Predicting Grades

Yannick Meier, Jie Xu, Onur Atan, and Mihaela van der Schaar, *Fellow, IEEE*

Abstract—To increase efficacy in traditional classroom courses as well as in Massive Open Online Courses (MOOCs), automated systems supporting the instructor are needed. One important problem is to automatically detect students that are going to do poorly in a course early enough to be able to take remedial actions. Existing grade prediction systems focus on maximizing the accuracy of the prediction while overseeing the importance of issuing timely and personalized predictions. This paper proposes an algorithm that predicts the final grade of each student in a class. It issues a prediction for each student individually, when the expected accuracy of the prediction is sufficient. The algorithm learns online what is the optimal prediction and time to issue a prediction based on past history of students' performance in a course. We derive a confidence estimate for the prediction accuracy and demonstrate the performance of our algorithm on a dataset obtained based on the performance of approximately 700 UCLA undergraduate students who have taken an introductory digital signal processing over the past seven years. We demonstrate that for 85% of the students we can predict with 76% accuracy whether they are going do well or poorly in the class after the fourth course week. Using data obtained from a pilot course, our methodology suggests that it is effective to perform early in-class assessments such as quizzes, which result in timely performance prediction for each student, thereby enabling timely interventions by the instructor (at the student or class level) when necessary.

Index Terms—Forecasting algorithms, online learning, grade prediction, data mining, digital signal processing education.

I. INTRODUCTION

EDUCATION is in a transformation phase; knowledge is increasingly becoming freely accessible to everyone (through Massive Open Online Courses, Wikipedia, etc.) and is developed by a large number of contributors rather than by a single author [1]. Furthermore, new technology allows for personalized education enabling students to learn more efficiently and giving teachers the tools to support each student individually if needed, even if the class is large [2].

Grades are supposed to summarize in a single number or letter how well a student was able to understand and apply the knowledge conveyed in a course. Thus it is crucial for students to obtain the necessary support to pass and do well in a class. How-ever, with large class sizes at universities and even larger class sizes in Massive Open Online Courses (MOOCs), which have undergone a rapid development in the past few years, it has become impossible for the instructor and teaching assistants to keep track of the performance of each student individually. This can lead to students failing in a class who could have passed if appropriate remedial actions had been taken early enough or excellent students not receiving the necessary promotion to benefit maximally from the course. Remedial or promotional actions could consist of additional online study material presented to the student in a personalized and/or automated manner [3]. Hence, in both offline and online education, it is of great importance to develop automated personalized systems that predict the performance of a student in a course before the course is over and as soon as possible. While in online teaching systems a variety of data about a student such as responses to quizzes, activity in the forum and study time can be collected, the available data in a practical offline setting are limited to scores in early performance assessments such as homework assignments, quizzes and midterm exams.

In this paper we focus on predicting grades in traditional classroom-teaching where only the scores of students from past performance assessments are available. However, we believe that our methods can also be applied for online courses such as MOOCs. We design a grade prediction algorithm that finds for each student the best time to predict his/her grade such that, based on this prediction, a timely intervention can be made if necessary. Note that we analyze data from a digital signal processing course where no interventions were made; hence, we do not study the impact of inventions and consider only a single grade prediction for each student. However, our algorithm can be easily extended to multiple predictions per student.

A timely prediction exclusively based on the limited data from the course itself is challenging for various reasons. First, since at the beginning most students are motivated, the score of students in early performance assessments (e.g., homework assignments) might have little correlation with their score in later performance assessments, in-class exams and the overall score. Second, even if the same material is covered in each year of the course, the assignments and exams change every year. Therefore, the informativeness of particular assignments with regard to predicting the final grade may change over the years. Third, the predictability of students having a variety of different backgrounds is very diverse. For some students an accurate prediction can be made very early based on the first few performance assessments. If for example a student shows an excellent performance in the first three homework assignments and in the midterm exam, it is highly likely that he/she will pass the class. For other students it might take more time to make an equally accurate prediction. If a student for example performs below average but not terribly at the beginning, it is risky to predict

whether he/she is going to pass or fail and, therefore, to decide whether or not to intervene. This third challenge illustrates the necessity to make the prediction for each student individually and not for all at the same time.

The main contributions of this paper can be summarized as follows.

1) We propose an algorithm that makes a personalized and timely prediction of the grade of each student in a class. The algorithm can both be used in regression settings, where the overall score is predicted, and in classification settings, where the students are classified into two (e.g., do well/poorly) or more categories.

2) We accompany each prediction with a confidence estimate indicating the expected accuracy of the prediction.

3) We derive a bound for the probability that the prediction error is larger than a desired value ϵ.

4) We exclusively use the scores students achieve in early performance assessments such as homework assignments and midterm exams and do not use any other information such as age, gender or previous GPA. This makes our algorithm applicable in all practical traditional classroom and online teaching settings, where such information may not be available.

5) Since the algorithm is learning from past years, the predictions become more accurate when more data from previous years become available.

6) We demonstrate that the algorithm shows good robustness if different instructors have taught the course in past years.

7) We analyze real data from an introductory digital signal processing course taught at UCLA over 7 years and use the data to experimentally demonstrate the performance of our algorithm compared to benchmark prediction methods. As benchmark algorithms we use well known algorithms such as linear/logistic regression and k-Nearest Neighbors, which are still a current research topic [4]–[6].

8) Based on our simulations, we suggest a preferred way of designing courses that enables early prediction and early intervention. Using data from a pilot course, we demonstrate the advantages of the suggested design.

The rest of the paper is organized as follows. Section II discusses related work in the field of grade and GPA prediction in education. In Section III we introduce notation, define data structures, formalize the problem and present the grade prediction algorithm. We analyze the data, describe benchmark methods and present simulation results including our and benchmark algorithms in Section IV. Finally, we draw conclusions in Section V.

II. RELATED WORK

Various studies have investigated the value of standardized tests [7]–[9] admissions exams [10] and GPA in previous programs [8] in predicting the academic success of students in undergraduate or graduate schools. They agree on a positive correlation between these predictors and success measures such as GPA or degree completion. Besides standardized tests, the relevancy of other variables for predictions of a student's GPA have been investigated, usually resulting in the conclusion that GPA from prior education and past grades in certain subjects (e.g., math, chemistry) [11], [12] have a strongly positive correlation

as well. Reference [11] observes that simple linear and more complex nonlinear (e.g., artificial neural network) models frequently lead to similar prediction accuracies and concludes that there is either no complex nonlinear pattern to be found in the underlying data or the pattern cannot be recognized by their approach. Our simulations support the statement that simple linear models show a similar accuracy in grade predictions as more complex methods.

Reference [13] argues that the accuracy of GPA predictions frequently is mediocre due to different grading standards used in different classes and shows a higher validity for grade predictions in single classes. Consequently, many works focus on identifying relationships between a student's grade in a particular class and variables related to the student [14]–[23]. Relevant factors were found to include the student's prior GPA [14], [15], [19]–[21], [23], performance in related courses [20], [21], [23], previous semester marks [17], performance in entrance exams [15], performance in early assignments of the class [21], [23], class attendance [19], self-efficacy [22] and whether the student is repeating the class [21].

A limitation of the algorithms in the previously discussed papers is that they are difficult to apply in many education scenarios. Frequently, variables related to the student such as performance in related classes, GPA or self-efficacy are not available to the instructor because the data has not been collected or is not accessible due to privacy reasons. However, the instructor always has access to data he/she collects from his/her own course, such as the performance of each student in early homework assignments or midterm exams. This paper, therefore, focuses on predicting the final grade based on this easily accessible data, which is collected anyway by the instructor.

Other works [24]–[30], which also exclusively use data from the course itself, differ significantly from this paper in several aspects. First, they rely on logged data in online education or Massive Open Online Course (MOOC) systems such as information about video-watching behavior, time spent on specific questions or forum activity. In contrast, our results are applicable to both online and offline courses, which include some kind of graded assignments or related feedback from the students during the course. Second, in order for the instructor to be able to take corrective actions it is of great importance to predict with a certain confidence the performance of students as early as possible. While our algorithm takes this into account by deciding for each student individually the best time to make the prediction using a confidence measure, related works do not provide a metric indicating the optimal time to predict. Third, while related works need training data from the course whose grades they want to predict, we show that we can use training data from past year classes of the same course. Finally, in contrast to algorithms from related work, which are only shown to be applicable to classification settings (e.g., pass/fail or letter grade), our algorithm can be used both in regression and classification settings.

To make the predictions, related works use various data mining models such as regression models [14], [26], decision trees [15]–[18], [25], [26], [30], support vector machines [14], [23]–[25], neural networks [15], [27], [29], Bayesian classifiers [15], [25], clustering [26] and nearest neighbor techniques [23], [24], [29], [30].

|	[20], [22]	[15]–[18], [23]	[14], [19], [21]	[24]	[25]–[30]	Our Work
Goal of Paper	Find Relevant Features	Predict Course Grade	Predict Course Grade	Predict Accuracy of Answer	Predict Course Grade	Predict Course Grade
Features	Other	Course & Other	Course & Other	From Course	From Course	From Course
Learning from Past Years	n/a	No	No	No	No	Yes
Accuracy-Timeliness Trade-Off	n/a	No	No	No	No	Yes
Regression / Classification	n/a	Classification	Both	Classification	Classification	Both
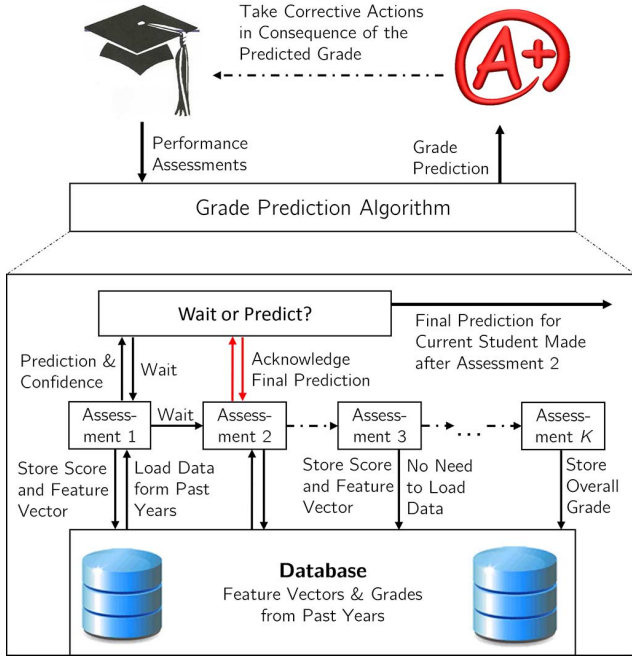
Fig. 1. System diagram for a single student.

