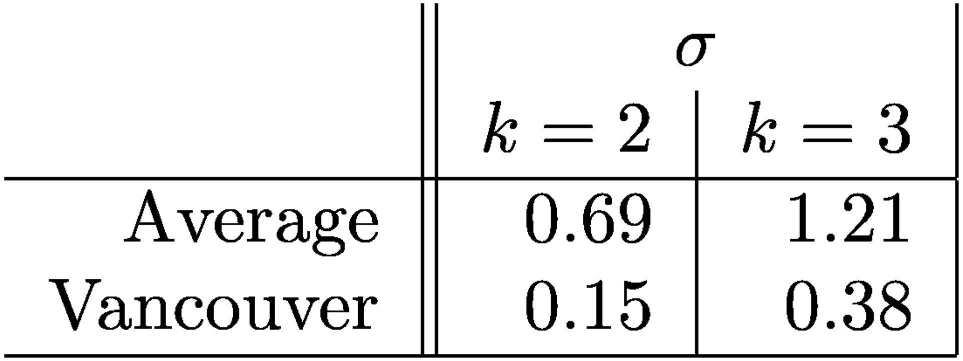
- Items send to users their consensus grade, and variance.
- Users update their variance.

The Vancouver Algorithm



Iterate, until we have consensus grades and variances for all items.

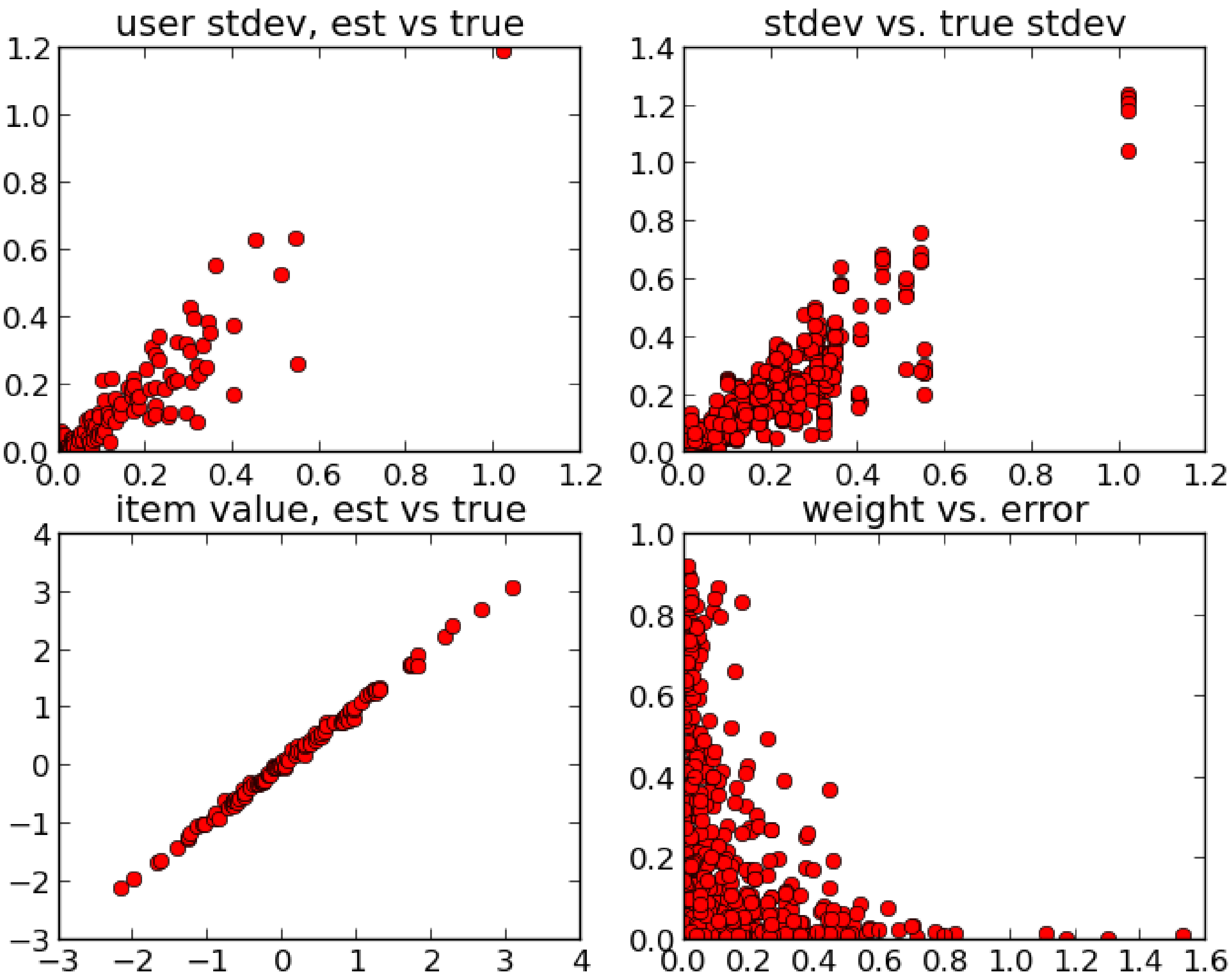
Vancouver: Performance on synthetic data



