# Analysis of the PeerRank Method for Peer Grading

By

Joshua Kline

* * * * * * * * *

Submitted in partial fulfillment
of the requirements for a degree in the
Departments of Computer Science and Mathematics

UNION COLLEGE

June, 2015

**Abstract**

**Advisors:** Matthew Anderson & William Zwicker

Peer grading can have many benefits in education, including a reduction in the time instructors spend grading and an opportunity for students to learn through their analysis of others work. However, when not handled properly, peer grading can be unreliable and may result in grades that are vastly different from those which a student truly deserves. Therefore, any peer grading system used in a classroom must consider the potential for graders to generate inaccurate grades. One such system is the PeerRank rule proposed by Toby Walsh [11], which uses an iterative, linear algebra based process reminiscent of the Google PageRank algorithm [6] in order to produce grades by weighting the peer grades with the graders accuracies. However, this system has certain properties which make it less than ideal for peer grading in the classroom. We propose a modification of PeerRank that attempts to make the system more applicable in a classroom environment by incorporating the concept of "ground truth" to provide a basis for accuracy. We then perform a series of experiments to compare the accuracy of our method to that of PeerRank. We conclude that, in cases where a grader's accuracy in grading others is a reflection of their own grade, our method produces grades with a similar accuracy to PeerRank. However, in cases where a grader's accuracy and grade are unrelated, our method performs more accurately than PeerRank.

# Contents

# 1 Introduction

In the context of education, peer grading is a process in which each student in a class receives a set of grades from their classmates for an assignment, and they, in turn, provide grades for their classmates' assignments. An example of one student's role in this process is diagrammed in Figure 1. Many educators have argued that peer grading can be a valuable tool in enhancing the overall learning experience for students. Sadler and Good [8] suggest several potential benefits to peer grading in the classroom, including (i) a reduction in the time instructors spend on grading, (ii) faster and more detailed feedback on student assignments, (iii) an increase in a student's understanding of the material by requiring them to evaluate the work of their classmates, and (iv) the possibility for students to identify the strengths and weaknesses in their own assignments. Topping [10] expresses similar opinions, and adds that it "involves students directly in the learning process" and helps them develop social skills such as the ability to accept criticism. In addition, Cho and Schunn [3] point out that when instructors bear the burden of grading all submissions for an assignment they may be forced to limit the number of writing assignments given in the class, which hampers the ability of students to develop their writing skills. Furthermore, the recent introduction of massive open-access online courses (MOOCs) such as those provided by EdX and Coursera makes it possible for thousands of students to participate in the same class, in which case it is impossible for an instructor to manually grade all the submissions for any assignment. It is therefore clear that peer grading systems have the potential to be a beneficial, and sometimes essential, tool in education.

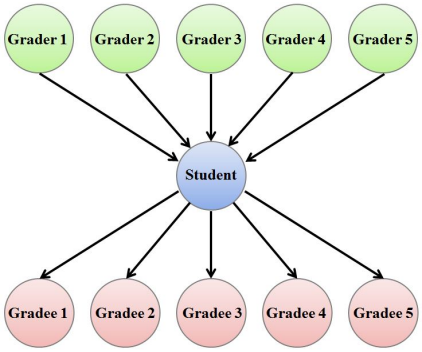



Figure 1: An example of one student's role in the peer grading process. Here the student both grades and is graded by five of their peers.

However, an immediate concern about the peer grading approach is the overall reliability of a grading system, that is, the likelihood that each student receives a grade that is close to the one they truly deserve. If a system is unreliable then it is possible that a student may receive a grade that is significantly higher or lower than their deserved grade, and the negative consequences of peer grading could outweigh any potential benefits it may provide. While there are several possible sources for inaccurate grades, we identify two particular causes:

Issue 1. A grader may have an overall tendency to grade their peers either more leniently or more harshly than is deserved due to either their own inexperience in grading or a poor understanding of the material being graded.

Issue 2. If a grader does not care about the peer grading process then it is likely that they will make no effort to grade with accuracy.

In our opinion, it is essential that any peer grading system used in a classroom address these issues in order to provide an accurate grade.

There have been several different approaches to providing accurate peer grading, each producing algorithms with varying mathematical foundations. We discuss several of these systems in Section 2, and there are two common aspects of them that address the previously-mentioned issues:

Solution 1. Make the individual grades from inaccurate graders have a smaller impact on a student's overall grade than those provided by accurate graders.

Solution 2. Provide students with some sort of incentive to grade accurately.

One unique approach to peer evaluation in grant reviewing is the PeerRank system proposed by Toby Walsh [11]. In PeerRank, each member of a group of "agents" provides a grade for the work submitted by each of the agents in the group. The PeerRank algorithm combines this set of individual grades into a final grade, using an iterative process with an underlying mathematical foundation in linear algebra similar to that of Google's PageRank algorithm for ranking web pages [6]. In short, an individual's grade is based both on the grades provided by the other agents, which are weighted by the grading agent's own grade, and on their own ability to grade accurately. The details of how PeerRank achieves this are given in Section 3.

However, Walsh's PeerRank system has certain issues that can cause it to produce inaccurate grades. First, if a majority of the agents share a common belief relating to the subject of the works being graded, then the grades of the agents outside that majority could unfairly suffer. For example, suppose that PeerRank is being used to grade a set of physics research proposals whose authors are requesting funding. If a majority of the physicists believe in string theory and reject all other models of the universe, then any proposal that affirms belief in some other model may receive a lower grade than the proposals for string theory research, regardless of its quality. Second, the peer grades provided by each agent are weighted based on an important assumption.

**Walsh's Assumption.** *An agent's accuracy in grading their peers is equal to their own grade.*

In other words, Walsh assumes that there is a single score that reflects both how well an agent performs on their own submission and how well they grade their peers' submissions. PeerRank uses this score when weighing the peer grades from different agents. While Walsh's Assumption could be true in PeerRank's original context of grant reviewing, it is not necessarily true in classroom grading. For example, a student who may not have understood how to answer a question when completing their own assignment may understand and learn from the answers of their peers, in which case their grading accuracy could be higher than their own grade. These issues are discussed in more depth in Section 3.3.

We address issues found in Walsh's PeerRank by developing a system that can be used to provide accurate peer grading in the classroom. In classes, we assume that there is some sort of "ground truth" grade for each assignment that is determined by the instructor. In our approach to peer grading, we propose that the instructor provides their own sample submission to be used as a basis for determining ground truth. We then use this basis to produce an accuracy score for each grader that is independent of their own grade, rejecting Walsh's Assumption. Our proposed peer grading system uses these accuracy scores, rather than the graders' own grades, when weighing the peer grades in order to integrate the concept of ground truth with the ideas developed by Walsh for PeerRank. We evaluated this system by testing it against simulated grade data and comparing the results to those from PeerRank, and found that our system had a slight improvement in accuracy over PeerRank in cases where Walsh's Assumption fails to hold.

In Section 2, we discuss some of the past approaches to the problem of accurate peer grading. Then, in

Section 3, we explain the details of exactly how PeerRank calculates grades and outline certain properties of the system, as well as discuss some of the issues that could arise from applying PeerRank in the classroom. In Section 4, we present our proposed method and discuss how it addresses the issues resulting from PeerRank. In Section 5, we discuss the experiments with which we tested the performance of our method against PeerRank, and then present our results. Finally, in Sections 6 and 7, we discuss our conclusions and present potential continuations of our research.

## 2 Background and Related Work

There have been many articles published about peer grading and evaluation, both about its benefits and impacts in education [3, 8, 10], and about systems for conducting it. As stated in the introduction, the work most closely related to this thesis is the PeerRank system [11]. PeerRank is based largely on the same foundations in linear algebra as the PageRank algorithm for ranking web pages developed by Page *et al.* [1, 6] and used in the Google search engine. The process by which PeerRank determines grades, as well as the issues with its implementation in the classroom, will be discussed in depth in Section 3.

Piech *et al.* [2] developed probabilistic models for peer grading. Their algorithms attempt to provide accurate grading by adjusting for grader bias and, like PeerRank, incorporate a grader's accuracy as a factor in their own grade. Their process was established in the context of MOOCs offered on Coursera. Each student is given five assignments to grade, one of which comes from a set of three to five randomly-selected submissions that have been declared as "ground truth". While the authors referred to these submissions as "ground truth", it is important to note that the submissions were not necessarily examples of a perfect submission. The authors instead gave the set this name in order to reflect its use as a basis for determining grader accuracy. Since a MOOC can have thousands of students and there are a very small number of ground truth submissions, each one is graded hundreds of times. By the law of large numbers, it is assumed that the average of the grades for a ground truth submission is close to the correct grade, and therefore these submissions can be used as a basis for determining grader accuracy. However, because the ground truth grade is determined by the grading students rather than the instructor it is possible that the resulting grade will be inaccurate, especially if a majority of the graders have a poor understanding of the material. This

means that the method is somewhat susceptible to Issue 1 from Section 1. Another major difference between this work and PeerRank is that the probabilistic models they use produce a "belief distribution" of grades, instead of a single grade, for each student. The size of this distribution is based on the model's confidence in the score it generates. As we will explain in Section 4, the process through which this system generates grader accuracy scores served as an inspiration for our proposed method.