Table I summarizes the comparison between our paper and related work investigating and predicting student performance in a course.

III. FORMALISM, ALGORITHM AND ANALYSIS

In this section we mathematically formalize the problem and propose an algorithm that predicts the final score or a classification according to the final grade of a student with a given confidence.

A. Definitions and System Description

Consider a course which is taught for several years with only slight modifications. Students attending the course have to complete performance assessments such as graded homework assignments, course projects and in-class exams and quizzes throughout the entire course.[1] Our goal is to predict with a certain confidence the overall performance of a student before all performance assessments have been taken. See Fig. 1 for a depiction of the system.

We consider a discrete time model with $y = 1, 2, \ldots, Y$ and $k = 1, 2, \ldots, K$ where y denotes the year in which the course is taught and k the point in time in year y after the kth performance assessment has been graded. Y gives the total number of years

during which the course is taught and K is the total number of performance assessments of each year. For a given year y we use index i as a representation of ith student of the year and I_y to denote the total number of students attending in year y. Except for the rare case that a student retakes the course, the students in each year are different. Let $a_{i,y,k} \in [0, 1]$ denote the normalized score or grade of student i in performance assessment k of year y.

The feature vector of yth year student i after having taken performance assessment k is given by $\mathbf{x}_{i,y,k} = (a_{i,y,1}, \ldots, a_{i,y,k})$. The normalized overall score $z_{i,y} \in [0, 1]$ of yth year student i is the weighted sum of all performance assessments

$$z_{i,y} = \sum_{k=1}^{K} w_k a_{i,y,k} \quad (1)$$

where the w_k denote the weight of performance assessment k so that $\sum_{k=1}^{K} w_k = 1$. The weights are set by the instructor and we assume that in each year the number, sequence and weight of performance assessments is the same. This assumption is reasonable since the content of a course usually does not change drastically over the years and frequently the same course material (e.g., course book) is used.[2] This is especially true in an introductory course such as the one we investigate in Section IV. The residual (overall score) $c_{i,y,k}$ of yth year student i after performance assessment k is defined as

$$c_{i,y,k} = \begin{cases} \sum_{l=k+1}^{K} w_l a_{i,y,l} & k \in \{1, \ldots, K-1\} \\ 0 & k = K \end{cases} \quad (2)$$

Using this definition we can write the overall score of yth year student i as

$$z_{i,y} = c_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}. \quad (3)$$

Note that after having taken the performance assessment k, the instructor has access to all the scores up to assignment k but the residual scores $c_{i,y,k}$ need to be estimated. We denote the estimate of the residual score for yth year student i at time k by $\hat{c}_{i,y,k}$ and the corresponding estimate of the overall score by $\hat{z}_{i,y,k}$. In binary classification settings, where the goal is to predict whether a student achieves a letter grade above or below a certain threshold, we denote the class of yth year student i by $b_{i,y} \in \{0, 1\}$.

For each student i we store the set of feature vectors $\mathbf{X}_{i,y} = \{\mathbf{x}_{i,y,k} \mid k \in \{1, \ldots, K\}\}$, the set of residuals $\mathbf{C}_{i,y} = \{c_{i,y,1}, \ldots, c_{i,y,K-1}\}$ and the student's overall score

---

[1]The performance assessments are usually graded by teaching assistants, by the instructor or even by other students through peer review [31].

[2]This assumption is made for simplicity. As we discuss in Section IV-B and show in Fig. 6 we can apply our algorithm to settings where different instructors using a different number and sequence of performance assessments and using different weights for each performance assessment teach the course.

$z_{i,y}$. All feature vectors from all students of year y are given by $\mathbf{X}_y = \bigcup_{i=1}^{I_y} \mathbf{X}_{i,y}$ and $\mathbf{X} = \bigcup_{y=1}^{Y} \mathbf{X}_y$ denotes all feature vectors of all completed years. Similarly $\mathbf{C} = \bigcup_{y=1}^{Y} \bigcup_{i=1}^{I_y} \mathbf{C}_{i,y}$ and $\mathbf{Z} = \bigcup_{y=1}^{Y} \bigcup_{i=1}^{I_y} z_{i,y}$ denote all residuals and overall scores of all completed years. Let $\mathbf{X}^{k'} = \{\mathbf{x}_{i,y,k} \mid k = k', \forall i, y\}$ denote the set of feature vectors and $\mathbf{C}^{k'} = \{c_{i,y,k} \mid k = k', \forall i, y\}$ denote the set of residuals saved after performance assessment k'.

B. Problem Formulation

Having introduced notations, definitions and data structures, we now formalize the grade prediction problem. We will investigate two different types of predictions. The objective of the first type, which we refer to as regression setting, is to accurately predict the overall score of each student individually in a timely manner. The second problem, referred to as classification setting, aims at making a binary prediction whether the student will do well or poorly or whether he/she will necessitate additional help or not. Again, the prediction is personalized and takes timeliness into account. For both types of predictions, the same algorithm can be used with only slight modifications, which we discuss in Section III-D. We will also show that the binary prediction problem can easily be generalized to a classification into three or more classes.

Irrespective of the type of the prediction, the decision for a yth year student i consists of two parts. First, we decide after which performance assessment $k_{i,y}^*$ to predict for the given student and second we determine his/her estimated overall score $\hat{z}_{i,y}$ or his/her estimated binary classification $\hat{b}_{i,y}$. At a point in time k of year y all scores including the overall scores of all students of past years $1, \ldots, y-1$ are known. Thus all feature vectors $\mathbf{x} \in \mathbf{X}$, residuals $c \in \mathbf{C}$ and overall scores $z \in \mathbf{Z}$ of all completed years are known. Furthermore, the scores $a_{i,y,1}, \ldots, a_{i,y,k}$ of yth year student i up to assessment k are known as well and do not have to be estimated. However, to determine the overall score of the student we need to predict his/her residual score $c_{i,y,k}$ consisting of performance assessments $k+1, \ldots, K$ since they lie in the future and are unknown. At time k we have to decide for each student of the current year whether this is the optimal time $k_{i,y}^* = k$ to predict or whether it is better to wait for the next performance assessment. If we decide to predict, we determine the optimal prediction of the overall score $\hat{z}_{i,y} = \hat{z}_{i,y,k_{i,y}^*}$. Both decisions are made based on the feature vector $\mathbf{x}_{i,y,k}$ of the given student and the feature vectors $\mathbf{x} \in \mathbf{X}^k$ and residuals $c \in \mathbf{C}^k$ of past students. To determine the optimal time to predict, we calculate a confidence $q_{i,y}(k)$ indicating the expected accuracy of the prediction for each student after each performance assessment. The prediction for a particular student is made as soon as the confidence exceeds a user-defined threshold $q_{i,y}(k) > q_{\text{th}}$. The problem of finding the optimal prediction time for yth year student i is formalized as follows:

$$\begin{array}{ll} \underset{k}{\text{minimize}} & k \\ \text{subject to} & q_{i,y}(k) > q_{\text{th}} \end{array} \qquad (4)$$

The optimization problem results in the optimal prediction time $k_{i,y}^*$.

C. Grade Prediction Algorithm, Regression Setting

In this section we propose an algorithm that learns to predict a student's overall performance based on data from classes held in past years and based on the student's results in already graded performance assessments. We describe the algorithm for the regression setting and explain the changes needed to use the algorithm in the classification setting in Section III-D.

Since at time k we know the scores $a_{i,y,1}, \ldots, a_{i,y,k}$ of the considered student from past performance assessments as well as the corresponding weights $w_1, \ldots, w_k$, we only predict the residual $c_{i,y,k}$ and calculate the prediction of the overall score with (3). To make its prediction for the current residual of a student with feature vector $\mathbf{x}_{i,y,k}$, the algorithm finds all feature vectors from similar students of past years and their corresponding residuals $c_{i,y,k}$. We define the similarity of students through their feature vectors. Two feature vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^k$ are similar if $\langle \mathbf{x}_i, \mathbf{x}_j \rangle_k \leq r$ where $\langle \cdot, \cdot \rangle_k$ is a distance metric defined on the feature space $\mathbf{X}^k$ and r is a parameter. For two feature vectors $\mathbf{x} \in \mathbf{X}^{k_1}$ and $\mathbf{x}' \in \mathbf{X}^{k_2}$ from different feature spaces (i.e., $k_1 \neq k_2$) the distance metric is not defined since we only need to determine distances within a single feature space. Different feature spaces can have different definitions of the distance metric; we are going to define the distance metrics we use in Section IV-B. We define a neighborhood $B(\mathbf{x}_c, r)$ with radius r of feature vector $\mathbf{x}_c \in \mathbf{X}^k$ as all feature vectors $\mathbf{x} \in \mathbf{X}^k$ with $\langle \mathbf{x}_c, \mathbf{x} \rangle_k \leq r$.