	σ	
	$k = 2$	$k = 3$
Average	0.69	1.21
Vancouver	0.15	0.38

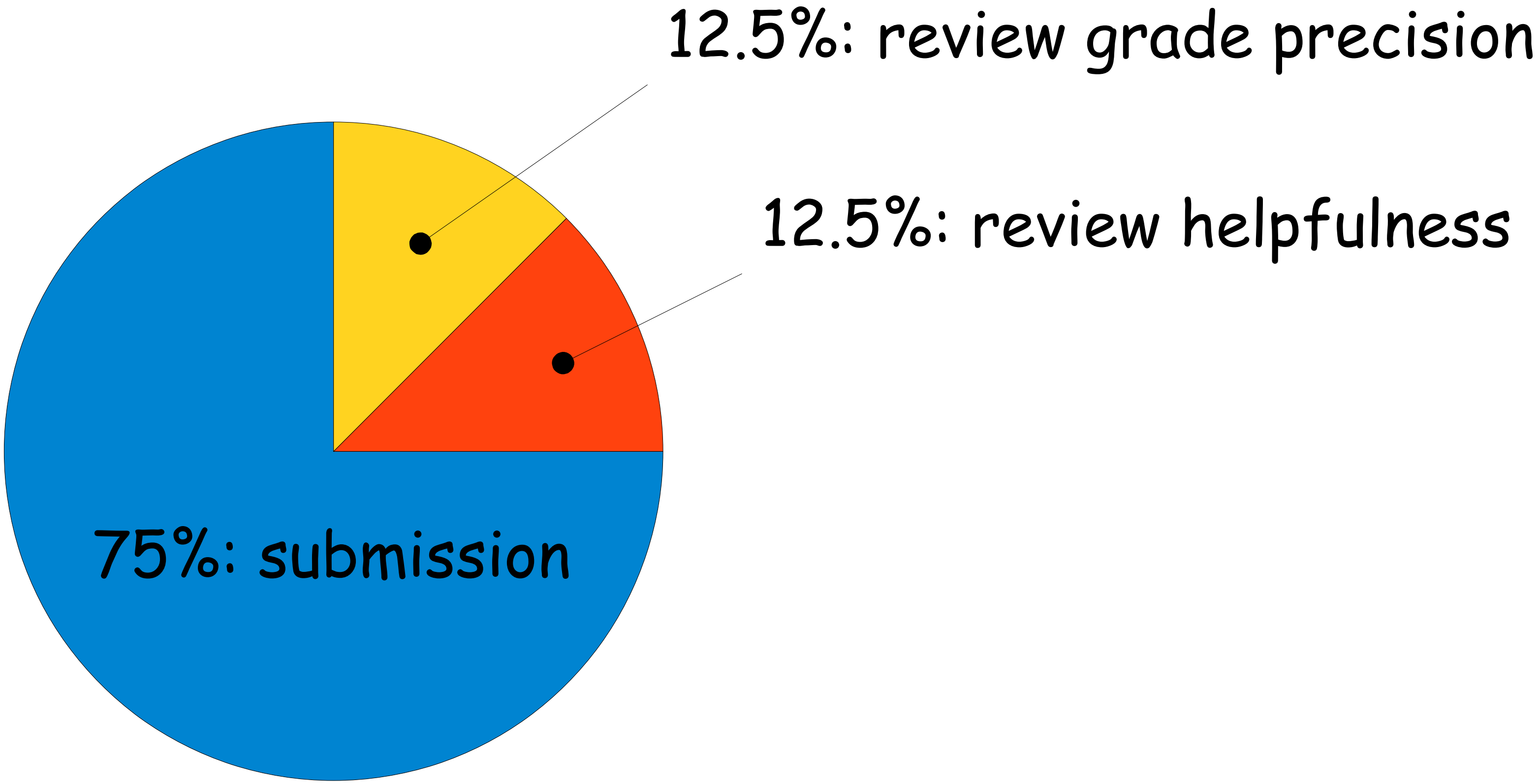
- 50 items, 50 users, 6 reviews per item.
- Users have gamma-distributed variance, with shape k=2, k=3.