Another peer grading system for education is the CrowdGrader system developed by de Alfaro and Shavlovsky [4]. In CrowdGrader, a student's final grade is a combination of three separately determined grades. First is the consensus grade, which is an average of the grades received from peers, taken after the highest and lowest grades have been removed and the grades have been weighted by the graders' accuracies. Second is the accuracy grade, which reflects the student's overall accuracy in grading others and is computed using an average square error. The third is the helpfulness grade, which incorporates an additional step in which the students rate the quality of the feedback given to them by their graders. By combining these ratings into the helpfulness grade, CrowdGrader encourages graders to provide useful feedback to their gradees, instead of just a grade. However, according to the authors the absolute difference (on a grading scale from 0% to 100%) between the grades produced by CrowdGrader and control grades produced by instructors averaged around 15%. They note that this inaccuracy is comparable to that in grades produced by teaching assistants, and that in cases where students claimed to have been mis-graded the instructors were able to use the student feedback to determine the correct grades. However, such a low accuracy is concerning for a system that is intended to grade students.

While it is not a grading system, a peer review system that is closely related to many other such systems, including PeerRank and CrowdGrader, is the system developed by Merrifield and Saari for ranking research proposals [7], which was adapted by the NSF for the Signal and Sensing Systems program. Each individual from the group submitting proposals is sent a certain number of proposals to organize in a ranked list, and the individual lists are combined into an overall ranked list using a mathematical process based on the Borda count, that is, an algorithm for combining voters' individual preferences into an overall ranking. The overall list is then compared to each evaluator's list to determine their accuracy in ranking proposals, and those evaluators who ranked accurately are rewarded by having their own proposal moved

up slightly in the overall list.

# 3 PeerRank

In this section, we define our grading scenario, and then provide the details of PeerRank's process of calculating grades. Suppose that we have a group of $m$ agents, numbered from 0 to $m-1$, which in the context of classroom grading are students. Each agent $j$ provides a grade $A_{i,j}$ in the range $[0,1]$ for each agent $i$'s submission (meaning that we have a total of $m^2$ peer grades). After the peer grades are generated they are assembled into a grade matrix $A$, where the $i$th row of $A$ contains all of the grades received by agent $i$ and the $j$th column contains all of the grades given by agent $j$. Therefore, we denote the grade that agent $i$ received from agent $j$ as $A_{i,j}$.

PeerRank [11] uses an iterative process in order to generate a grade vector $\vec{X}$, where $X_i$ contains the overall grade for agent $i$, and this grade vector is repeatedly updated. We begin the process using a vector $\vec{X}^0$ called the "initial seed", which we substitute into an equation in order to generate a new grade vector $\vec{X}^1$. This process is repeated using each of the new grade vectors we generate until we approximate a fixed point, which is a grade vector that no longer changes with additional iterations (i.e., $\vec{X}^{n+1} \approx \vec{X}^n$).

Walsh gives two different versions of his PeerRank process, which combine the peer grades in $A$ to determine a final grade $X_i$ for each agent $i$. The first is a basic version that provides no incentive for accurate grading, and the second is a generalized version that includes an additional term to incentivize accurate grading. We describe both versions, and prove properties of each.

## 3.1 Basic Version

Let $X_i^n$ be the grade of agent $i$ in the $n$th iteration of PeerRank. We start by choosing a value for $\alpha$ such that $0 < \alpha < 1$. Walsh states that the choice of $\alpha$ has no effect on the final result, and merely affects the speed of convergence [11].

Next, we construct our initial seed grade vector $\vec{X}^0$. For each agent $i$, we set $i$'s initial grade to be the

average of the grades they received.

$$X_i^0 = \frac{1}{m} \sum_{j=0}^{m-1} A_{i,j} \tag{1}$$

However, as we will show later in Theorem 1, the choice of our initial seed has no impact on the final result, as long as it does not contain any zeros. Next, we iteratively calculate each agent's grade in the $(n+1)$st iteration using the grades in the $n$th iteration:

$$X_i^{n+1} = (1-\alpha) \cdot X_i^n + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} \tag{2}$$

This equation can be rewritten in vector form:

$$\vec{X}^{n+1} = (1-\alpha) \cdot \vec{X}^n + \frac{\alpha}{\left|\left|\vec{X}^n\right|\right|_1} \cdot A\vec{X}^n \tag{3}$$

where $\left|\left|\vec{V}\right|\right|_1 = \sum_i |v_i|$ is the *one norm* of $\vec{V}$.

The second term in this equation produces weighted averages of the grades each agent received, where the weights used are the graders' grades in the previous iteration. This equation is repeated iteratively until we approximate a fixed point, i.e., until $\vec{X}^{n+1} \approx \vec{X}^n$. The resulting fixed point is the output of PeerRank, and contains the final grade assigned to each agent.

#### 3.1.1 Properties

In this section, we prove several important properties about the basic version of PeerRank. The first two of these properties, along with several others, were proven by Walsh [11]. The first property, named here as Proposition 1, states that the fixed point, $\vec{X}$, for this version of PeerRank is an eigenvector of the grade matrix $A$ with eigenvalue $\left|\left|\vec{X}\right|\right|$, meaning that $A\vec{X} = \left|\left|\vec{X}\right|\right| \vec{X}$. The second property, named here as Proposition 2, states that the domain for grades in the output $\vec{X}$ is the same as the domain for grades in the grade matrix $A$. We then use these properties, along with some additional propositions, to prove that the fixed point for the basic version of PeerRank is unique. This is an extremely useful property since it means that we can use almost any grade vector as our initial seed and reach the same fixed point.

It should be noted that Walsh defines the domain of possible grades as [0,1]. However, in Proposition 4 we will show that Theorem 1 is not guaranteed to apply if we allow grades of 0. Therefore, our statements and proofs of Proposition 1, Proposition 2, and Theorem 1 restrict the domain of allowed grades to $(0, 1]$.

We begin by proving the first two properties following the proofs given by Walsh. Note that our statement and proof of Proposition 1 are stronger than Walsh's equivalent statements [11].

**Proposition 1.** *([11], Proposition 1) The vector $\vec{X} \in (0, 1]^m$ is a fixed point of the basic version of PeerRank if and only if it is an eigenvector of the grade matrix $A$ with eigenvalue $\left|\left|\vec{X}\right|\right|$.*

*Proof.* For the forward direction, assume that $\vec{X}$ is a fixed point of the basic version of PeerRank with the grade matrix $A$. This means that

$$\begin{aligned}
\vec{X} &= (1-\alpha)\cdot\vec{X} + \frac{\alpha}{\left|\left|\vec{X}\right|\right|_1}\cdot A\vec{X}, \\
\vec{X} &= \vec{X} - \alpha\vec{X} + \frac{\alpha}{\left|\left|\vec{X}\right|\right|_1}\cdot A\vec{X}, \\
\frac{\alpha}{\left|\left|\vec{X}\right|\right|_1}\cdot A\vec{X} &= \alpha\vec{X}.
\end{aligned}$$

If we divide by $\alpha$ and let $\lambda = \left|\left|\vec{X}\right|\right|_1$, we have

$$\begin{aligned}
\frac{1}{\lambda}\cdot A\vec{X} &= \vec{X}, \\
A\vec{X} &= \lambda\vec{X}.
\end{aligned}$$

Therefore, $\vec{X}$ is an eigenvector of $A$ with eigenvalue $\left|\left|\vec{X}\right|\right|_1$.

We can reverse these steps to prove the reverse direction. □

**Proposition 2.** *([11], Proposition 2) If $\vec{X}$ is a fixed point of the basic version of PeerRank with an $m \times m$ grade matrix $A$, then $\vec{X} \in (0, 1]^m$.*

*Proof.* We will prove that $X_i^n \in (0, 1]$ for all $i$ and all $n$ using induction. For the base case, it is clear that $X_i^0 \in (0, 1]$ since it is the average of terms which are all in $(0, 1]$. For the inductive case, assume that

$X_i^n \in (0, 1]$ for all $i$. Then we have

$$\begin{aligned}
X_i^{n+1} &= (1-\alpha) \cdot X_i^n + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} \\
&> (1-\alpha) \cdot 0 + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} && \text{since } X_i^n > 0 \\
&> \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} 0 && \text{since } X_j^n > 0 \text{ and } A_{i,j} > 0 \\
&= 0.
\end{aligned}$$

Therefore $X_i^{n+1} > 0, \forall i$. Now let $X_i^n = 1 - \epsilon$ where $0 \leq \epsilon < 1$, we have

$$\begin{aligned}
X_i^{n+1} &= (1-\alpha)(1-\epsilon) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} \\
&= 1 - \alpha - \epsilon(1-\alpha) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} \\
&\leq 1 - \alpha - \epsilon(1-\alpha) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n && \text{since } A_{i,j} \leq 1 \\
&= 1 - \alpha - \epsilon(1-\alpha) + \alpha \\
&= 1 - \epsilon(1-\alpha) \\
&\leq 1.
\end{aligned}$$

Therefore $0 < X_i^{n+1} \leq 1$. □

Next we will prove that the fixed point of the basic version of PeerRank is unique regardless of our choice of the initial seed (as long as it does not contain zeros). Our proof is based heavily on the proof for the uniqueness of the fixed point in PageRank [1]. First, we present the following proposition from Bryan and Leise [1].