Let C^k denote the random variable representing the residual score after performance assessment k. $v^k(C^k \mid \mathbf{x})$ denotes the probability distribution over the residual score for a student with feature vector $\mathbf{x}$ at time k and $\mu^k(\mathbf{x})$ denotes the student's expected residual score. Let $p^k(\mathbf{x})$ denote the probability distribution of the students over the feature space $\mathbf{X}^k$. Intuitively $p^k(\mathbf{x})$ is the fraction of students with feature vector $\mathbf{x}$ at time k. Note that the distributions $v^k(C^k \mid \mathbf{x})$ and $p^k(\mathbf{x})$ are not sampling distributions but unknown underlying distributions. We assume that the distributions do not change over the years.

We define the probability distribution of the students in a neighborhood $B(\mathbf{x}_c, r)$ with center $\mathbf{x}_c$ and radius r as

$$p_{\mathbf{x}_c, r}^k(\mathbf{x}) := \frac{p^k(\mathbf{x})}{\int_{\mathbf{x} \in B(\mathbf{x}_c, r)} dp^k(\mathbf{x})} \mathbf{1}_{B(\mathbf{x}_c, r)}(\mathbf{x}),$$

where $\mathbf{1}$ is the indicator function. Intuitively $p_{\mathbf{x}_c, r}^k(\mathbf{x})$ is the fraction of students in neighborhood $B(\mathbf{x}_c, r)$ with feature vector $\mathbf{x}$. Let $C^k(B(\mathbf{x}_c, r))$ be the random variable representing the residual score of students in neighborhood $B(\mathbf{x}_c, r)$ after having taken performance assessment k. The distribution of $C^k(B(\mathbf{x}_c, r))$ is given by

$$f_{\mathbf{x}_c, r}^k(C^k) := \int_{\mathbf{x} \in \mathbf{X}^k} v^k(C^k \mid \mathbf{x}) dp_{\mathbf{x}_c, r}^k(\mathbf{x})$$

We denote the true expected value of the residual scores after assignment k of students in a particular neighborhood by $\mu^k(\boldsymbol{x}_c, r) := \mathbb{E}(C^k(B(\mathbf{x}_c, r)))$. Note that

$$\mu^k(\boldsymbol{x}_c, r) = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}_c, r}^k}[\mathbb{E}[C^k \mid \mathbf{x}]] = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}_c, r}^k}[\mu^k(\mathbf{x})]$$

$$= \int_{\mathbf{x} \in \mathbf{X}^k} \mu^k(\mathbf{x}) dp_{\mathbf{x}_c, r}^k.$$

Our estimation of the true expected residual of students within a particular neighborhood $B(\boldsymbol{x}_{i,y,k}, r)$ is given by

$$\hat{\mu}(C^k(B(\mathbf{x}_{i,y,k}, r))) = \frac{\sum\limits_{\mathbf{x} \in B(\mathbf{x}_{i,y,k}, r)} c_{\mathbf{x},k}}{|B(\mathbf{x}_{i,y,k}, r)|} \quad (5)$$

where $c_{\mathbf{x},k}$ denotes the residual after time k of the student with feature vector $\mathbf{x}$. For notational simplicity, we use $\hat{\mu}^k(\boldsymbol{x}_{i,y,k}, r) := \hat{\mu}(C^k(B(\mathbf{x}_{i,y,k}, r)))$ to denote the estimated expectation. In the following we are going to derive how confident we are in the estimation of the residual score based on a given neighborhood $B(\mathbf{x}, r)$ and how we use this confidence $q(B(\mathbf{x}, r))$ to both select the optimal radius of the neighborhood and to decide when to predict.

Intuitively, if the feature vectors after performance assessment k in a neighborhood $B(\mathbf{x}, r)$ of $\mathbf{x}$ contain a lot of information about the residual $c_{\mathbf{x},k}$, past students with feature vectors in this neighborhood should have had similar residuals. Hence, the variance of the residuals $\mathrm{Var}(C^k(B(\boldsymbol{x}_{i,y,k}, r)))$ of the students in the neighborhood should be small. To mathematically support this intuition, we consider the residuals $c_{i,y,k}$ in a neighborhood $B(\mathbf{x}, r)$ of feature vector $\mathbf{x}$ with distribution $f^k_{\mathbf{x}_c, r}(C^k)$. For any confidence interval ϵ the probability that the absolute difference between the unknown residual $c_{\mathbf{x},k}$ of the student with feature vector $\mathbf{x}$ and the expected value of the residual distribution $\mu^k(\mathbf{x}, r)$ in his/her neighborhood is smaller than ϵ can be bounded by

$$P[|C^k(B(\boldsymbol{x}, r)) - \mu^k(\boldsymbol{x}, r)| < \epsilon] > 1 - \frac{\mathrm{Var}(C^k(B(\boldsymbol{x}, r)))}{\epsilon^2}. \quad (6)$$

This statement directly follows from Chebyshev's inequality.

We conclude that the lower the variance of the residual distribution in the neighborhood, the more confident we are that the true residual $c_{\mathbf{x},k}$ will be close to $\mu^k(\mathbf{x}, r)$. Since both the expected value $\mu^k(\mathbf{x}, r)$ and the variance $\mathrm{Var}(C^k(B(\boldsymbol{x}, r)))$ of the distribution are unknown, we estimate the two values through the sample mean from (5) and the sample variance $\widehat{\mathrm{Var}}(C^k(B(\boldsymbol{x}, r)))$ given by

$$\widehat{\mathrm{Var}}(C^k(B(\boldsymbol{x}, r))) = \frac{\sum_{\mathbf{x} \in B(\mathbf{x}, r)} (c_{\mathbf{x},k} - \hat{\mu}^k(\boldsymbol{x}, r))^2}{|B(\mathbf{x}, r)| - 1}. \quad (7)$$

In the following we use $\mathrm{Var}^k(\boldsymbol{x}, r) := \mathrm{Var}(C^k(B(\boldsymbol{x}, r)))$ to denote the variance and $\widehat{\mathrm{Var}}^k(\mathbf{x}, r) := \widehat{\mathrm{Var}}(C^k(B(\boldsymbol{x}, r)))$ to denote the sample variance of the residual distribution in neighborhood $B(\mathbf{x}, r)$. From the law of large number it follows that the sample mean and the sample variance converge to the true expected value and the true variance for $|B(\mathbf{x}, r)| \to \infty$. We will provide a bound for the probability that the prediction error is larger than a given value in the theorem below. Given a desired confidence interval ϵ, we define the confidence on the prediction of the residual as

$$q(B(\mathbf{x}, r)) = 1 - \frac{\widehat{\mathrm{Var}}^k(\boldsymbol{x}, r)}{\epsilon^2}. \quad (8)$$

Using this confidence measure the radius of the optimal neighborhood after performance assessment k is given by

$r^* = \arg\max_r q(B(\mathbf{x}_{i,y,k}, r)) = \arg\min_r \widehat{\mathrm{Var}}^k(\boldsymbol{x}, r)$. To estimate r^* after each performance assessment k, our algorithm considers M different neighborhoods $B(\mathbf{x}_{i,y,k}, r_m), m = 1, \ldots, M$ with user-defined radii r_m and chooses the best neighborhood $\hat{m}_k(\mathbf{x}_{i,y,k})$ according to our confidence measure $\hat{m}_k(\mathbf{x}_{i,y,k}) = \arg\max_m q(B(\mathbf{x}_{i,y,k}, r_m))$. In the following we use $\hat{m}_k := \hat{m}_k(\mathbf{x}_{i,y,k})$ to denote the best neighborhood. Let

$$\hat{c}_{i,y,k} := \hat{\mu}^k(\mathbf{x}_{i,y,k}, r_{\hat{m}_k}) \quad (9)$$

denote the estimated residual of the best neighborhood at time k and $\hat{z}_{i,y,k}$ denotes the corresponding estimated overall score

$$\hat{z}_{i,y,k} = \hat{c}_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}. \quad (10)$$

If the confidence bound for the best neighborhood $q_{i,y}(k) = q(B(\mathbf{x}_{i,y,k}, r_{\hat{m}_k}))$ is above a given threshold $q_{i,y}(k) \geq q_{\mathrm{th}}$, the algorithm returns the final prediction of the overall score $\hat{z}_{i,y} = \hat{z}_{i,y,k}$ for the considered student.

If the confidence is below the threshold, we wait for the next performance assessment and start the next iteration. Fig. 2 illustrates the neighborhood selection process. Algorithm 1 provides a formal description of the grade prediction algorithm in pseudocode.