Vancouver: Performance on synthetic data



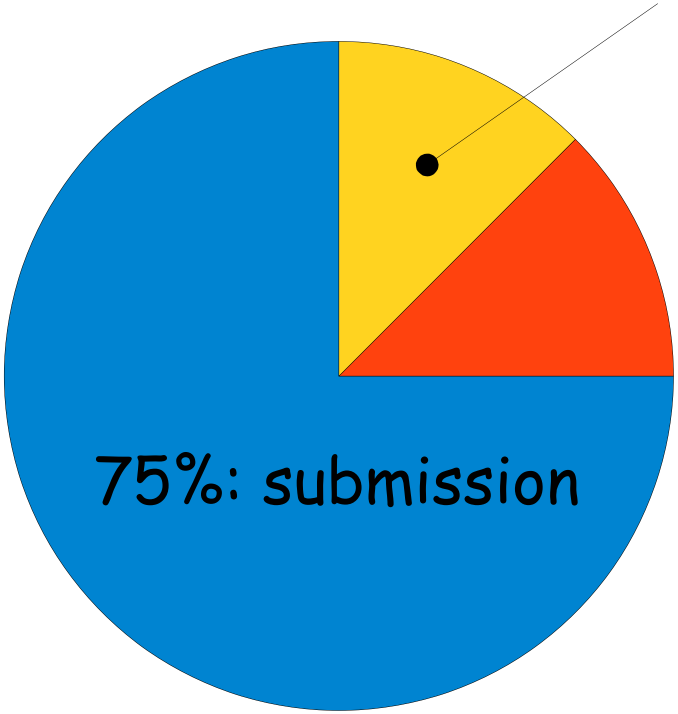
100 users
6 reviews
per item

Review Incentive: Crowd-Grade composition



The percentages can be changed by the instructor.

Review Incentive: Crowd-Grade composition

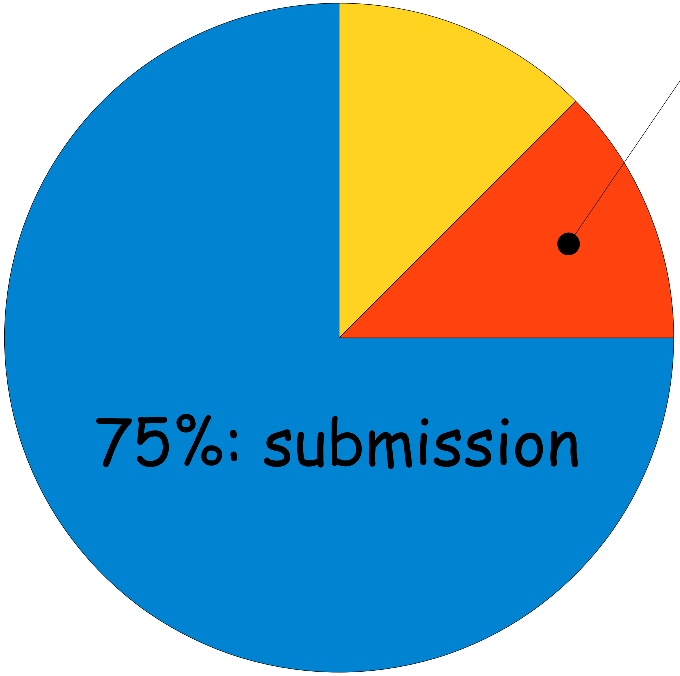


12.5%: review grade precision

$$f \cdot \left(1 - \frac{\sigma_u}{\sigma_r}\right)$$

- *f* : fraction of reviews done
- σ_u : standard dev of user
- σ_r : standard deviation of “standard” random user