**Proposition 3.** *([1], Proposition 3) Let $\vec{v}$ and $\vec{w}$ be linearly independent vectors in $\mathbb{R}^m$, $m \geq 2$. Then, for some real values $s$ and $t$ that are not both zero, the vector $\vec{x} = s\vec{v} + t\vec{w}$ has both positive and negative components.*

*Proof.* Because $\vec{v}$ and $\vec{w}$ are linearly independent, neither are equal to $\vec{0}$. Let $d = \sum_i v_i$. We now have two cases:

Case 1: Assume $d = 0$. Then because $\vec{v} \neq \vec{0}$, $\vec{v}$ must have both positive and negative components. Therefore, if we let $s = 1$ and $t = 0$, then $\vec{x} = s\vec{v} + t\vec{w} = \vec{v}$ has both positive and negative components.

Case 2: Assume $d \neq 0$. Then let $s = -\frac{\sum_i w_i}{d}$ and $t = 1$. Since $\vec{v}$ and $\vec{w}$ are linearly independent, $\vec{x} = s\vec{v} + t\vec{w} \neq \vec{0}$. However, $\sum_i x_i = 0$. Therefore, $\vec{x}$ must have both positive and negative components.

□

Next we prove the following lemma, which will be essential in proving Theorem 1. We define an *eigenspace* as the space of eigenvectors of a matrix with the same eigenvalue, and the *dimension* of an eigenspace as the size of a maximal set of linearly independent vectors in the eigenspace.

**Lemma 1.** *Suppose that $\vec{X}$ is a fixed point of the basic version of PeerRank with grade matrix $A$. Then the eigenspace with eigenvalue $\left|\left|\vec{X}\right|\right|_1$ has dimension 1.*

*Proof.* Suppose for contradiction that there exist two linearly independent fixed points $\vec{V}$ and $\vec{W}$ both in the same eigenspace of A, $V_\lambda$, where $\lambda = \left|\left|\vec{V}\right|\right|_1 = \left|\left|\vec{W}\right|\right|_1$ by Proposition 1. Then for any real numbers $s$ and $t$ both not zero, the vector $\vec{X} = s\vec{V} + t\vec{W}$ is nonzero and must be in $V_\lambda$ since vector spaces are closed under linear combination. We can then rescale our choice of $s$ and $t$ so that $\left|\left|\vec{X}\right|\right|_1 = \lambda$. By Proposition 1, this means $\vec{X}$ is also a fixed point. So by Proposition 2, the components of $\vec{X}$ are all in the range $(0, 1]$. However, by Proposition 3, for some choice of $s$ and $t$ that are not both zero, $\vec{X}$ must contain both positive and negative components, a contradiction. Therefore, $V_\lambda$ cannot contain two linearly independent vectors, and so it has dimension 1. □

Now, we prove Theorem 1, which states that a fixed point of the basic version of PeerRank is unique regardless of our choice of initial seed.

**Theorem 1.** *The basic version of PeerRank at most one fixed point for each grade matrix $A$.*

*Proof.* Suppose for contradiction that there exist two linearly independent fixed points $\vec{V}$ and $\vec{W}$, both nonzero, for PeerRank with the $m \times m$ grade matrix $A$. Then by Proposition 1, $\vec{V}$ is an eigenvector of $A$ with eigenvalue $\left|\left|\vec{V}\right|\right|_1$, and $\vec{W}$ is an eigenvector of $A$ with eigenvalue $\left|\left|\vec{W}\right|\right|_1$. Furthermore, by Lemma 1, the eigenspaces with eigenvalues $\left|\left|\vec{V}\right|\right|_1$ and $\left|\left|\vec{W}\right|\right|_1$ each have dimension 1. Therefore, since $\vec{V}$ and $\vec{W}$ are linearly independent, $\left|\left|\vec{V}\right|\right|_1 \neq \left|\left|\vec{W}\right|\right|_1$.

Suppose, without loss of generality, $\left|\left|\vec{V}\right|\right|_1 > \left|\left|\vec{W}\right|\right|_1$. Let $\vec{X} = s\vec{V} + t\vec{W}$, where $s = -1$ and $t = \max_{1 \leq j \leq n} \frac{V_j}{W_j}$. Note that $\vec{X}$ has no negative components, since for $t = \frac{V_j}{W_j}$ and $i$ where $1 \leq i \leq n$ we have

$$\begin{aligned} x_i &= sV_i + tW_i \\ &= -V_i + \frac{V_j}{W_j} W_i \\ &\geq -V_i + \frac{V_i}{W_i} W_i \\ &= 0. \end{aligned}$$

Then by linearity

$$\begin{aligned} A\vec{X} &= As\vec{V} + At\vec{W} \\ &= s\left|\left|\vec{V}\right|\right|_1 \vec{V} + t\left|\left|\vec{W}\right|\right|_1 \vec{W} \\ &= \begin{bmatrix} s\left|\left|\vec{V}\right|\right|_1 V_1 + t\left|\left|\vec{W}\right|\right|_1 W_1 \\ \vdots \\ s\left|\left|\vec{V}\right|\right|_1 V_n + t\left|\left|\vec{W}\right|\right|_1 W_n \end{bmatrix}. \end{aligned}$$

Isolating the $j$th component of $A\vec{X}$, we have

$$\begin{aligned} s\left|\left|\vec{V}\right|\right|_1 V_j + t\left|\left|\vec{W}\right|\right|_1 W_j &= -\left|\left|\vec{V}\right|\right|_1 V_j + \frac{V_j}{W_j}\left|\left|\vec{W}\right|\right|_1 W_j \\ &= -\left|\left|\vec{V}\right|\right|_1 V_j + \left|\left|\vec{W}\right|\right|_1 V_j \\ &< 0 \end{aligned}$$

since $\left|\left|\vec{V}\right|\right|_1 > \left|\left|\vec{W}\right|\right|_1$.

This means that the $j$th component of $A\vec{X}$ is negative. However, because all components in $A$ are positive and all components in $\vec{X}$ are nonnegative, all components of $A\vec{X}$, including the $j$th component, must also be nonnegative. This means we have reached a contradiction. Therefore, there can be at most one non-zero fixed point. □

Note that while Theorem 1 proves that there can be at most one fixed point, neither we nor Walsh prove that a fixed point is guaranteed to exist or that PeerRank will converge to a fixed point from every initial seed. However, experimental results suggest that these properties are true.

Finally, we address our decision to restrict the domain of possible grades to $(0, 1]$. In his proposal of PeerRank, Walsh specifies the domain of grades as $[0, 1]$, which includes grades of 0. However, in the proof of following proposition we show that if we allow grades of 0 then we can provide counterexamples of Theorem 1, meaning that the fixed point of the basic version of PeerRank is no longer guaranteed to be unique.

**Proposition 4.** *Suppose that the grade matrix $A$ contains entries all in the range $[0, 1]$. Under this assumption, the basic version of PeerRank may have multiple fixed points.*

*Proof.* Let $A$ be the following $2 \times 2$ grade matrix:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

First, let our initial seed be $\vec{X}^0 = [1, 0]^T$. Then the basic version of PeerRank produces the fixed point $[1, 0]^T$.

Now, let our initial seed be $\vec{X}^0 = [0, 1]^T$. Then the basic version of PeerRank produces the fixed point $[0, 1]^T$. Therefore, multiple fixed points exist. □

While it may seem like a problem for a grading system to not allow grades of 0, this issue can be solved by replacing grades of 0 with an extremely small positive value. The effect of these low grades on the weighing of peer grades in PeerRank is extremely similar to the effect from grades of 0, and the added requirement of positivity ensures that PeerRank can only produce one fixed point. Therefore in all of our future examples, grades of 0 are assumed to have been replaced with a very small positive grade.

## 3.2 Generalized Version

The generalized version of PeerRank adds an extra term to the equation that provides an incentive for graders to grade accurately. Let $\alpha$ and $\beta$ be nonnegative values such that $\alpha + \beta \leq 1$. Next, we construct our initial seed grade vector $\vec{X}^0$ using Equation 1 as in the basic version of PeerRank. However, we define our update step as

$$X_i^{n+1} = (1 - \alpha - \beta) \cdot X_i^n + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n} \cdot \sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} + \frac{\beta}{m} \cdot \sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| \tag{4}$$

The new term in this equation, as compared to Equation 2, incorporates a factor of $\beta$ into agent $i$'s grade based on their accuracy in grading others. It does so by measuring the absolute difference between the grade given by agent $i$ to agent $j$ for all $j$, subtracting each of these differences from 1 to gain an accuracy measure, and then averaging the results. The resulting value represents agent $i$'s overall grading accuracy, which is then multiplied by $\beta$. Therefore, our choice of $\beta$ impacts how much influence each agent's grading accuracy has on their own grade. Note that when $\beta = 0$, we are left with Equation 2.

### 3.2.1 Properties

We now wish to prove that the same useful properties that apply to the basic version of PeerRank also apply to the generalized version. We therefore reprove the proposition covering the domain of the fixed point.