Algorithm 1: Grade Prediction Algorithm, Regression Setting

Input: All $\mathbf{x}$ and z from past years, q_{th}, number M and radii $r_1, \ldots, r_M$ of neighborhoods
Output: Predictions $\hat{z}$ for the overall scores of the students
1: **for all** years y **do**
2:   **for all** performance assessments k **do**
3:     **for all** current-year students i for whom the final prediction has not been made **do**
4:       **if** $k = K$ **then**
5:         Calculate $z_{i,y}$ according to (1)
6:         Return $z_{i,y}$ as final prediction for student i
7:       **end if**
8:      Create M neighborhoods with radii $r_1, \ldots, r_M$
9:      **for all** neighborhoods m **do**
10:       Estimate residual $\hat{c}(B(\mathbf{x}_{i,y,k}, r_m))$ with (5)
11:       Compute $\widehat{\mathrm{Var}}(\mathbf{x}, r_m)$ with (7)
12:       Compute $q(B(\mathbf{x}_{i,y,k}, r_m))$ with (8)
13:      **end for**
14:      Find $\hat{m}_k = \arg\max_m q(B(\mathbf{x}_{i,y,k}, r_m))$
15:      **if** $q(B(\mathbf{x}_{i,y,k}, r_{\hat{m}_k})) \geq q_{th}$ **then**
16:       Compute $\hat{z}_{i,y}$ with (3)
17:       Return $\hat{z}_{i,y}$ as final prediction for student i
18:      **end if**
19:      Add $\mathbf{x}_{i,y,k}$ and $a_{i,y,k}$ to database
20:     **end for**
21:   **end for**
22:   Calculate all $c_{i,y,k}$ of year y according to (2)
23:   Add all $c_{i,y,k}$ to database
24: **end for**

To conclude the discussion of the grade prediction algorithm in the regression setting, we derive a bound for the probability that the prediction error is larger than a value ϵ. Before we state
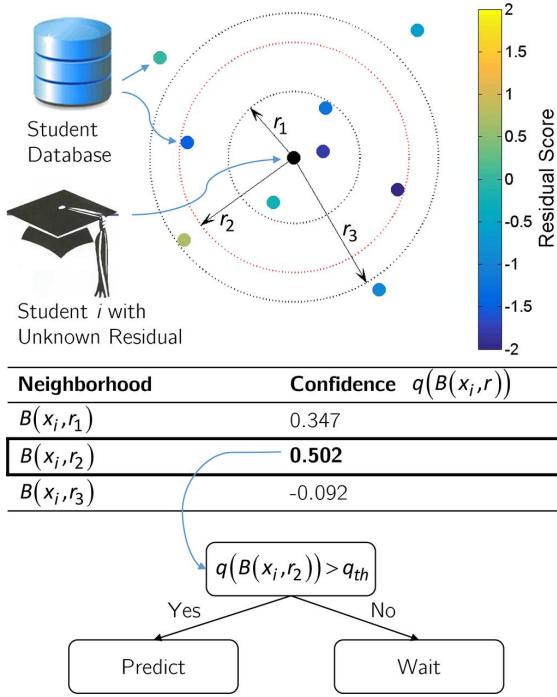
Fig. 2. Illustration of the neighborhood selection process.

the theorem, we introduce some further notations. Let $m_k^*(\mathbf{x})$ denote the index of the neighborhood with the smallest variance of residuals for the student with feature vector $\mathbf{x}$ at time k

$$m_k^*(\mathbf{x}) = \arg\min_{1 \le m \le M} \text{Var}^k(\mathbf{x}, r_m). \qquad (11)$$

Note that $m_k^*(\mathbf{x})$ is not necessarily equal to $\hat{m}_k(\mathbf{x})$, the index of the neighborhood with the highest confidence chosen by our algorithm, since the confidence defined in (8) is calculated with the known sample variance of residuals $\widehat{\text{Var}}(\mathbf{x}, r)$ and not with the unknown true variance $\text{Var}^k(\mathbf{x}, r)$ used in (11).

Similarly $m_{k,2}^*(\mathbf{x})$ denotes the index of the neighborhood with the second highest confidence.

$$m_{k,2}^*(\mathbf{x}) = \arg\min_{1 \le m \le M, m \neq m_k^*(\mathbf{x})} \text{Var}^k(\mathbf{x}, r_m).$$

Let $\Delta_k(\mathbf{x})$ denote the difference between the standard deviations of the residual distribution of neighborhoods $m_k^*(\mathbf{x})$ and $m_{k,2}^*(\mathbf{x})$

$$\Delta_k(\mathbf{x}) = \sqrt{\text{Var}^k(\mathbf{x}, r_{m_{k,2}^*})} - \sqrt{\text{Var}^k(\mathbf{x}, r_{m_k^*})}. \qquad (12)$$

Theorem: Without loss of generality we assume that all scores a are normalized to the range $[0, 1]$. Consider the prediction $\hat{z}_{i,y,k}$ of the overall score of yth year student i with feature vector $\mathbf{x}$ made by algorithm 1. The probability that the absolute error the prediction exceeds ϵ is bounded by

$$P[|z_{i,y} - \hat{z}_{i,y,k}| \ge \epsilon] \le \frac{4\text{Var}^k\left(\mathbf{x}, r_{m_k^*(\mathbf{x})}\right)}{\epsilon^2}$$
$$+ 2\exp\left[-\epsilon^2 \min_{1 \le m \le M} \frac{|B(\mathbf{x}, r_m)|}{2}\right]$$
$$+ 2M\exp\left[-\Delta_k(\mathbf{x})^2 \min_{1 \le m \le M} \frac{|B(\mathbf{x}, r_m)| - 1}{8}\right]$$

Proof: See Appendix.　■

This theorem illustrates two important aspects of algorithm 1. First, we see that for a given neighborhood the accuracy of our predictions increases with an increasing number of neighbors. Hence, our algorithm learns the best predictions online as the knowledge base is expanded after each year, when the feature vectors and results from the past-year students are added to the database. In Section IV-D1 we show that this learning can be experimentally illustrated with our data from the digital signal processing course taught at UCLA. Second, the term $\text{Var}^k(\mathbf{x}, r_{m_k^*})/\epsilon^2$ shows that the prediction accuracy will be higher if the variance of the residuals in a neighborhood is small. With increasing time k we expect this variance to decrease since we have more information about the students and we expect the students in a neighborhood to be more similar and achieve similar (residual) scores.

Note that it is possible to restrict the data kept in the knowledge base to recent years, which allows the algorithm to adapt faster to slowly changing students and to changes in the course.

D. Grade Prediction Algorithm, Classification Setting

In the binary classification setting we predict the overall score analogously to the regression setting and then determine the class by comparing the predicted overall score $\hat{z}_{i,y}$ with a threshold score z_{th}. To illustrate how we find z_{th} let us assume that we want to predict whether a student does well (letter grades $\ge B-$) or does poorly (letter grades $\le C+$). To determine z_{th}, we find the average $z_{\text{avg},B-}$ of all students from past years who received a $B-$ and the average $z_{\text{avg},C+}$ of all students from past years who achieved a $C+$. Subsequently, we define z_{th} as $z_{\text{th}} = (z_{\text{avg},B-} + z_{\text{avg},C+})/2$. The predicted classification $\hat{b}_{i,y}$ of yth year student i is then given by

$$\hat{b}_{i,y} = \begin{cases} 0 & \hat{z}_{i,y} \ge z_{\text{th}} \\ 1 & \hat{z}_{i,y} < z_{\text{th}}. \end{cases} \qquad (13)$$

We are more confident in the classification not only if the variance of the neighbor-scores is small, which is the metric we used for the confidence in the regression setting, but also if the distance $d(B(\mathbf{x}, r)) = |\hat{z}(B(\mathbf{x}, r)) - z_{\text{th}}|$ between the predicted score and the threshold score is large. Note that $\hat{z}(B(\mathbf{x}, r))$ is the estimate of the overall score based on neighborhood $B(\mathbf{x}, r)$. Because of this intuition we use a modified confidence

$$q^{\text{bin}}(B(\mathbf{x}, r)) = 1 - e^{-d(B(\mathbf{x},r))} \frac{\widehat{\text{Var}}(C^k(B(\mathbf{x}, r)))}{\epsilon^2} \qquad (14)$$

to decide whether to make the final prediction in binary classification settings. Since $d(B(\mathbf{x}, r))$ should only influence whether the final prediction is made for a given neighborhood but not the neighborhood selection process, we still use the unmodified confidence from (8) to select the optimal neighborhood.

In summary, four changes have to be made to algorithm 1 to make it applicable to binary classification settings. First, z_{th} has to be determined/updated at the beginning of each new year. Second, we calculate $\hat{b}_{i,y}$ after line 16 according to (13). Third, we return $\hat{b}_{i,y}$ at line 17 instead of $\hat{z}_{i,y}$. Fourth, we use the modified confidence q^{bin} according to (14) in line 15 instead of the unmodified confidence q. We use the unmodified confidence from (8) in line 14.

The described binary classification algorithm can easily be generalized to a larger number of categories. In a classification

with L categories, we define $L - 1$ threshold values $z_{\mathrm{th},1} < z_{\mathrm{th},2} < \cdots < z_{\mathrm{th},L-1}$ and determine in which of the L score intervals $\{[0, z_{\mathrm{th},1}), [z_{\mathrm{th},1}, z_{\mathrm{th},2}), \ldots, [z_{\mathrm{th},L-1}, 1]\}$ the predicted overall score $\hat{z}_{i,y}$ of a student lies. The index of the interval corresponds to the classification of the student. In this general classification setting, the modified confidence from (14) can be used as well by defining d as the distance of $\hat{z}_{i,y}$ to the nearest threshold value.

We discuss the performance of the proposed algorithm 1 in both regression and classification settings in Section IV-D.

E. Confidence-Learning Prediction Algorithm

Besides the radii of the neighborhoods r_i, the only parameter to be chosen by the user in algorithm 1 is the desired confidence threshold q_{th}. Since for an instructor it is more natural and practical to specify a desired prediction accuracy or error directly rather than the confidence threshold, we show in this section how to automatically learn the appropriate confidence threshold to achieve a certain prediction performance and what consequences this has on the average prediction time. We will discuss a possible way of choosing the radii r_i of the neighborhoods in Section IV-B.

Formally we define the problem as follows. Let $p(k, q_{\mathrm{th}}) \in [0, 1]$ denote the proportion of current year students for which the grade prediction algorithm working with confidence threshold $q_{\mathrm{th}} \leq 1$ has predicted the overall score by time (performance assessment) $k \in [0, K]$. $p_{\min}$ is the minimum percentage of current year students whose grade the user wants to predict with a specified accuracy. $E(k, q_{\mathrm{th}}) \geq 0$ denotes the average absolute prediction error up to time k for the given confidence. $E_{\max}$ is the maximum error the user is willing to tolerate. $k(p, q_{\mathrm{th}})$ is the time necessary to predict the grade for proportion p of all students of the class using confidence threshold q_{th}. Please note that since the variables p, E, q_{th} and k are dependent, we can only independently specify two of the four variables. If we for example specify to predict all ($p = 1$) students with zero error ($E = 0$), the algorithm will have to wait until the end of the course when the overall score is known ($k = K$) and will use maximum confidence ($q_{\mathrm{th}} = 1$). Without making any assumptions on the dependence of the variables of each other, multiple pairs (k, q_{th}) might lead to the same specified pair (p, E).