Review Incentive: Crowd-Grade composition



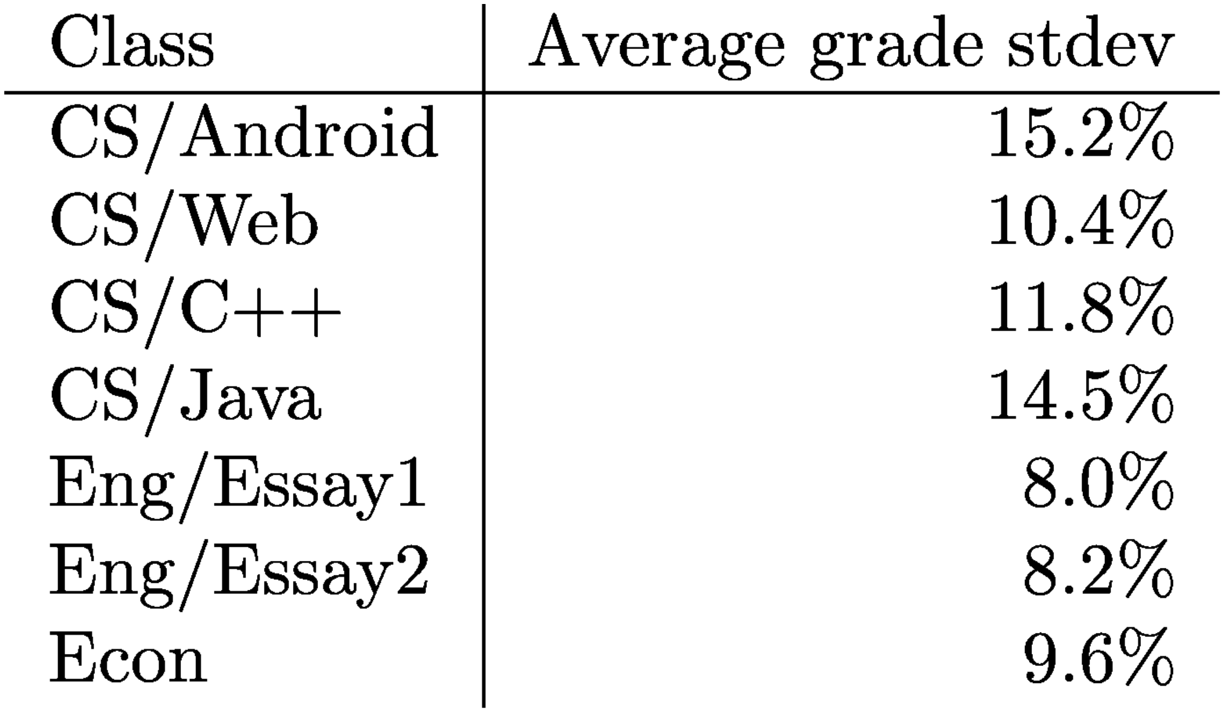
- Students give feedback on the reviews, and rate them:
  - +2: very helpful
  - ...
  - -2: bogus, very unhelpful
- We discard lowest rating, to avoid tit-for-tat.
- Average the rest, weighing negatives twice as much.
- Add to offset of 0.8, multiply by f

Effect of on-line predictive review assignment

Assignment		$\lvert S \rvert$	RevsDue	MinRevs	AvgRevs
CS/Android	hw 1	60	6	2	5.4
	hw 2	61	6	2	5.3
	hw 3	68	6	0	4.8
	hw 4	62	6	6	6.1
	hw 5	57	6	5	5.3
CS/C++	hw 1	102	5	0	4.6
	hw 2	97	5	3	4.6
	hw 3	91	5	4	5.1
	hw 4	97	5	3	4.6
	hw 5	90	5	4	5.1

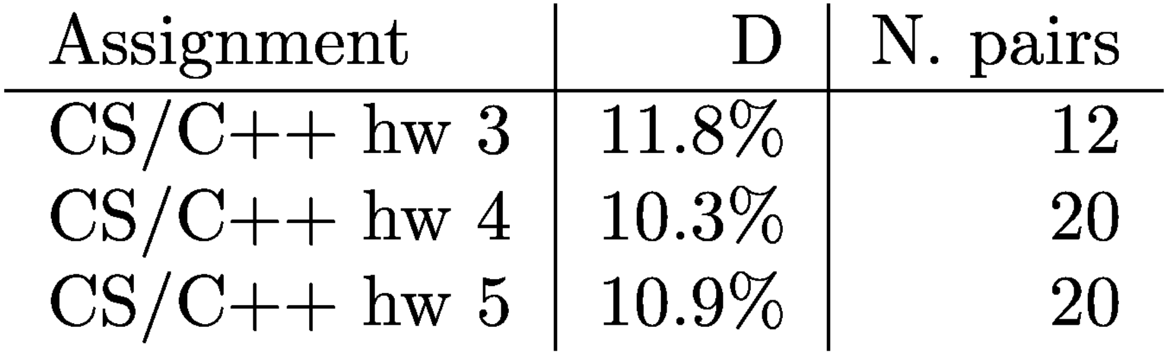
▨ : before on-line predictive review assignment was implemented.