**Proposition 5.** *If $\vec{X}$ is a fixed point of the generalized version of PeerRank with an $m \times m$ grade matrix $A$, then $\vec{X} \in (0,1]^m$.*

*Proof.* We will prove that $X_i^n \in (0,1]$ for all $i$ and all $n$ using induction. For the base case, it is clear that $X_i^0 \in (0,1]$ since it is the average of terms which are all in $(0,1]$. For the inductive case, assume that $X_i^n \in (0,1]$ for all $i$. Then we have

$$\begin{aligned}
X_i^{n+1} &= (1-\alpha)\cdot X_i^n + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| \\
&> (1-\alpha)\cdot 0 + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| && \text{since } X_i^n > 0 \\
&> \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} 0 + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| && \text{since } X_j^n > 0 \text{ and } A_{i,j} > 0 \\
&\geq \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - 1 && \text{since } X_j^n, A_{j,i} \in (0,1] \\
&= 0.
\end{aligned}$$

Therefore $X_i^{n+1} > 0$. Also, letting $X_i^n = 1-\epsilon$ where $0 \leq \epsilon < 1$, we have

$$\begin{aligned}
X_i^{n+1} &= (1-\alpha-\beta)(1-\epsilon) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| \\
&= 1-\alpha-\beta-\epsilon(1-\alpha-\beta) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} X_j^n \cdot A_{i,j} + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^n\right| \\
&\leq 1-\alpha-\beta-\epsilon(1-\alpha-\beta) + \frac{\alpha}{\sum_{j=0}^{m-1} X_j^n}\cdot\sum_{j=0}^{m-1} X_j^n + \frac{\beta}{m}\cdot\sum_{j=0}^{m-1} 1 \\
&\leq 1-\alpha-\beta-\epsilon(1-\alpha-\beta) + \alpha + \beta \\
&\leq 1-\epsilon(1-(\alpha+\beta)).
\end{aligned}$$

Since $0 \leq \alpha + \beta \leq 1$ and $\epsilon \geq 0$, we know $\epsilon(1 - (\alpha + \beta)) \geq 0$. So

$$\begin{aligned} X_i^{n+1} &\leq 1 - \epsilon(1 - (\alpha + \beta)) \\ &\leq 1. \end{aligned}$$

Therefore $0 < X_i^{n+1} \leq 1$. □

Neither we nor Walsh prove that the fixed point of the generalized version of PeerRank is unique, or that PeerRank will always converge to a fixed point. However, experimental results suggest that our choice of initial seed does not impact the fixed point.

### 3.3 Issues in the Classroom

There are several issues with the way PeerRank calculates grades that limit its usefulness in a classroom setting. First, recall

**Walsh's Assumption.** *An agent's accuracy in grading their peers is equal to their own grade.*

This means that PeerRank uses the graders' own grades as the weights when averaging the peer grades in the update step. However this assumption may not always hold for every grader. For example, suppose that we have a student who is having trouble with the course material at the time they complete their assignment. Their grade will likely be low, as they may not have understood how to answer certain questions. However, when they grade the assignments submitted by their classmates, they may see the correct answers, realize why they are correct, and gain a new understanding about the material. If this is true, then their grading accuracy will be higher than their own grade if we use the version of PeerRank in which $\beta = 0$, which is the same as the basic version. Walsh claims that this basic version still should provide accurate grades, and that the generalized version merely provides an incentive for graders to grade accurately. However, because PeerRank does not differentiate between one's grade and their grading abilities in its calculation of grades, the grades given by our student will have little impact on the final results, despite their accuracy. Therefore, PeerRank seems to ignore the potential for students to learn from their classmates, which is one of the major benefits of peer grading that we identified in Section 1. This flaw can

also be demonstrated by the reverse of this example. Now, suppose that we have a student who receives a high grade on their assignment, but grades inaccurately. In this case, if we once again let $\beta = 0$, this student's grade will be much higher than their grading accuracy, but they will still have a strong impact on the grades of their peers. As these examples illustrate, it may be beneficial to use some independent measure of a grader's accuracy as the weight in PeerRank, rather than the grader's own grade.

The other major issue with using PeerRank in the classroom can be demonstrated by the following example. Suppose that we have five students in a class, and those students are given an assignment that consists of a single true or false question. The students who give the correct answer should receive full credit (i.e. a grade of 1), and those who answer incorrectly should receive a grade of 0. Now, suppose that two of the students answer correctly, and the other three answer incorrectly. Assuming that each student believes themselves to be correct, we will have the following peer grading matrix:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Since we know which students are correct and incorrect, we would hope that the grades produced by our peer grading system would be $[1, 1, 0, 0, 0]^T$. However, because PeerRank has no knowledge of which students are correct, it simply favors the majority group and produces the grades $\vec{X} = [0, 0, 1, 1, 1]^T$. While this example seems rather extreme, it shows that extremely incorrect grades could be produced if the majority of the students in a class grade based on some fundamental misunderstanding of the material in an assignment. Clearly, in order to use PeerRank in a classroom, we must implement some method for specifying a basis of "correctness" for each assignment.

# 4 Proposed Changes

In this section, we present our proposed changes to PeerRank in order to address the issues described in Section 3.3. We start by explaining our goal as it relates to the concept of "ground truth" in education, and then we present our proposed solution.

## 4.1 Ground Truth

In education, we assume that there is a notion of ground truth in assignments that determines which submissions are considered "correct" or "incorrect". For example, in an elementary calculus course most problems have a single answer that is fundamentally correct, while all other answers are incorrect. This idea of ground truth can also be extended to courses with assignments that require answers that are not as clear-cut. In a math course based in proofs, a "correct" answer is a proof that uses given assumptions in order to prove a statement without any flaws in logic. Even with writing assignments there is often some sort of ground truth on which an essay is graded, which includes using proper grammar, writing a persuasive argument, using sources effectively, or meeting the guidelines set by a rubric.

The ground truth in an assignment is normally determined by the course's instructor when they grade each submission. For example, if a homework assignment consists of questions, the instructor provides the ground truth by marking each answer as either correct or incorrect. However, in PeerRank the instructor has no role as grading is left entirely to the students. Therefore, as stated in Section 3.3, PeerRank incorporates no factor reflecting a concept of ground truth. We wish to give the instructor a role in the PeerRank process that allows them to provide the basis of ground truth, and have each grader's weight in the grade calculation be a reflection of their grading accuracy in relation to the provided basis.

## 4.2 Our Proposed Solution

We now present our new peer grading process, which incorporates an independent measure of grading accuracy. First, the instructor submits and determines the correct grade for their own sample submission. The knowledge of which assignment belongs to the instructor and the correct grade is not shared with the

students so that each assignment will be treated equally in the grading process. The students will then grade each of the submitted assignments, including the instructor's. An example of this process, using the same example student as in Figure 1, is shown in Figure 2. We then compute an accuracy score $ACC_i$ for each grader $i$ by substituting both the grade $A_{I,i}$ given by $i$ to the instructor's submission and the correct grade $X_I$ provided by the instructor into the following equation:

$$ACC_i = 1 - |A_{I,i} - X_I| \tag{5}$$

Because both $A_{I,i}$ and $X_I$ are in the range $[0, 1]$, $ACC_i$ is also in the range $[0, 1]$. Note that this is a measure of accuracy analogous to that which Walsh uses in the $\beta$ term of the generalized version of PeerRank.

Next, we change the PeerRank equation so that these accuracy scores are used as the weights in grade calculation instead of the grader's grades. Therefore the update step of the basic version of PeerRank, represented by Equation 3, becomes

$$\vec{X}^{n+1} = (1 - \alpha) \cdot \vec{X}^n + \frac{\alpha}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC} \tag{6}$$

where $\overrightarrow{ACC}$ is the vector composed of the accuracy scores. Because the second term of this equation involves division by $\left|\left|\overrightarrow{ACC}\right|\right|_1$, our method does not work when $\overrightarrow{ACC}$ is equal to $\vec{0}$. However this is not a

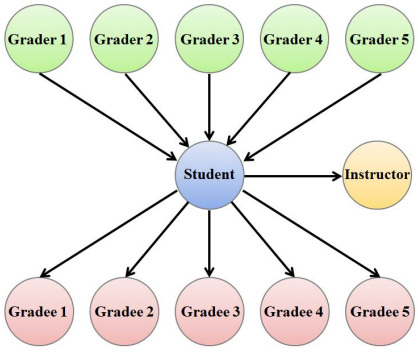



Figure 2: An example of one student's role in our new peer grading process. The student now grades the instructor's assignment in addition to the five other students.

major problem since if every grader has a grading accuracy of 0, then no peer grading system can produce the "correct" final grades. The issue can also be avoided entirely if the instructor's assignment is chosen such that $X_I \in (0,1)$. Therefore, we assume that $\overrightarrow{ACC} \neq \vec{0}$. Also, because we now know the fixed weights at the beginning of the grade calculation process we are no longer required to use an iterative process. We now prove that the iterative method inherited from PeerRank simplifies to produce the following grades in our setting:

$$\vec{X} = \frac{1}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC} \tag{7}$$

**Proposition 6.** *The fixed point of the iterative version of our process is equal to* $\frac{1}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC}$.

*Proof.* At the fixed point $\vec{X}$, we have

$$\begin{aligned}
\vec{X} &= (1-\alpha) \cdot \vec{X} + \frac{\alpha}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC}, \\
\vec{X} &= \vec{X} - \alpha\vec{X} + \frac{\alpha}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC}, \\
\alpha\vec{X} &= \frac{\alpha}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC}, \\
\vec{X} &= \frac{1}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC}.
\end{aligned}$$

□

Note that our method simply calculates weighted averages of the peer grades given to each student, where the weights used are the accuracy scores. We will use this "basic" version of our method, which does not include an equivalent of the incentive term used in the generalized version of PeerRank, throughout the rest of the paper. However, it is simple to include this incentive in our method by using a two-stage process. Choose a value of $\beta$ where $0 \leq \beta \leq 1$, which represents the portion of a student's grade that should be determined by their accuracy. First we calculate an initial grade vector $\vec{X}^0$ using Equation 7, which reflects the grades deserved based solely on the quality of each student's submission. Then, we recalculate each student's grade $X_i$ by including a factor of $\beta$ based on the student's accuracy in grading

the submissions of others, just like Walsh does in his generalized version of PeerRank. So our two stages are:

$$\vec{X}^0 = \frac{1}{\left|\left|\overrightarrow{ACC}\right|\right|_1} \cdot A \cdot \overrightarrow{ACC} \tag{8}$$

and

$$X_i = (1-\beta) \cdot X_i^0 + \frac{\beta}{m} \cdot \sum_{j=0}^{m-1} 1 - \left|A_{j,i} - X_j^0\right|. \tag{9}$$

By incorporating the grader's accuracy in grading all of the assignments, rather than just the instructor's submission, into our incentive term we encourage the graders to grade all of the assignments with an equal level of effort.