Our goal is, therefore, to learn from past data the yth year estimate of the minimal time k_y and corresponding confidence threshold $q_{\mathrm{th},y}$ necessary to achieve the desired share $p_{\min}$ of students predicted and the desired maximum average prediction error $E_{\max}$. This is formally defined as:

$$\begin{aligned} \underset{q_{\mathrm{th}}}{\mathrm{minimize}} \quad & k(p, q_{\mathrm{th}}) \\ \mathrm{subject\ to} \quad & p(k, q_{\mathrm{th}}) \geq p_{\min} \\ & E(k, q_{\mathrm{th}}) \leq E_{\max} \end{aligned} \qquad (15)$$

Note that while the goal of optimization problem (4) is to find the minimum time to predict the overall score of a particular student with a desired confidence, this problem (15) aims at finding the minimum time by which the overall scores of a specific percentage of all students can be predicted with a desired maximum error.

At a given year y we solve this optimization problem using a brute force approach using the all available data from years

$1, \ldots, y$. For this purpose, we extract k, p and E from algorithm 2 for a large number of different confidence thresholds q_{th}. We then select the confidence threshold $q_{\mathrm{th},y}$ which is optimal with respect to optimization problem (15) and determine the corresponding prediction time k_y. To make the grade predictions for year $y+1$ we use the learned confidence threshold $q_{\mathrm{th},y}$ as input to prediction algorithm 1. Since there are no training data available yet at year $y = 1$, the algorithm uses a user-defined starting value $q_{\mathrm{th},0}$ for the grade predictions of the first year. Algorithm 2 summarizes the learning algorithm in pseudocode.

Algorithm 2: Confidence-Learning Prediction Algorithm

Input: $E_{\max}$, $p_{\min}$, $q_{\mathrm{th},0}$, all $\mathbf{x}$ and z from past years, number M and radii $r_1, \ldots, r_M$ of neighborhoods

Output: Predictions $\hat{z}$, k_y and $q_{\mathrm{th},y}$ for all years
1: **for all** years y **do**
2:    **if** $y > 1$ **then**
3:       Find k_{y-1} and $q_{\mathrm{th},y-1}$ according (15) by running algorithm 1 with various q_{th}
4:       Return k_{y-1} and $q_{\mathrm{th},y-1}$
5:    **end if**
6:    Use algorithm 1 with $q_{\mathrm{th}} = q_{\mathrm{th},y-1}$ to predict and return the grades of current year y students
7: **end for**

IV. EXPERIMENTS

In this section, we present the data, discuss details of the application of algorithm 1 to our dataset, illustrate the functioning of the algorithm and evaluate its performance by comparing it against other prediction methods in both regression and binary classification settings. Due to space limitations, we will not show experimental results for classification settings with more than two categories.

A. Data Analysis

Our experiments are based on a dataset from an undergraduate digital signal processing course (EE113) taught at UCLA over the past 7 years. The dataset contains the scores from all performance assessments of all students and their final letter grades. The number of students enrolled in the course for a given year varied between 30 and 156, in total the dataset contains the scores of approximately 700 students. Each year the course consists of 7 homework assignments, one in-class midterm exam taking place after the third homework assignment, one course project that has to be handed in after homework 7 and the final exam. The duration of the course is 10 weeks and in each week one performance assessments takes place. The weights of the performance assessments are given by: 20% homework assignments with equal weight on each assignment, 25% midterm exam, 15% course project and 40% final exam.[3] Fig. 3(a) shows the distribution of the letter grades assigned over the 7 years. We observe that on average B is the grade the instructor assigned most frequently. A was assigned second most and C third most

---

[3]As we explain in footnote 2 and show in Section IV-D2, our algorithm can also be applied to settings where the number and weights of performance assessments change over the years.
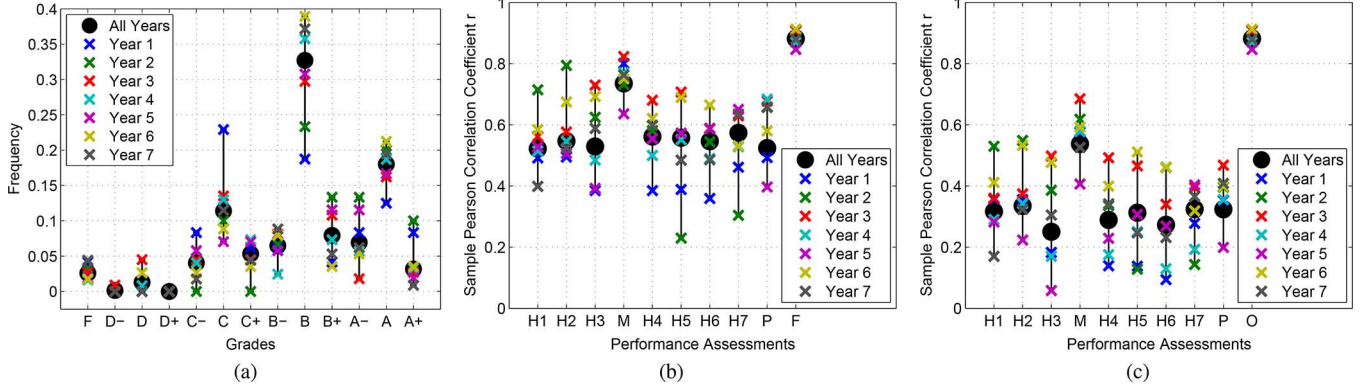
Fig. 3. Data analysis: (3a) shows the distribution of letter grades for all years. (3b) and (3c) present the sample Pearson correlation coefficient between individual performance assessments and the overall (3b) or final exam (3c) score. Note that we use the abbreviations Hi (homework assignment i), M (midterm exam), F (final exam) and O (overall score) in the figures. (a) Grade distribution, (b) Correlation coefficients overall score, (c) Correlation coefficients final score.

frequently. Surprisingly, however, the distribution varies drastically over the years; in year 1 for example only 18.75% received a B while in year 6 the frequency was 38.9%.

To understand the predictive power of the scores in different performance assessments, Fig. 3(b) shows the sample Pearson correlation coefficient between all performance assessments and the overall score. We make several important observations from this graph. First, on average the final exam has the strongest correlation to the overall score, followed by the midterm exam. This is not surprising, since the final contributes 40% and the midterm contributes 25% to the overall score. Second, the score from the course project on average does not have a higher correlation with the overall score than the homework assignments despite the fact that it accounts for 15% of the overall score. Third, all homework assignments have similar correlation coefficients. Fourth, the correlation between the individual performance assessments and the overall score varies greatly over the years. This indicates that predicting student scores based on training data from past years might be difficult.

Since all performance assessments are part of the overall score and, therefore, a high correlation is expected, it is also informative to consider the correlation between the performance assessments and the final exam shown in Fig. 3(c). It is interesting to observe that still the midterm exam shows, besides the overall score, the highest correlation with the final exam. A possible explanation for this is that both the midterm and final are in-class exams while the other performance assessments are take-home.

B. Our Algorithm

In this section we discuss four important details of the application of algorithm 1 to the dataset from the undergraduate digital signal processing course.

First, the rule we use to normalize all scores $a_{i,y,k}$ in our dataset is given by

$$a_{i,y,k} = \frac{a^*_{i,y,k} - \hat{\mu}_{y,k}}{\hat{\sigma}_y}, \qquad (16)$$

where $a^*_{i,y,k}$ is the original score of the student, $\hat{\mu}_{y,k}$ is the sample mean of all yth year student's original scores in performance assessment k and $\hat{\sigma}_y$ is the standard deviation of all yth year student's original overall scores. A normalization of

the scores is needed for several reasons. First, the instructor-defined maximum score in a particular performance assessment may differ greatly across years and since we use data from past years to predict the performance of students in a given year, we need to make the data across years comparable. Second, also the difficulty of individual performance assessments might be different across years, homework 2 might for instance be very easy in year 2 so that almost everyone achieves the maximum score and very difficult in year 3 so that few achieve half of the maximum score. The normalization according to (16) eliminates this bias by transforming the absolute scores of a student to scores relative to his/her classmates of the same year. Note that algorithm 1 does not require a specific normalization and it does not matter that the normalized scores according to (16) will not be in the interval [0, 1] as assumed in Section III for simplicity.

Second, we use feature vectors that simply contain the scores of all performance assessments student i has taken up to time k in the order they occurred $\mathbf{x}_{i,y,k} = (a_{i,y,1}, \ldots, a_{i,y,k})$. To incorporate the fact that students who have performed similarly in a performance assessment with a lot of weight should be nearer to each other in the feature space than students that have had similar scores in a performance assessment (e.g., homework assignment) with low weight, we use a weighted metric to calculate the distance between two feature vectors. We define the distance of two feature vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^k$ as

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle_k = \frac{\sum_{l=1}^{k} w_l |x_{i,l} - x_{j,l}|}{\sum_{l=1}^{k} w_l}, \qquad (17)$$

where k is the length of the feature vectors, w_l is the weight of performance assessment l and $x_{i,l}$ denotes entry l of feature vector $\mathbf{x}_i$.