Variance of grades given to the same assignment



Class	Average grade stdev
CS/Android	15.2%
CS/Web	10.4%
CS/C++	11.8%
CS/Java	14.5%
Eng/Essay1	8.0%
Eng/Essay2	8.2%
Econ	9.6%

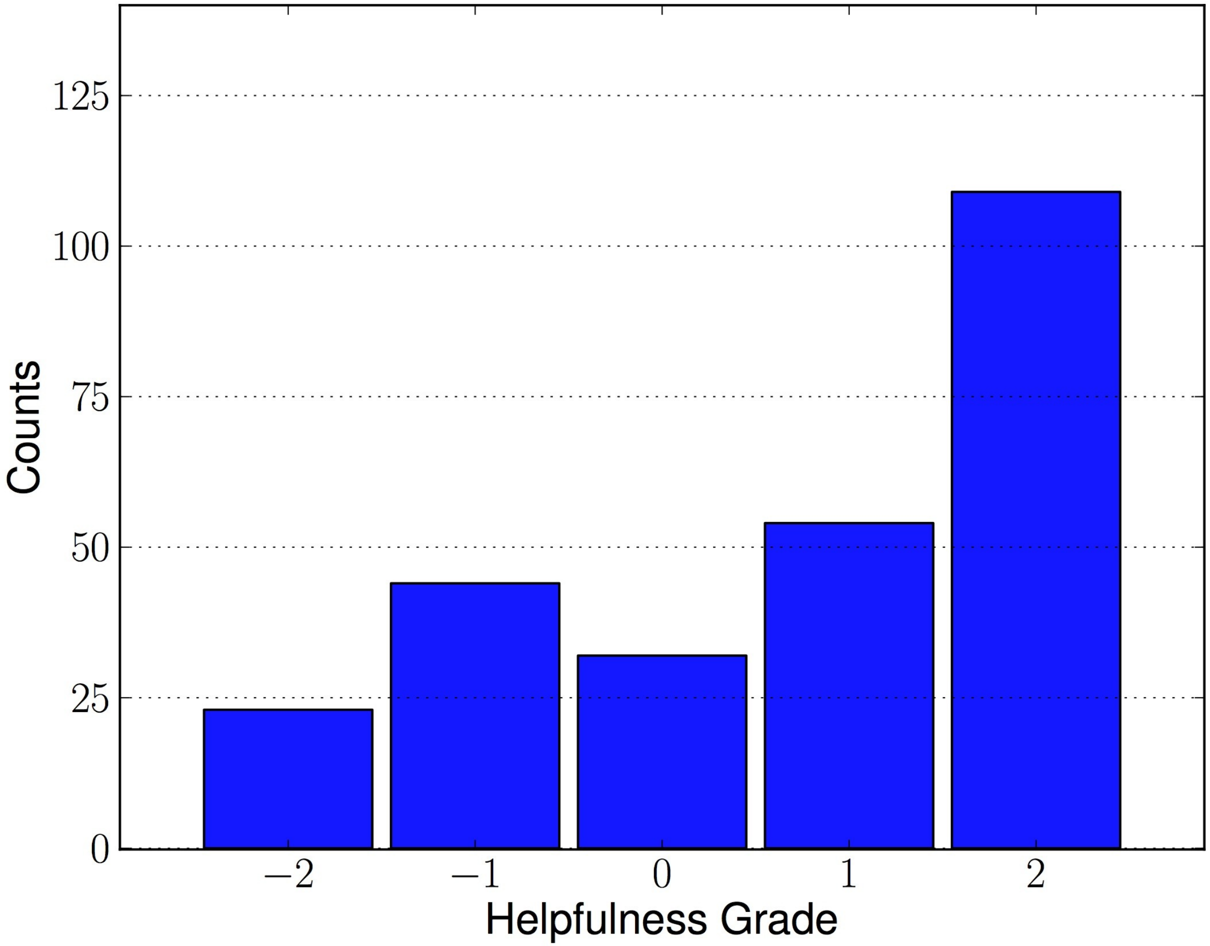
Difference in consensus grades received by pairs of identical submissions



Assignment	D	N. pairs
CS/C++ hw 3	11.8%	12
CS/C++ hw 4	10.3%	20
CS/C++ hw 5	10.9%	20