As we stated in Section 2, the method by which we determine each grader's accuracy score was heavily inspired by the work of Piech *et. al* [2]. Recall that in their method of determining grading accuracy, each grader was required to grade a submission from a small set of student assignments. Since their method was used in MOOCs, which involve extremely large classes, each of these "ground truth" submissions was graded hundreds of times, and so the average of the grades received by a ground truth submission was assumed to be the correct grade by the law of large numbers. Each grader's accuracy in grading these submissions, when compared to the determined ground truth grade, was then used to determine the grader's accuracy score. This process is extremely similar to our process, in which a grader's accuracy score is determined by their accuracy in grading an instructor's submission with a previously determined grade.

The main difference between the method used by Peich *et. al* and our method is the manner in which the grade for the "ground truth submission" is determined. In their method the correct grade for the assignment is determined by the group of grading students, while in our method the grade is determined by the instructor. This difference in how ground truth is determined provides our system with two key advantages over that of Piech *et. al*. First, their system required the ground truth submissions to be graded hundreds of times in order to apply the law of large numbers, which means that their system can only be used in extremely large classes. Our system, however, can be applied to smaller classes since the correct grade for the ground truth submission is determined solely by the instructor. Second, Piech *et. al*'s application of the law of large numbers assumes that the class contains more accurate graders than inaccurate graders. This

opens their system up to one of the problems with PeerRank, as stated in Section 3.3. If the majority of the students in a class have a fundamental misunderstanding of the material then the grade given to the ground truth submission will be inaccurate, as it will reflect that common misunderstanding. However, as we will explain in Section 4.3, our method's use of the instructor to provide a basis of correctness addresses this issue.

## 4.3 Addressed Issues

The solution we have presented addresses Issues 1 and 2 from Section 1. Because we generate accuracy scores and use those scores to properly weight the peer grades, we anticipate the potential for inaccurate graders and make the grades from those graders count less towards the final grades, addressing Issue 1. Also, the version of our method presented in Equations 8 and 9 provide graders with an incentive to grade accurately, addressing Issue 2. In addition, our solution addresses the two main issues with PeerRank described in Section 3.3. First, we no longer follow Walsh's Assumption that a grader's accuracy is equal to their own grade. We instead calculate an independent score reflecting their accuracy by evaluating their performance in grading a sample assignment. Second, by having the instructor create both a sample submission and the correct grade for it, we allow the instructor to provide a basis of correctness for the assignment. Each grader's accuracy is then determined in relation to this basis, which allows our method to determine who the correct graders are. While it is possible that a student could use this fact in an attempt to manipulate their own accuracy score if they are able to identify the instructor's assignment, we assume that the instructor remains anonymous. It is also likely that such attempts would fail since the correct grade for the submission is withheld from the students.

These changes can be demonstrated by returning to the "majority vs. minority" example from Section 3.3. Recall that in this example, we had a single question assignment, two students submitted a correct assignment, and three students submitted an incorrect assignment. Without a way of specifying a basis of correctness, PeerRank produced the grades $\vec{X} = [0, 0, 1, 1, 1]^T$ instead of the correct grades $[1, 1, 0, 0, 0]^T$.

Now, suppose that the instructor submits a correct assignment and gives it a grade of 1. This means that

we will have the following peer grade matrix

$$A' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & - \\ 1 & 1 & 0 & 0 & 0 & - \\ 0 & 0 & 1 & 1 & 1 & - \\ 0 & 0 & 1 & 1 & 1 & - \\ 0 & 0 & 1 & 1 & 1 & - \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where the new row contains the grades given to the instructor's assignment. Because the instructor only grades themselves, the missing grades in the last column are marked with a dash. Using our method of generating accuracy scores, we obtain $\overrightarrow{ACC} = [1, 1, 0, 0, 0, 1]^T$. This means that only the two correct students will have an impact on the grade calculations, since the incorrect students each have an accuracy of 0. Now, substituting the grade matrix

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

and $\overrightarrow{ACC}$ into Equation 7 we obtain the final grades $\vec{X} = [1, 1, 0, 0, 0]^T$, which is the correct solution.

# 5 Evaluation

We tested our proposed peer grading scheme against grade data to see how its performance compares against the basic version of PeerRank. In this section, we first discuss how we implemented the two systems and our test procedures. We then present the procedure with which we simulated peer grading data using statistical models. Finally, we explain our experiments and present our results.

## 5.1 Implementation

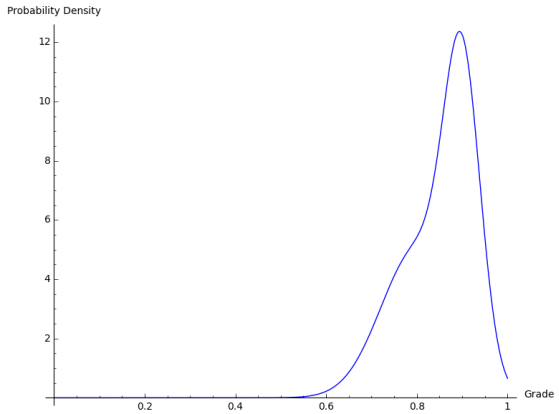
We implemented both PeerRank and our method using Sage [9], an open-source programming language based on Python that incorporates various additional mathematics packages and operations. These added operations include the ability to represent and perform operations on matrices and vectors, which are essential in implementing the PeerRank system. We also used Sage to create various automated test procedures that we used in our experiments. The code used for each of these tasks is given in Appendix A.

## 5.2 Simulated Data

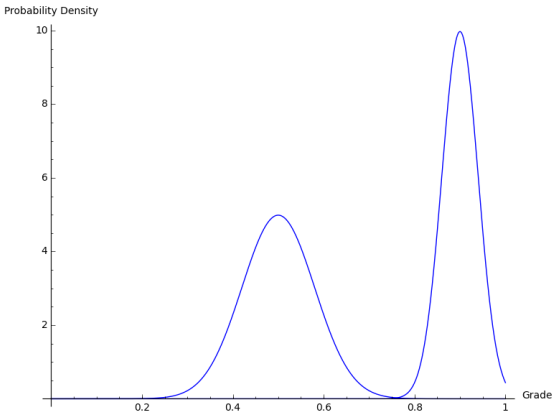
In order to conduct our experiments, we required grade data with which we could test the accuracies of the two peer grading methods. However, real peer grading data from actual classes is not easily available, and the potential data sources that were available to us did not meet the requirement that each student in the class grades all of the other students. Therefore, we chose to use simulated data based on statistical models based on the advice of Professor Roger Hoerl, a statistician in the Union College Mathematics Department [5]. When creating the correct grades that a student should receive based on a ground truth, we used a bimodal distribution comprised of two normal distributions. A *normal distribution*, often called a "bell curve", is a symmetric probability distribution which is centered at a given mean, or average value, and approaches zero on both sides of the mean. In other words, a value close to the mean has a high chance of being selected, while a value is less likely to be selected the farther it is from the mean. The graph of a normal distribution is defined by the function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where $\mu$ is the man value and $\sigma$ is the *standard deviation*, a measure of the dispersion of the values in the distribution from the mean. One of the normal distributions constituting our bimodal distribution represented a group of high-achieving students, and had a high mean grade and low standard deviation. The other normal distribution represented a group of lower-achieving students, and had a lower mean grade and higher standard deviation. The two means and standard deviations, as well as the percentage




(a) Graph of a distribution using values suggested by Professor Roger Hoerl [5]. The high-achieving distribution has a mean of 0.9 and a standard deviation of 0.04. The low-achieving distribution has a mean of 0.8 and a standard deviation of 0.08.




(b) Graph of the distribution used in the experiments described in Section 5.4, which contains a greater division between the high-achieving and low-achieving students. The high-achieving distribution has a mean of 0.9 and a standard deviation of 0.04. The low-achieving distribution has a mean of 0.5 and a standard deviation of 0.08.

Figure 3: Graphs of two different student distributions. Any grades produced by these distributions that are outside the range $[0, 1]$ are "clipped" to the bounds of the range.

of students drawn from each distribution, were parameters in our experiment. Two example distributions are shown in Figure 3. If any of the grades drawn were outside the $[0, 1]$ range required by the grading systems, they were "clipped" to the bounds of the range. The resulting correct grades are assembled into a grade vector $\vec{G}$. The pseudocode for this grade generation process is given in Figure 4.