Third, rather than specifying the radii of the neighborhoods to consider as an input, as suggested in the pseudo-code of algorithm 1, we automatically adapt the radii of the neighborhoods such that they contain a certain number of neighbors. Since the sample variance gets more accurate with an increasing number of samples, we refrain from considering neighborhoods with only 2 neighbors. Therefore, the smallest radius considered r_1 is the minimal radius such that the neighborhood includes 3 neighbors. For subsequent neighborhoods the minimal radius is chosen such that the neighborhood includes at least

one neighbor more than the previous neighborhood. Formally, we define the selection of the radii recursively as

$$r_1 = \min r, \text{ s.t. } |B(\mathbf{x}_{i,y,k}, r)| \geq 3$$
$$r_{m+1} = \min r, \text{ s.t. } |B(\mathbf{x}_{i,y,k}, r)| > |B(\mathbf{x}_{i,y,k}, r_m)|. \quad (18)$$

Fourth, to be able to apply our algorithm in settings in which structure of the course (e.g., the number, weight and sequence of assessments) changes across years, we need to pre-process the data from past years. In particular, the data from past years is pre-processed so that the number and sequence of in-class and take-home assessments is the same as in the current year. In addition, to identify the most similar students, it is important that the performance assessments that cover the same topic are at the same place of the sequence/feature vector and, therefore, are compared to each other. Consequently, we might have to pre-process the data from past years even if the total number of performance assessments is the same. We use two different types of modifications to pre-process data from past years.

Modification 1 applies to cases where a topic of the course was tested with a larger number of performance assessments in a past year than in a current year. For example, consider a signal processing course which contained two homework assignments on the Fast Fourier Transform in year 1 but the same topic was covered in only one homework assignment in year 2. In this case, the two performance assessments on the same topic from year 1 are combined to a single assessment. If N assessments are combined, the score of the combined assessment a_{comb} is calculated based on the weights $w_1, \ldots, w_N$ of the past assessments with scores $a_1, \ldots, a_N$ according to

$$a_{\text{comb}} = \frac{\sum_{k=1}^{N} w_k a_k}{\sum_{k=1}^{N} w_k} \quad (19)$$

Modification 2 applies to the case where a topic of the course was tested with a lower number of performance assessments in a past year than in a current year. In this case the past-year performance assessment on this topic is duplicated. Note that through duplication this performance assessment gets more weight in the process of selecting similar students. This is desired because the instructor probably uses more performance assessments to test a certain topic because he thinks that this topic is very important and hence it will be informative in terms of predicting the grade.

Finally, if necessary the sequence of performance assessments from the past years is reordered to match the sequence of performance assessments from the current year. The reordering has to be done so that the performance assessments on the same topic are at the same position of the sequence/feature vector in both years. Additionally, in-class assessments should only be compared to in-class assessments and take-home assessments should only be compared to take-home assessments. After this pre-processing of past-year data, the standard Algorithm 1 can be applied to make the predictions. Note that our algorithm always uses the weights of the current-year course to find students similar to the student for whom it needs to issue a personalized grade prediction and does not consider the weights that were used in past-year courses for the various assessments. The grade predictions for the current-year course are usually made based on data from several past-year courses. The data for each of the past years might have to be pre-processed separately.

TABLE II
CASE STUDY: ILLUSTRATIVE EXAMPLE

	HW 1	HW 2	HW 3	Midterm	Do Well/Poorly
Student 1	0.53	0.00	-0.37	-1.35	Poorly
Student 2	1.07	0.87	-0.30	-1.06	Poorly
Student 3	-1.39	-1.54	-2.15	0.50	Well

C. Benchmarks

We compare the performance of our algorithm against five different prediction methods.

- We use the score $a_{i,y,k}$ student i has achieved in the most recent performance assessment k alone to predict the overall grade.
- A second simple benchmark makes the prediction based on the scores $a_{i,y,1}, \ldots, a_{i,y,k}$ student i has achieved up to performance assessment k taking into account the corresponding weights of the performance assessments.
- The k-Nearest Neighbors algorithm with 7 neighbors. This number provided the best results with training data from the first year.
- Linear regression using the ordinary least squares (OLS) finds the least squares optimal linear mapping between the scores of first k performance assessments and the overall score.
- In classification settings we use logistic regression instead of linear regression.
- Support vector machines (SVMs) are used in the classification setting.

The advantage of the method we use in our algorithm over linear and logistic regression is that being a nearest neighbor method, it is able to recognize certain patterns such as trends in the data that are missed in linear/logistic regression where a single parameter per performance assessment has to fit all students. In contrast, our algorithm is able to detect such patterns if there have been students in the past who have shown similar patterns.

Table II illustrates this through a case study extracted from the UCLA undergraduate digital signal processing course data. We present cases from a simulation where we predicted whether students are going to do well (letter grade $\geq B-$) or do poorly (letter grade $\leq C+$) and consider the students for which our algorithm decided to predict after the midterm exam. The table shows 3 students whom logistic regression classified wrongly while our algorithm made the accurate prediction. In columns 2-4 we present the scores the students achieved up to the midterm exam and the last column shows the true classification of the students. These cases are typical examples of settings where our algorithm outperforms logistic regression. Student 1 and 2 both showed a good performance in homework assignment 1. However, in later assignments and especially at the midterm exam their performance successively deteriorated, an indication that the students might do poorly the class if they or the instructor and teaching assistants do not take corrective actions. Our algorithm is likely to have learned such patterns from past data and predicts the students to do poorly. On average, however, their performance in the first four performance assessments is still about average and, therefore, logistic regression predicts that the students will do well. For student 3 the situation is the other way around.
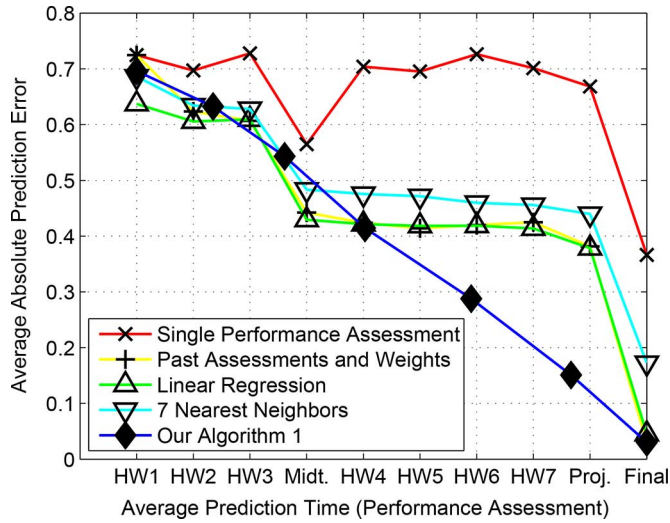
Fig. 4.   Performance comparison of different prediction methods.

Fig. 5.   Illustration of learning from past data: Error of grade predictions for year 7 depending on training data.

D. Results

In this section we evaluate the performance of our algorithm 1 in different settings and compared to benchmarks in both regression and the classification tasks.

As a performance measure in the regression setting, we use the average of the absolute values of the prediction errors E. Since we normalized the overall score to have zero mean and a standard deviation of 1, E directly corresponds to the number of standard deviations the predictions on average are away from the true values. The overall performance measure in classification settings is the accuracy of the classification. Furthermore, we use the quantities precision, recall and false positive/negative rate besides accuracy to measure performance. Please note that positive in our case means that the student does poorly.

1) Performance Comparison With Benchmarks in Regression Setting: Having discussed the various performance measures, we first address the regression setting. Fig. 4 visualizes the performance of the algorithm we presented in Section III-C and of benchmark methods. We generated Fig. 4 by predicting the overall scores of all students from years 2–7. To make the prediction for year y, we used the entire data from years 1 to $y - 1$ to learn from. Unlike our algorithms, the benchmark methods do not provide conditions to decide after which performance assessment the decision should be made. Therefore, for benchmark methods we specified the prediction time (performance assessment) k for an entire simulation and repeated the experiment for all $k = 1, \ldots, 10$; the results are plotted in Fig. 4. To generate the curve of our algorithm 1, we ran simulations using different confidence thresholds q_{th} and for each threshold we determined E and the performance assessment (time) $\bar{k}$ after which the prediction was made on average.

Irrespective of the prediction method, Fig. 4 shows the trade-off between timeliness and accuracy; the later we predict the more accurate our prediction gets. From the curve for the prediction using a single performance assessment we infer that there is a low correlation between homework assignments/course project and the overall score and a high correlation between the in-class assessments (midterm and final exam) and the overall score. This observation is congruent with the correlation analysis from Section IV-A. If the prediction is
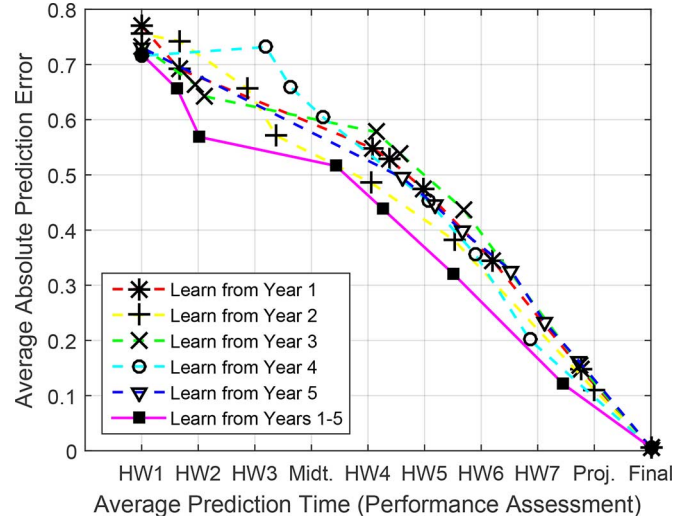
made early, before the midterm, all methods (except the prediction using a single performance assessment) lead to similar prediction errors. We observe that while the error decreases approximately linearly for our algorithm, the performance of benchmark methods steeply increases after the midterm and the final but stays approximately constant during the rest of the time. The reason for this is that we obtained the points of the curve for our algorithm by averaging the prediction time of all students. Therefore, the point of the curve above the midterm was not generated by predicting after the midterm for all students; some predictions were made earlier, some later. If on average the prediction is made after homework 4, our algorithm shows a significantly smaller error E than benchmark methods outperforming linear regression by up to 65%.