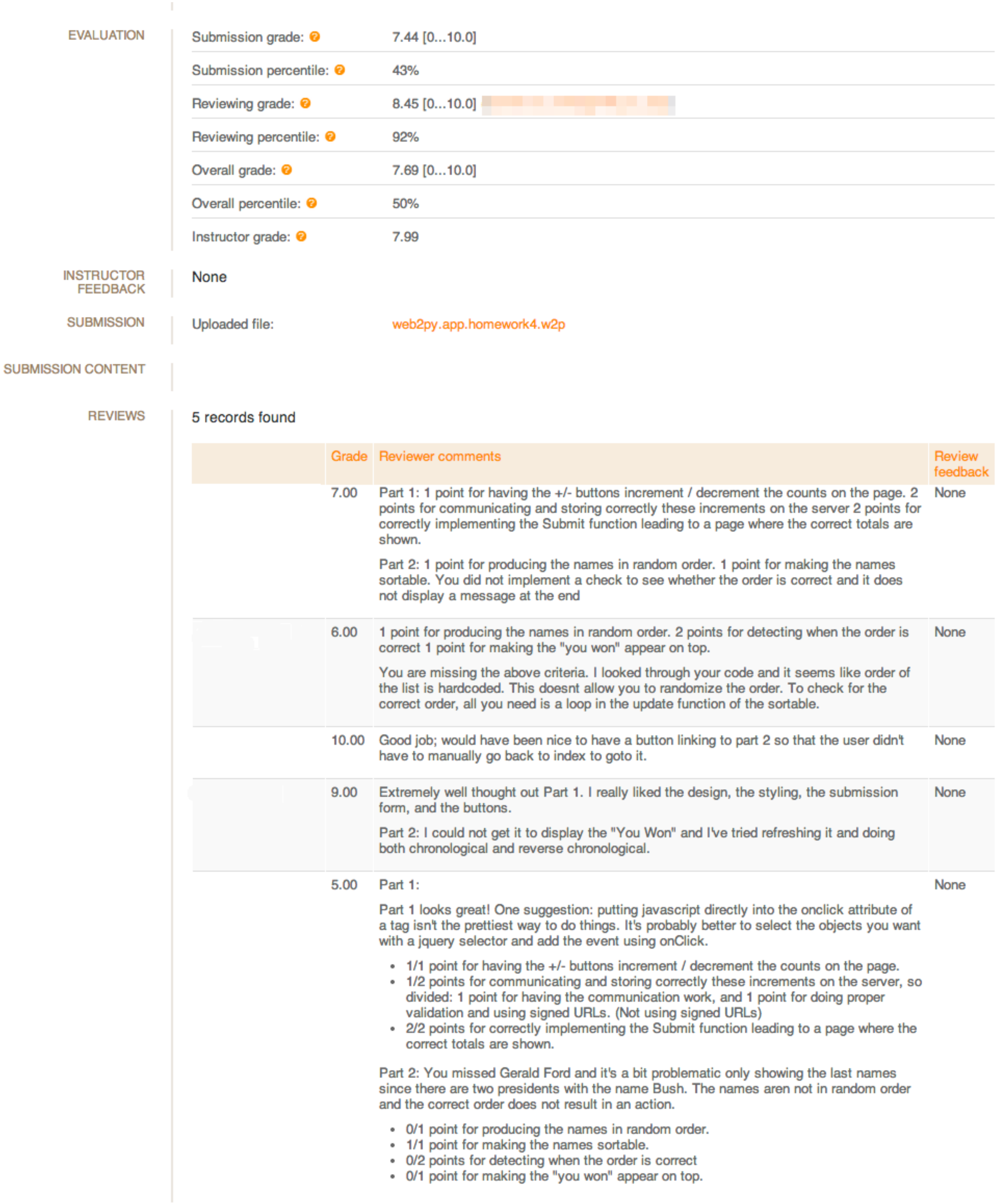
D is the square root of the mean square difference of the grades received by identical submissions, expressed as a percentage of the maximum grade M

Review helpfulness



How did the students benefit?

- **Motivation**: working for their peers.
- **Ability to examine other people's work**
  - If you cannot get it to work, you can look at how others solved it.
  - Multiple working examples.
  - Also multiple examples of errors.
- **Learn to be a reviewer** (important in code!)
- **Lots of feedback!**



EVALUATION

Submission grade:	7.44 [0...10.0]
Submission percentile:	43%
Reviewing grade:	8.45 [0...10.0] [redacted]
Reviewing percentile:	92%
Overall grade:	7.69 [0...10.0]
Overall percentile:	50%
Instructor grade:	7.99