Next, we must generate the accuracy scores which our method produces through each grader's grading of the instructor's sample submission. We simulate this process by drawing the accuracy score for each

1: $strongStudentNum \leftarrow \lceil strongStudentPercentage * classSize \rceil$
2: **for** $i \leftarrow 0$ **to** $strongStudentNum$ **do**
3: $\quad G[i] \leftarrow \text{NORMALDISTRIBUTION}(strongMean, strongStdev)$
4: $\quad G[i] \leftarrow \min(\max(G[i], 0), 1)$
5: **for** $i$ **to** $classSize$ **do**
6: $\quad G[i] \leftarrow \text{NORMALDISTRIBUTION}(weakMean, weakStdev)$
7: $\quad G[i] \leftarrow \min(\max(G[i], 0), 1)$

Figure 4: Pseudocode for the generation of correct grades.

grader from a normal distribution with a mean equal to the grader's grade and a fixed standard deviation selected as a parameter in our experiment. Once again, if we draw an accuracy score outside the range $[0, 1]$, we clip it at the nearest bound. We treat the resulting accuracy scores as if they were the values produced through the grading of the instructor's submission, and so we assemble the scores into the vector $\overrightarrow{ACC}$. The pseudocode for this process is given in Figure 5. Note that when the standard deviation is equal to 0, we always draw an accuracy score equal to the grader's grade, satisfying Walsh's Assumption. However, as we increase the standard deviation we can draw from a wider range of accuracy scores, meaning that we relax Walsh's Assumption as we increase the standard deviation. This will be an important fact in our experiments.

Finally we simulate the peer grades assigned by each grader to each student using a *uniform distribution*, a probability distribution in which every value in a given range has an equal chance of being selected, since we assume that if a grader has a certain amount of inaccuracy then they will produce peer grades from within a certain error range with equal likelihood. For each grader-student pair, the bounds of the uniform distribution are defined as

$$G_i \pm (1 - ACC_j) \cdot G_i$$

where $G_i$ is the correct grade for student $i$ and $ACC_j$ is the accuracy score for grader $j$. We then draw a grade from the distribution, and apply the same clipping procedure if the resulting grade is outside the $[0, 1]$ bound. The peer grades that we generate are assembled into a matrix $A$. The pseudocode for this process is given in Figure 6.

## 5.3 Experimental Method

In our experiments, we selected the number of different classes to simulate, the size of the classes, the means and standard distributions used to draw the correct grades each student should receive, the percentage of

1: **for** $i \leftarrow 0$ **to** $classSize$ **do**
2: $\quad ACC[i] \leftarrow \text{NORMALDISTRIBUTION}(G[i], accStdev)$
3: $\quad ACC[i] \leftarrow \min(\max(ACC[i], 0), 1)$

Figure 5: Pseudocode for the generation of accuracy scores.

the class that was in each of the two groups, and the standard deviation for the normal distributions used to draw accuracy scores. For each test, we independently generated the correct grades, accuracy scores, and peer grades for a class using the process described in Section 5.2. We then used the peer grade matrix $A$ to calculate grades from PeerRank, and used $A$ and the accuracy vector $\overrightarrow{ACC}$ to calculate grades using our method. We then compared each of the two resulting vectors to the correct grade vector $\vec{G}$. In order to generate an average error for each of the two methods, we subtracted $\vec{G}$ from the grade vector $\vec{X}$, took the *two norm* (or Euclidean distance), of the difference, and then divided the result by the square root of the class size. This means that we are left with two average error values for test $t$:

$$E^t_{PR} = \frac{\left|\left|\vec{X}^t_{PR} - \vec{G}^t\right|\right|_2}{\sqrt{m}} \tag{10}$$

$$E^t_{Acc} = \frac{\left|\left|\vec{X}^t_{Acc} - \vec{G}^t\right|\right|_2}{\sqrt{m}} \tag{11}$$

where $E^t_{PR}$ is the average error resulting from PeerRank, $E^t_{Acc}$ is the average error resulting from our method, and $\left|\left|\vec{V}\right|\right| = \sqrt{\sum_i v_i^2}$ is the two norm of $\vec{V}$.

After all $numTests$ tests have been completed, we average both the errors from PeerRank and the errors from our method in order to obtain the average errors over all of the tests.

$$E_{PR} = \frac{1}{numTests} \sum_{t=0}^{numTests-1} E^t_{PR} \tag{12}$$

$$E_{Acc} = \frac{1}{numTests} \sum_{t=0}^{numTests-1} E^t_{Acc} \tag{13}$$

1: **for** $j \leftarrow 0$ **to** $classSize$ **do**
2:  **for** $i \leftarrow 0$ **to** $classSize$ **do**
3:  $min \leftarrow G[i] - (1 - ACC[j]) * G[i]$
4:  $max \leftarrow G[i] + (1 - ACC[j]) * G[i]$
5:  $A[i, j] \leftarrow$ UNIFORMDISTRIBUTION$(min, max)$
6:  $A[i, j] \leftarrow \min(\max(A[i, j], 0), 1)$

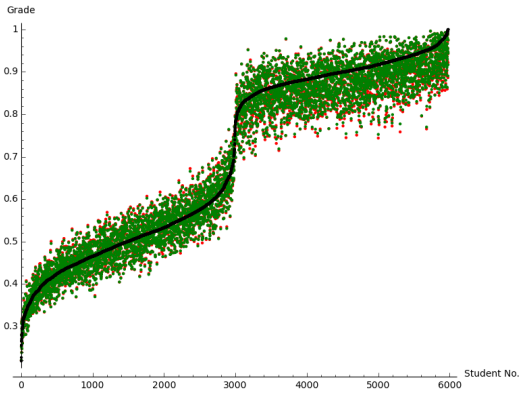
Figure 6: Pseudocode for the generation of peer grades.

## 5.4 Results

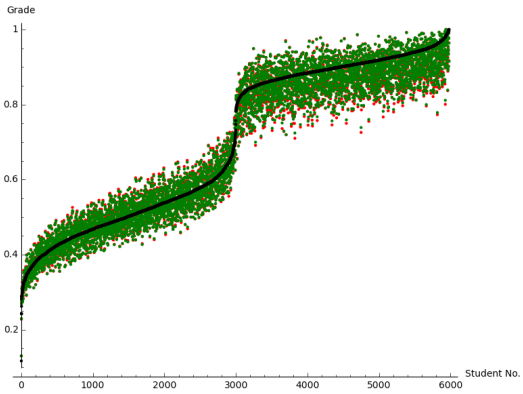
Our goal was to see how our method's performance compares again that of PeerRank on simulated grade data. We conducted two separate experiments with different class sizes. In Experiment 1, we tested 1,000 different small classes of 6 students, and in Experiment 2 we tested 120 different large classes of 50 students. The number of different classes to use in each experiment were chosen so that the two experiments would both test the same total number of students (6,000). We were most interested in how relaxing Walsh's Assumption that a grader's accuracy is equal to their own grade impacted the relative errors of both methods. Therefore, the only variable we changed between tests in each experiment was the standard deviation used in the normal distributions that generate accuracy scores. For the other variables, we let the number of classes and the class size be fixed at the values defined for the experiment, the mean grade and standard distribution for the higher-achieving group be 0.9 and 0.04 respectively, the mean grade and standard distribution for the lower-achieving group be 0.5 and 0.08 respectively, and the percentage of students in each group be 50%. These values, which may not accurately represent the average real-world class, were chosen so that our simulated classes would have an even mix of strong students and poor students.

For our experiments, we started by following Walsh's Assumption and letting each grader's accuracy be equal to their own grade (i.e. we used a standard deviation of 0). We then increased the accuracy standard deviation to 0.02 and 0.10. Recall that as the standard deviation increases, the connection between a grader's accuracy and their grade decreases, so these small increases in the standard deviation allow us to judge how well the two methods perform as we slightly relax Walsh's Assumption. Finally, in order to test the methods' performance in cases that completely violate Walsh's Assumption, we increased the standard deviation to 0.50 and 1.00. In Figures 7 and 8 we graph the grades resulting from PeerRank and our method alongside the correct grades. In Tables 1 and 2, we present the standard deviations, the average absolute errors of PeerRank and our method, and the average improvement achieved by using our method.

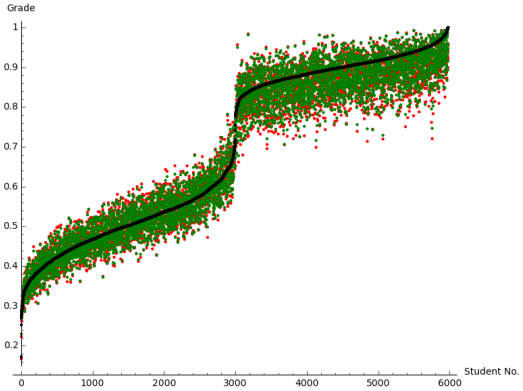
Notice that for low standard deviations, where Walsh's Assumption holds, the grades produced by our method are approximately equal to those produced by PeerRank. In Figures 7 and 8 (a) and (b) the range of grades produced by PeerRank is the approximately the same as the range of grades produced by our method, while in Figures 7 and 8 (c) the range of grades produced by PeerRank is only slightly larger than