2) Learning Across Years and Instructors in Regression Setting: Consider Fig. 5 demonstrating the performance increase of our algorithm when more data to learn from become available. To generate the figure, we used our algorithm to predict the overall scores of all 7th year students for different confidence thresholds. The curves in dashed lines stem from simulations using only one of the years 1–5 as training data and the solid magenta curve uses all years 1–5 to learn from. We observe that the prediction performance strongly depends on the training data and differs if different years are used. Most importantly, the performance is highest irrespective of the average prediction time if the combination of the data from all 5 years is used. This shows that our algorithm is able to learn and improves its predictions over time.

The undergraduate digital signal processing course is taught twice a year by three different instructors at UCLA. While we used only the data from one instructor in the previous plots, Fig. 6 investigates the situation when we predict the grades for a class of instructor 1 based exclusively on past data from a different instructor 2. In practice this happens when a new instructor takes over a course previously taught by someone else. It is interesting to see whether our grade prediction still works well in this setting. A good performance is not self-evident for several reasons. Different instructors might set a different focus concerning the knowledge imparted, they might use a different
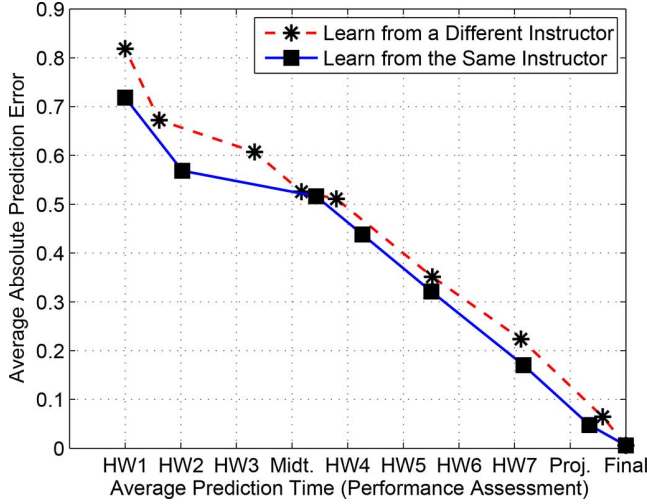
Fig. 6. Illustration of learning across instructors: Error of grade predictions for year 7 depending on training data.
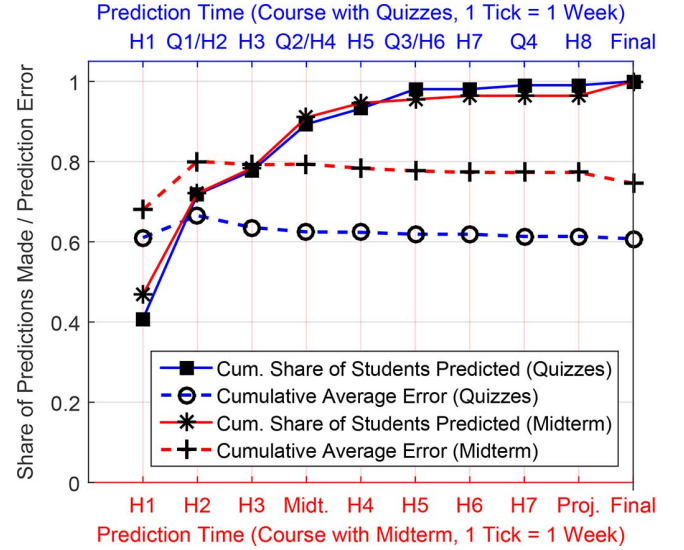

Fig. 7. Comparison of prediction time and accuracy between the UCLA course EE113, which contains a midterm exam, and the UCLA course EE103, which contains four in-class quizzes instead of a midterm exam. Note that the tick labels Qi/Hi above the plot stand for quiz/homework i and that for EE103 there are weeks in which both a homework and a quiz take place.
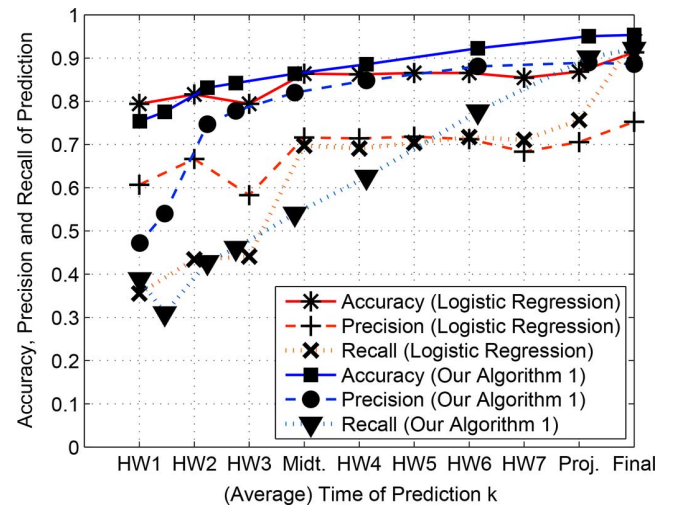

Fig. 8. Performance comparison between our algorithm and logistic regression using accuracy, precision and recall for binary do well/poorly classification.

textbook and they might prefer different styles of homework assignments and in-class exams. Furthermore, the structure of the course, e.g., the number and sequence of homework assignments, the time when the midterm exam takes place, the weights of performance assessments and whether a course project and quizzes exist, might change drastically. To generate Fig. 6, we predicted the overall score for the year 7 class of instructor 1 based on two different sets of previous data. The solid blue curve was generated by using the data from the classes in years 1–5 from the same instructor 1 as training data. To obtain the dashed red curve, we used the data from classes in years 1–5 from instructor 2 to learn from. While the predictions using training data from the same instructor are slightly more accurate, the performance with training data from a different instructor is still very satisfying, showing a good robustness of our algorithm with respect to different instructors. For the subsequent results we again exclusively use data from one instructor.

3) Performance Comparison With Course Containing Early Quizzes in Regression Setting: The results in both the data analysis section (Fig. 3(b)) and Section IV-D1 (Fig. 4) indicate that scores in in-class exams are much better predictors of the overall score than homework assignments. To verify this, we consider two consecutive years of the UCLA course EE103, which contains four in-class quizzes in course weeks 2, 4, 6 and 8 instead of a midterm. Fig. 7 visualizes that, starting from the first quiz in week 2, indeed our algorithm is able to predict the same percentage of the students with an up to 22% smaller cumulative average prediction error by a certain week. We generated Fig. 7 by using algorithm 1 to predict for both courses the overall scores of the students in a particular year based on data from the previous year. Note that for the course with quizzes, the increase in the share of students predicted is larger in weeks that contain quizzes than in weeks without quizzes. This supports the thesis that quizzes are good predictors as well.

According to this result, it is desirable to design courses with early in-class exams. This enables a timely and accurate grade prediction based on which the instructor can intervene if necessary.

4) Performance Comparison With Benchmarks in Classification Setting: The performances in the binary classification settings are summarized in Fig. 8. Since logistic regression turns out to be the most challenging benchmark in terms of accuracy in the classification setting, we do not show the performance of the other benchmark algorithms for the sake of clarity. The goal was to predict whether a student is going to do well, still defined as letter grades equal to or above $B-$, or do poorly, defined as letter grades equal to or below $C+$. Again, to generate the curves for the benchmark method, logistic regression, we specified manually when to predict. For our algorithm we again averaged the prediction times of an entire simulation and varied q_{th} to obtain different points of the curve. Up to homework 4, the performance of the two algorithms is very similar, both showing
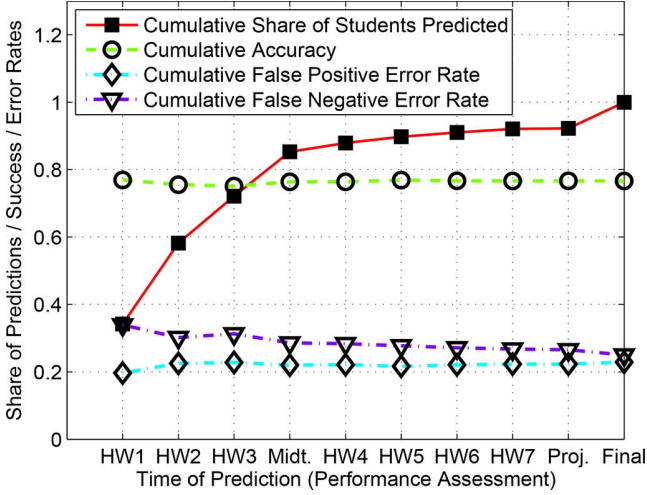
Fig. 9. Cumulative prediction time, accuracy, false positive and false negative error rates for a binary do well/poorly classification with fixed q_{th}.

a high prediction accuracy even with few performance assessments. Starting from homework 4, our algorithm performs significantly better, with an especially drastic improvement of recall. It is interesting to see that even with full information, the algorithms do not achieve a 100% prediction accuracy. The reason for this is that the instructor did not use a strict mapping between overall score and letter grade and the range of overall scores that lead to a particular letter grade changed slightly over the years.

5) Decision Time and Accuracy in Classification Setting: To better understand when our algorithm makes decisions and with what accuracy, consider Fig. 9. We again investigate binary do well/poorly classifications as discussed above. The red curve shows (square markers) for what share of the total number of students the algorithm makes the prediction by a specific point in time. The remaining curves show different measures of cumulative performance. We can for example see that by the midterm exam we classify 85% of the students with an accuracy of 76%. These timely predictions are desirable since the earlier the prediction is made the more time an instructor has to take corrective action. The cumulative accuracy stays almost constant around 80% irrespective of the prediction time. We believe that the reason for this is that thanks to the confidence threshold, the easy decisions are made early and harder decisions are made later. Consequently, the expected accuracy of all predictions remains more or less constant irrespective of the prediction time.