INSTRUCTOR FEEDBACK

None

SUBMISSION

Uploaded file: web2py.app.homework4.w2p

SUBMISSION CONTENT

REVIEWS

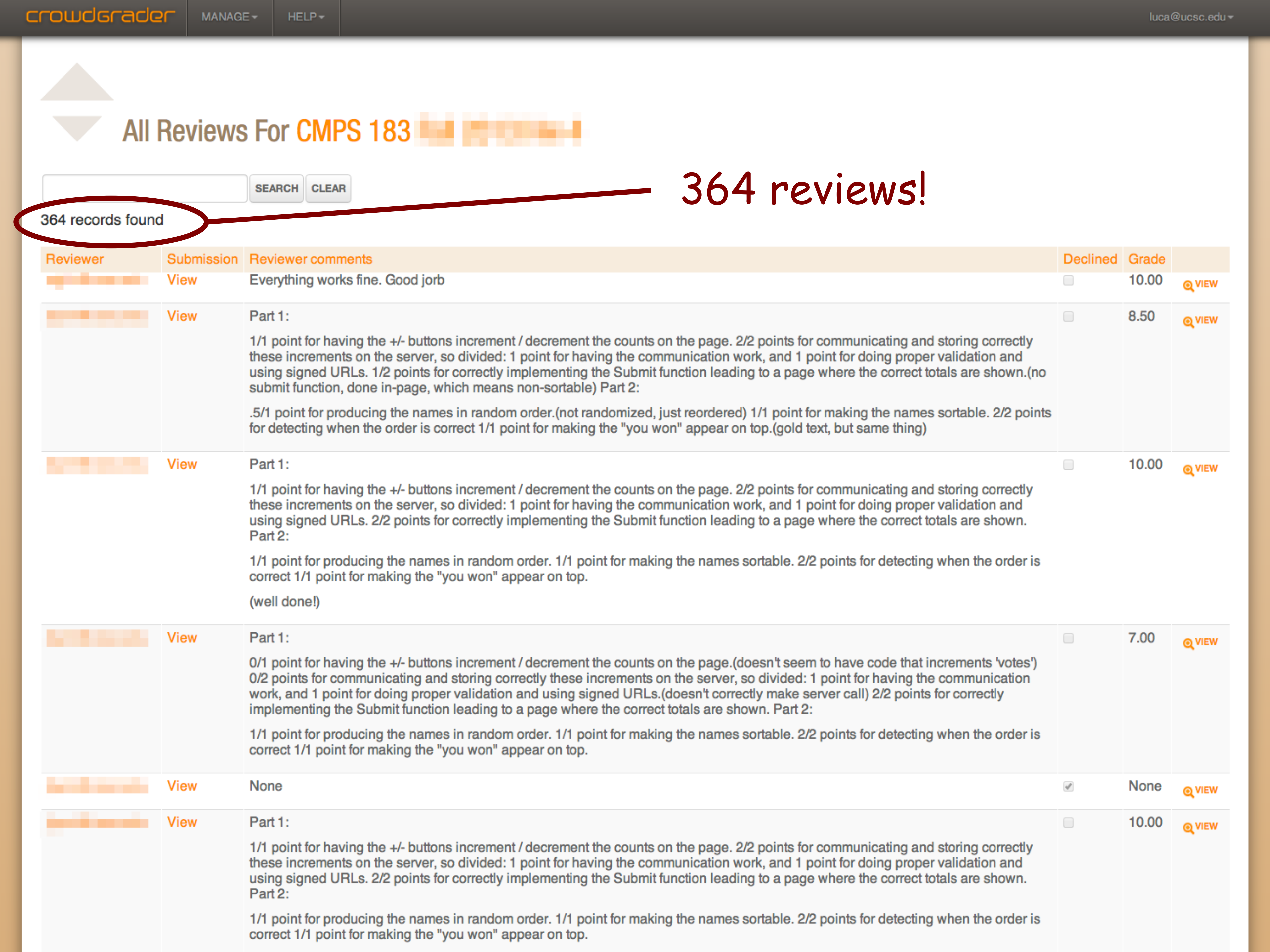
5 records found

	Grade	Reviewer comments	Review feedback
	7.00	Part 1: 1 point for having the +/- buttons increment / decrement the counts on the page. 2 points for communicating and storing correctly these increments on the server 2 points for correctly implementing the Submit function leading to a page where the correct totals are shown. Part 2: 1 point for producing the names in random order. 1 point for making the names sortable. You did not implement a check to see whether the order is correct and it does not display a message at the end	None
	6.00	1 point for producing the names in random order. 2 points for detecting when the order is correct 1 point for making the "you won" appear on top. You are missing the above criteria. I looked through your code and it seems like order of the list is hardcoded. This doesnt allow you to randomize the order. To check for the correct order, all you need is a loop in the update function of the sortable.	None
	10.00	Good job; would have been nice to have a button linking to part 2 so that the user didn't have to manually go back to index to goto it.	None
	9.00	Extremely well thought out Part 1. I really liked the design, the styling, the submission form, and the buttons. Part 2: I could not get it to display the "You Won" and I've tried refreshing it and doing both chronological and reverse chronological.	None
	5.00	Part 1: Part 1 looks great! One suggestion: putting javascript directly into the onclick attribute of a tag isn't the prettiest way to do things. It's probably better to select the objects you want with a jquery selector and add the event using onClick. • 1/1 point for having the +/- buttons increment / decrement the counts on the page. • 1/2 points for communicating and storing correctly these increments on the server, so divided: 1 point for having the communication work, and 1 point for doing proper validation and using signed URLs. (Not using signed URLs) • 2/2 points for correctly implementing the Submit function leading to a page where the correct totals are shown. Part 2: You missed Gerald Ford and it's a bit problematic only showing the last names since there are two presidents with the name Bush. The names aren not in random order and the correct order does not result in an action. • 0/1 point for producing the names in random order. • 1/1 point for making the names sortable. • 0/2 points for detecting when the order is correct • 0/1 point for making the "you won" appear on top.	None

Much more feedback than from a busy TA or instructor.

How do instructors benefit?

- Can handle large classes
- Motivate students
- Can assign homework regularly
- **Lots and lots of feedback** on how the class is going



All Reviews For CMPS 183 [redacted]

364 records found

364 reviews!

Reviewer	Submission	Reviewer comments	Declined	Grade	
[redacted]	View	Everything works fine. Good jorb	☐	10.00	VIEW