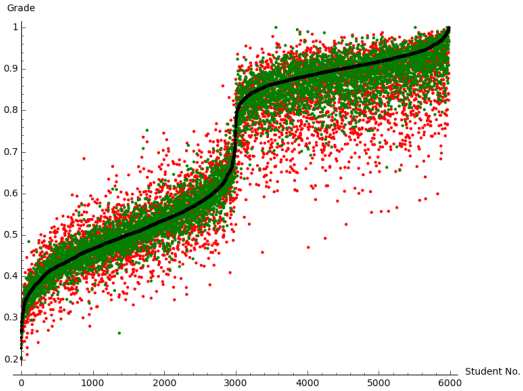

(a) Standard deviation = 0.00



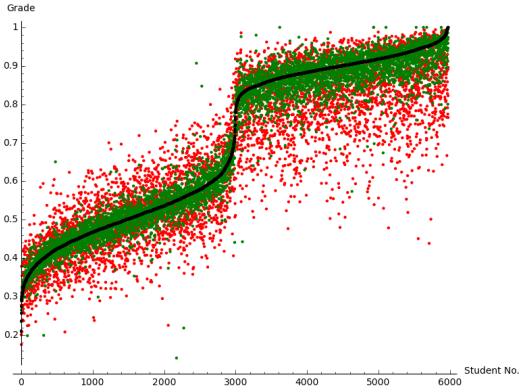

(b) Standard deviation = 0.02




(c) Standard deviation = 0.10

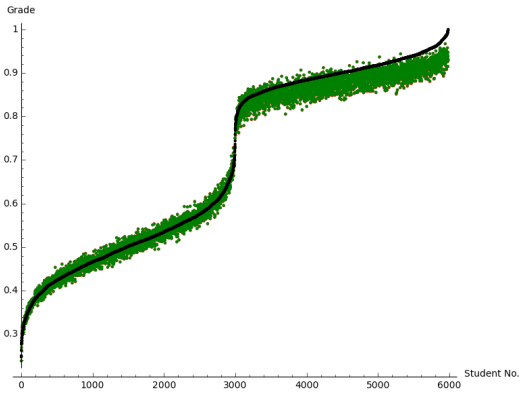



(d) Standard deviation = 0.50

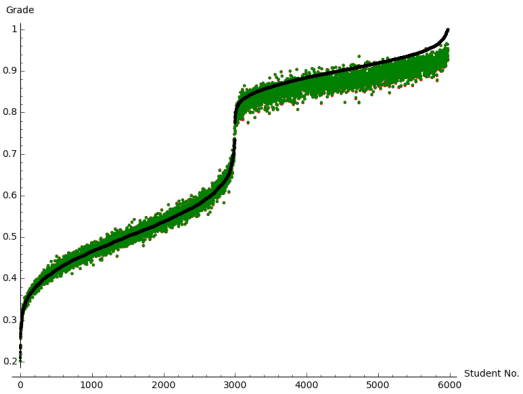



(e) Standard deviation = 1.00

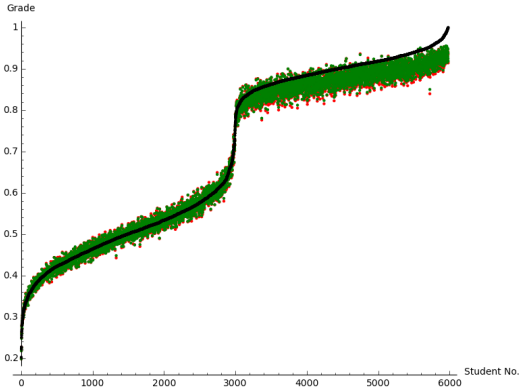
Figure 7: The graphs resulting from Experiment 1 with varying distributions. The black line represents the correct grades, the green area represents the grades produced by our method, and the red area represents the grades produced by PeerRank. In each graph, the green area covers a portion of the red area. The proximity of the edges of each of the two areas to the black line of correct grades can be interpreted as a measure of each system's accuracy.



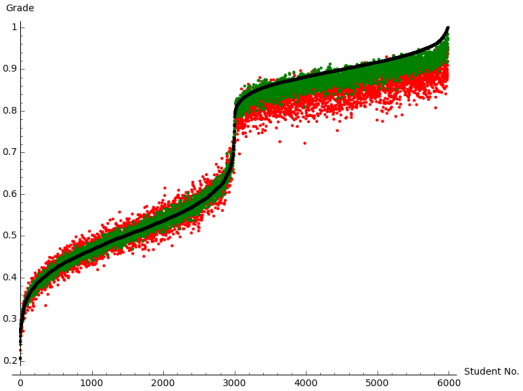

(a) Standard deviation = 0.00



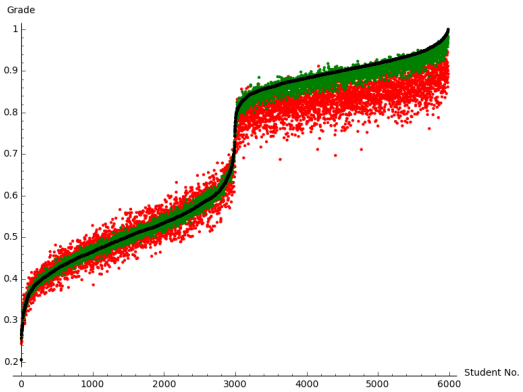

(b) Standard deviation = 0.02




(c) Standard deviation = 0.10




(d) Standard deviation = 0.50




(e) Standard deviation = 1.00

Figure 8: The graphs resulting from Experiment 2 with varying distributions. The black line represents the correct grades, the green area represents the grades produced by our method, and the red area represents the grades produced by PeerRank. In each graph, the green area covers a portion of the red area. The proximity of the edges of each of the two areas to the black line of correct grades can be interpreted as a measure of each system's accuracy.

| Standard Deviation | PeerRank Error | Our Method's Error | Improvement |
|---|---|---|---|
| 0.00 | 0.0399 | 0.0392 | 0.0007 |
| 0.02 | 0.0404 | 0.0394 | 0.0010 |
| 0.10 | 0.0426 | 0.0397 | 0.0029 |
| 0.50 | 0.0669 | 0.0363 | 0.0306 |
| 1.00 | 0.0813 | 0.0287 | 0.0526 |

Table 1: A table of results from Experiment 1, containing the average errors from PeerRank and our method, as well as the average improvement from our method, for each of the tested standard deviations.

| Standard Deviation | PeerRank Error | Our Method's Error | Improvement |
|---|---|---|---|
| 0.00 | 0.0231 | 0.0228 | 0.0003 |
| 0.02 | 0.0233 | 0.0229 | 0.0004 |
| 0.10 | 0.0252 | 0.0233 | 0.0019 |
| 0.50 | 0.0417 | 0.0184 | 0.0233 |
| 1.00 | 0.0532 | 0.0133 | 0.0399 |

Table 2: A table of results from Experiment 2, containing the average errors from PeerRank and our method, as well as the average improvement from our method, for each of the tested standard deviations.

the range produced by our method. Because of this, the average errors produced by the two methods are approximately equal in these cases, as shown by the first three rows of Tables 1 and 2. However, Figures 7 and 8 (d) and (e) show that as the standard deviation becomes larger and we break away from Walsh's Assumption, our method produces approximately the same range of grades (i.e., the green area in the graphs remain the same distance from the black line representing the correct grades), while the range produced by PeerRank grows extremely large. This is also reflected in Tables 1 and 2, which shows that when the standard deviation is 1, we achieve an average improvement in accuracy of of 0.05 for small classes and 0.04 for large classes (which is, for example, the difference between a percent grade of 90% or an A- and a percent grade of 86% or a B) by using our method. Also, while the results show that both methods produce higher absolute errors for smaller classes than for larger classes, both class sizes demonstrate the same pattern of improvement from using our method as the standard deviation increases. This suggests that these results hold regardless of the number of students in the class.

# 6 Conclusion

As our results on the simulated data demonstrate, if Walsh's Assumption that a grader's accuracy is equal to their grade holds to be true, our proposed method produces grades that are approximately equal to those produced by PeerRank. This is because we can simply replace the accuracy weights used in Equation 7 by the graders' grades, following Walsh's Assumption, and end up with a grade vector that is approximately equal to PeerRank's fixed point. As a result, we can see that our method performs no worse than PeerRank. However in certain cases where Walsh's Assumption is false and a grader's accuracy is entirely independent of their own grade, a potential we presented in Section 3.3, our method produces grades that are more accurate than those produced by PeerRank. This occurs because our method assumes no connection between grade and accuracy and therefore is unaffected in cases where a grader's accuracy is very different from their grade. PeerRank on the other hand assumes an explicit connection between grade and accuracy, and therefore its performance suffers when this is not true.

In conclusion, by rejecting Walsh's Assumption about the grade-accuracy connection and allowing an instructor to provide a basis of ground truth, our method seems to address the issues found in PeerRank, and therefore would allow for more accurate peer grading in a classroom setting.

# 7 Future Work

There are several different directions in which this research can be extended and continued. One immediate continuation would be to test the performance of both PeerRank and our proposed method in actual classrooms with actual students and instructors, in order to see whether the results of our tests on simulated data hold in practice. Another possible extension would be to propose additional methods of integrating ground truth into PeerRank, and then compare the performance of those methods against both PeerRank and our method. For example, instead of having all students grade an instructor's assignment, the instructor could personally grade a certain subset of the students in the class. The instructor's "grade" could then be fixed at a large value so that the grades they provide would be the main determining factor for those students' grades. The increased accuracy in those students' grades would then have an effect on the grades

of the students they graded, and so the accuracy provided by the instructor could propagate throughout the class.