[redacted]	View	Part 1: 1/1 point for having the +/- buttons increment / decrement the counts on the page. 2/2 points for communicating and storing correctly these increments on the server, so divided: 1 point for having the communication work, and 1 point for doing proper validation and using signed URLs. 1/2 points for correctly implementing the Submit function leading to a page where the correct totals are shown.(no submit function, done in-page, which means non-sortable) Part 2: .5/1 point for producing the names in random order.(not randomized, just reordered) 1/1 point for making the names sortable. 2/2 points for detecting when the order is correct 1/1 point for making the "you won" appear on top.(gold text, but same thing)	☐	8.50	VIEW
[redacted]	View	Part 1: 1/1 point for having the +/- buttons increment / decrement the counts on the page. 2/2 points for communicating and storing correctly these increments on the server, so divided: 1 point for having the communication work, and 1 point for doing proper validation and using signed URLs. 2/2 points for correctly implementing the Submit function leading to a page where the correct totals are shown. Part 2: 1/1 point for producing the names in random order. 1/1 point for making the names sortable. 2/2 points for detecting when the order is correct 1/1 point for making the "you won" appear on top. (well done!)	☐	10.00	VIEW
[redacted]	View	Part 1: 0/1 point for having the +/- buttons increment / decrement the counts on the page.(doesn't seem to have code that increments 'votes') 0/2 points for communicating and storing correctly these increments on the server, so divided: 1 point for having the communication work, and 1 point for doing proper validation and using signed URLs.(doesn't correctly make server call) 2/2 points for correctly implementing the Submit function leading to a page where the correct totals are shown. Part 2: 1/1 point for producing the names in random order. 1/1 point for making the names sortable. 2/2 points for detecting when the order is correct 1/1 point for making the "you won" appear on top.	☐	7.00	VIEW
[redacted]	View	None	☑	None	VIEW
[redacted]	View	Part 1: 1/1 point for having the +/- buttons increment / decrement the counts on the page. 2/2 points for communicating and storing correctly these increments on the server, so divided: 1 point for having the communication work, and 1 point for doing proper validation and using signed URLs. 2/2 points for correctly implementing the Submit function leading to a page where the correct totals are shown. Part 2: 1/1 point for producing the names in random order. 1/1 point for making the names sortable. 2/2 points for detecting when the order is correct 1/1 point for making the "you won" appear on top.	☐	10.00	VIEW

crowdgrader

CrowdGrader lets students submit and collaboratively grade their solutions to homework assignments.

Used in 20+ institutions for 200+ assignments, ranging from CS to biology, economics, writing, engineering, ...

Over 50,000 reviews.

You can use it at

www.crowdgrader.org

You don't even need to create an account.

Thanks!