This additional proposal of ground truth integration connects to an additional extension of this work, which is to modify PeerRank so that each agent grades only a subset of the class. In both Walsh's proposal of PeerRank and our work, we assume that each student in the class provides grades for the entire class, i.e. if there are $m$ students in the class then each student must grade $m$ assignments (or $m + 1$ assignments if we include an instructor's submission), for a total of $m^2$ grades. While this may be a reasonable proposal for an extremely small class, it becomes infeasible with even a twenty student class, which is considered small by the standards of most universities. Therefore it would be helpful to implement a "partial" grading scheme, in which each student only grades a small subset of their peers, rather than the entire class. A basic solution to this problem would be to simply use whatever grades were received for an assignment in order to determine the assignment's final grade, and to ignore the other students in the class. However this could present challenges since, if an assignment is only graded by inaccurate graders, there is a greater chance of error in the assignment's final grade. This raises additional questions of whether there is an ideal way to assign graders to assignments, or whether there is a clever method of approximating the grades that might be given to an assignment by graders who did not directly grade that assignment based solely on the known grades and accuracies.

# 8 Acknowledgments

I would like to thank my two advisors, Professor Matthew Anderson of the Computer Science department and Professor William Zwicker of the Mathematics department, for all of their help over the course of this project. Their knowledge and guidance have been essential in this project, and I truly appreciate all they have done. I would also like to thank Professor Roger Hoerl of the Mathematics department for his guidance in determining the proper distributions with which to model the grade data. Finally, I would like to thank the professors in both the Computer Science and Mathematics departments for their support not only in this thesis, but throughout my undergraduate career at Union College.

# A Sage Code

In this appendix we provide our implementations of the algorithms we used in our research. As we stated in Section 5.1, all of the code is written in the Sage programming language [9].

## A.1 Basic Version of PeerRank

```
# Implementation of the basic PeerRank rule.
# Variable names follow from those given by Walsh.
# A is the initial grade matrix and 0<alpha<1
def BasicPeerRank(A, alpha):
    m = A.nrows()
    Xlist = [0]*m
    for i in range(0, m):
        sum = 0.0
        for j in range(0, m):
            sum += A[i,j]
        X_i = sum/m
        Xlist[i] = X_i
    X = vector(Xlist)

    fixedpoint = False
    while not fixedpoint:
        oldX = X
        X = (1-alpha)*X + (alpha/X.norm(1))*(A*X)
        difference = X - oldX
        if abs(difference) < 10**-10:
            fixedpoint = True
    return X
```

## A.2 Generalized Version of PeerRank

```
# Implementation of the generalized PeerRank rule.
# Variable names follow from those given by Walsh.
# A is the initial grade matrix and alpha+beta<=1
def GeneralPeerRank(A, alpha, beta):
    m = A.nrows()
    Xlist = [0]*m
    for i in range(0, m):
        sum = 0.0
        for j in range(0, m):
            sum += A[i,j]
        X_i = sum/m
        Xlist[i] = X_i
    X = vector(Xlist)

    fixedpoint = False
    while not fixedpoint:
        oldX = X
        X = (1-alpha-beta)*X + (alpha/X.norm(1))*(A*X)
        for i in range(0, m):
            X[i] += beta - (beta/m)*((A.column(i)-oldX).norm(1))
        difference = X - oldX
        if abs(difference) < 10**-10:
            fixedpoint = True
    return X
```

## A.3 Basic Version of Our Method

```
# Implementation of the basic version of our method.
# A is the initial grade matrix and ACC is the vector of accuracy scores
```

```
def BasicProposedMethodWithAccuracies(A, ACC):
    return (1/ACC.norm(1))*(A*ACC)
```

## A.4 Generalized Version of Our Method

```
# Implementation of the basic version of our method.
# A is the initial grade matrix, ACC is the vector of accuracy scores, and 0<beta<1
def GeneralProposedMethodWithAccuracies(A, ACC, beta):
    m = A.nrows()
    X0 = (1/ACC.norm(1))*(A*ACC)
    X = vector([0.0]*m)
    for i in range(0,m):
        for j in range(0,m):
            X[i] += 1-abs(A[j,i]-X0[j])
        X[i] = (1-beta)*X0[i]+(beta/m)*X[i]
    return X
```

## A.5 Experimental Comparison of PeerRank and Our Method

```
# Runs multiple tests on simulated data and outputs the average errors.
# numTests - number of tests to run
# classSize - size of class
# highMean - mean grade for "strong" students
# highStdev - standard deviation for "strong" student distribution
# lowMean - mean grade for "weaker" students
# lowStdev - standard deviation for "weaker" student distribution
# highPercentage - percentage of students (between 0 and 1) that are in "strong" distribution
# accStdev - standard deviation around grade from which to draw accuracy score
def testGroundTruth(numTests, classSize, highMean, highStdev, lowMean, lowStdev, highPercentage, \
            accStdev):
    strongStudentNum = math.ceil(highPercentage*classSize)
```

```
weakStudentNum = classSize-strongStudentNum
AccMethodDict = {}        #Maps correct grade to the grade produced using the accuracy model
PRDict = {}        #Maps correct grade to the grade produced using basic PeerRank
accTwoNormSum = 0.0
prTwoNormSum = 0.0
for test in range(0, numTests):
    actualGrades = [0.0]*classSize
    i = 0
    while i<strongStudentNum:
        actualGrades[i] = random.gauss(highMean,highStdev)
        if actualGrades[i] > 1:
            actualGrades[i] = 1
        elif actualGrades[i] < 0:
            actualGrades[i] = 0
        i = i+1
    while i<classSize:
        actualGrades[i] = random.gauss(lowMean, lowStdev)
        if actualGrades[i] > 1:
            actualGrades[i] = 1
        elif actualGrades[i] < 0:
            actualGrades[i] = 0
        i = i+1
    actualGrades = vector(actualGrades)    #actualGrades contains the students' correct grades

    accuracies = [0.0]*classSize
    for i in range(0,classSize):
        accuracies[i] = random.gauss(actualGrades[i],accStdev)
        if accuracies[i] < 0:
            accuracies[i] = 0
        elif accuracies[i] > 1:
```

```
            accuracies[i] = 1
        accuracies = vector(accuracies)        #accuracies contains the graders' accuracy scores

        A = Matrix([[0.0]*classSize]*classSize)        #A is the matrix containing the peer grades
        for j in range(0,classSize):
            for i in range(0,classSize):
                min = actualGrades[i]-(1-accuracies[j])*actualGrades[i]
                max = actualGrades[i]+(1-accuracies[j])*actualGrades[i]
                A[i,j] = random.uniform(min,max)
                if A[i,j] > 1:
                    A[i,j] = 1
                elif A[i,j] < 0:
                    A[i,j] = 0

        accOutput = BasicProposedMethodWithAccuracies(A,accuracies,0.1)
                    #accOutput is the set of grades produced by our method
        prOutput = BasicPeerRank(A,0.1)        #prOutput is the set of grades produced by PeerRank
        accDiff = actualGrades-accOutput
        prDiff = actualGrades-prOutput

        for i in range(0,classSize):
            AccMethodDict[actualGrades[i]] = accOutput[i]  #Accumulate our method's points on graph
            PRDict[actualGrades[i]] = prOutput[i]        #Accumulate PeerRank's points on graph
        accTwoNorm = accDiff.norm(2)/sqrt(classSize)
        accTwoNormSum += accTwoNorm        #Accumulate 2-norm errors for our method
        prTwoNorm = prDiff.norm(2)/sqrt(classSize)
        prTwoNormSum += prTwoNorm        #Accumulate 2-norm errors for PeerRank

    gtGrades = sorted(AccMethodDict)
    gtPoints = [None]*len(gtGrades)
```

```
accPoints = [None]*len(gtGrades)
prPoints = [None]*len(gtGrades)
for i in range(0,len(gtGrades)):        #Create lists of points for graph
    gtPoints[i] = (i,gtGrades[i])
    accPoints[i] = (i,AccMethodDict[gtGrades[i]])
    prPoints[i] = (i,PRDict[gtGrades[i]])
gtPoints = point(gtPoints, rgbcolor='black')
accPoints = point(accPoints, rgbcolor='green')
prPoints = point(prPoints, rgbcolor='red')
show(prPoints+accPoints+gtPoints, axes_labels=['Student No.','Grade'])        #Graph points

print "Average Error by Two Norm for Our Method: " + str(accTwoNormSum/numTests)
print "Average Error by Two Norm for PeerRank: " + str(prTwoNormSum/numTests)
```

# References


[1] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *SIAM Review*, 48(3):569–581, January 2006.

[2] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in MOOCs. In R.A. Calvo S.K. D'Mello and A. Olney, editors, *Proceedings of the 6th International Conference on Educational Data Mining*, pages 153–160, 2013.

[3] K. Cho and C.D. Schunn. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Comput. Educ.*, 48(3):409–426, April 2007.

[4] L. de Alfaro and M. Shavlovsky. CrowdGrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 415–420, 2014.

[5] R. Hoerl. Personal communication, February 2015.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[7] M.R. Merrifield and D.G. Saari. Telescope time without tears: A distributed approach to peer review. *Astronomy & Geophysics*, 50(4):4.16–4.20, 2009.

[8] P.M. Sadler and E. Good. The impact of self- and peer-grading on student learning. *Educational Assessment*, 11(1):1–31, 2006.

[9] SageMath. Sage 6.3. `http://www.sagemath.org/`, 2014.

[10] K. Topping. Peer assessment between students in colleges and universities. *Review of Educational Research*, 68(3):249–276, 1998.

[11] T. Walsh. The PeerRank method for peer assessment. In *ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 909–914, 2014.