V. CONCLUSION

In this paper we develop an algorithm that allows for a timely and personalized prediction of the final grades of students exclusively based on their scores in early performance assessment such as homework assignments, quizzes or midterm exams. Using data from an undergraduate digital signal processing course taught at UCLA, we show that the algorithm is able to learn from past data, that it outperforms benchmark algorithms with regard to accuracy and timeliness both in classification and regression settings and that the predictions are robust even when the course is taught by different instructors.

We show that in-class exams are better predictors of the overall performance of a student than homework assignments. Hence, designing courses to have early in-class evaluations

enables timely identification of students who, with a high probability, would do poorly without intervention and enables remedial actions to be adopted at an early stage.

Our algorithm can easily be generalized to include context data from students such as their prior GPA or demographic data. If applied exclusively to MOOCs, the in-course data used for the predictions could be extended for example by the responses of students to multiple-choice questions, their forum activity, the course material they studied or the time they spent studying online. Another direction of future work is to apply our algorithm in practice and investigate to what extent the performance of students can be improved by a timely intervention based on the grade predictions. In this context, our algorithm could be extended to make multiple predictions for each student to monitor the trend in the predicted grade after an intervention.

One example for an intervention would be that the instructor provides additional study material to students with a low predicted grade. Alternatively, teaching assistants could spend additional time with these selected students to go through important topics again. In a MOOC setting, the intervention could take place in a fully automated way, for example by presenting the students additional study material in a personalized way using techniques discussed in [3]. To make students aware of their performance, they could be asked to predict their own overall grade and as a comparison the instructor could disclose the prediction of our algorithm to the students.

APPENDIX

In this Appendix, we proof the theorem from Section III-C. Before we start with the proof, we discuss some preliminary results.

Fact 1. (Chernoff-Hoeffding Bound): Let $X_1, X_2, \ldots, X_n$ be independent and bounded random variables with range $[0, 1]$ and expected value μ. Let $\hat{\mu}_n = (X_1 + \cdots + X_n)/n$ denote the sample mean of the random variables. Then, for all $\epsilon > 0$

$$P[|\hat{\mu}_n - \mu| \geq \epsilon] \leq 2 \exp[-2n\epsilon^2].$$

Proof: A proof of Fact 1 can be found in Hoeffding's paper [32]. ∎

Fact 2. (Empirical Bernstein Bound): Let $n \geq 2$ and $X_1, X_2, \ldots, X_n$ be independent and bounded random random variables with range $[0, 1]$ and variance Var. $\hat{\mu}_n$ denotes the n-sample mean $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\widehat{\text{Var}}_n$ denotes the n-sample variance $\widehat{\text{Var}}_n = \frac{1}{n-1}\sum_{i=1}^{N}(x_i - \hat{\mu}_n)^2$. Then, the following inequality bounds the probability that the error of the sample standard deviation, which is the square root of the sample variance, is larger than a given value

$$P[|\sqrt{\widehat{\text{Var}}_n} - \sqrt{\text{Var}}| \geq \epsilon] \leq 2\exp\left[-\frac{n-1}{2}\epsilon^2\right]$$

can be derived.

Proof: See [33] for a proof of Fact 2. ∎

Lemma 1: Let $\hat{m}_k(\mathbf{x})$ denote the index of the neighborhood selected by our algorithm for the student with feature vector $\mathbf{x}$ at time k and $m_k^*(\mathbf{x})$ is given by (11). M denotes the total number of neighborhoods our algorithm considers and $\Delta_k(\mathbf{x})$ is

given by (12). We can bound the probability that our algorithm chooses the wrong neighborhood by

$$P[\hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})] \leq 2e^{-\Delta_k(\mathbf{x})^2 \min_{1 \leq m \leq M} \frac{|B(\mathbf{x}, r_m)| - 1}{8}}.$$

(20)

Proof: Consider:

$$P[\hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})]$$
$$= P\left[\arg\min_{1 \leq m \leq M} \widehat{\mathrm{Var}}^k(\mathbf{x}, r_m) \neq m_k^*(\mathbf{x})\right]$$
$$= P\left[\arg\min_{1 \leq m \leq M} \sqrt{\widehat{\mathrm{Var}}^k(\mathbf{x}, r_m)} \neq m_k^*(\mathbf{x})\right].$$

If the estimation error of the standard deviation is smaller than $\Delta_k(\mathbf{x})/2$ for all neighborhoods

$$|\sqrt{\widehat{\mathrm{Var}}^k(\mathbf{x}, r_m)} - \sqrt{\mathrm{Var}^k(\mathbf{x}, r_m)}| \leq \frac{\Delta_k(\mathbf{x})}{2},$$

our algorithm chooses the optimal neighborhood $m_k^*(\mathbf{x})$. Therefore, we get

$$P\left[\arg\min_{1 \leq m \leq M} \sqrt{\widehat{\mathrm{Var}}^k(\mathbf{x}, r_m)} \neq m_k^*(\mathbf{x})\right]$$
$$\leq P\left[\bigcup_{1 \leq m \leq M} \left\{|\sqrt{\widehat{\mathrm{Var}}^k(\mathbf{x}, r_m)} - \sqrt{\mathrm{Var}^k(\mathbf{x}, r_m)}| \geq \frac{\Delta_k(\mathbf{x})}{2}\right\}\right]$$
$$\overset{(a)}{\leq} \sum_{m=1}^{M} P\left[|\sqrt{\widehat{\mathrm{Var}}^k(\mathbf{x}, r_m)} - \sqrt{\mathrm{Var}^k(\mathbf{x}, r_m)}| \geq \frac{\Delta_k(\mathbf{x})}{2}\right]$$
$$\overset{(b)}{\leq} \sum_{m=1}^{M} 2\exp\left[-\Delta_k(\mathbf{x})^2 \frac{|B(\mathbf{x}, r_m)| - 1}{8}\right]$$
$$\leq 2M\exp\left[-\Delta_k(\mathbf{x})^2 \min_{1 \leq m \leq M} \frac{|B(\mathbf{x}, r_m)| - 1}{8}\right]$$

where (a) is the union bound and (b) follows from Fact 2. ∎

Proof of Theorem: Note that

$$|z_{i,y} - \hat{z}_{i,y,k}|$$
$$\overset{(a)}{=} \left|c_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l} - \left(\hat{c}_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}\right)\right|$$
$$= |c_{i,y,k} - \hat{c}_{i,y,k}|$$

where (a) follows from (3) and (10).

There are three sources of error in the prediction of an overall score of algorithm 1:

1) The wrong neighborhood size may be selected due to inaccurate approximations of the true residual score variances of the neighborhoods through the sample variance.
2) If the optimal neighborhood is selected, the sample mean of the residual scores in the neighborhood may not be a good approximation of their true mean.
3) Even if the optimal neighborhood is selected and the sample mean equals the true mean, the residual score of the considered student may be different from the mean of the residual score distribution.

In the following we separate these three error sources and derive a bound for each one.

We have

$$P[|z_{i,y} - \hat{z}_{i,y,k}| \geq \epsilon] = P[|c_{i,y,k} - \hat{c}_{i,y,k}| \geq \epsilon]$$
$$\overset{(b)}{=} P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{\hat{m}_k(\mathbf{x})})| \geq \epsilon\right]$$
$$\overset{(c)}{=} P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{\hat{m}_k(\mathbf{x})})| \geq \epsilon, \hat{m}_k(\mathbf{x}) = m_k^*(\mathbf{x})\right]$$
$$\quad + P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{\hat{m}_k(\mathbf{x})})| \geq \epsilon, \hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})\right]$$
$$\overset{(d)}{\leq} P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \epsilon, \hat{m}_k(\mathbf{x}) = m_k^*(\mathbf{x})\right]$$
$$\quad + P[\hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})]$$
$$\overset{(e)}{\leq} P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \epsilon\right] + P[\hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})]$$

where (b) follows from (9), (c) is the law of total probability and (d) and (e) both follow from the fact that $P[A, B] \leq P[A]$. Lemma 1 provides a bound for the second term. Therefore, we focus on the first term

$$P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \epsilon\right]$$
$$= P\left[|c_{i,y,k} - \mu^k(\mathbf{x}, r_{m_k^*(\mathbf{x})}) + \mu^k(\mathbf{x}, r_{m_k^*(\mathbf{x})}) - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \epsilon\right]$$
$$\overset{(f)}{\leq} P\left[|c_{i,y,k} - \mu^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \frac{\epsilon}{2}\right]$$
$$\quad + P\left[|\mu^k(\mathbf{x}, r_{m_k^*(\mathbf{x})}) - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \frac{\epsilon}{2}\right]$$
$$\overset{(g)}{\leq} \frac{4\mathrm{Var}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})}{\epsilon^2} + 2\exp\left[-\frac{\epsilon^2}{2}|B(\mathbf{x}, r_{m_k^*(\mathbf{x})})|\right]$$
$$\leq \frac{4\mathrm{Var}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})}{\epsilon^2} + 2\exp\left[-\epsilon^2 \min_{1 \leq m \leq M} \frac{|B(\mathbf{x}, r_m)|}{2}\right]$$

where (f) follows from the triangle inequality, the fact that $P[X + Y \geq x_0 + y_0] \leq P[\{X \geq x_0\} \cup \{Y \geq y_0\}]$ and the union bound. The bound for the first term in step (g) follows from Chebyshev's inequality and the bound for the second term follows from the Chernoff-Hoeffding Bound from Fact 1.

Including the second term again and using Lemma 1 we get

$$P[|z_{i,y} - \hat{z}_{i,y,k}| \geq \epsilon] = P[|c_{i,y,k} - \hat{c}_{i,y,k}| \geq \epsilon]$$
$$\leq P\left[|c_{i,y,k} - \hat{\mu}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})| \geq \epsilon\right] + P[\hat{m}_k(\mathbf{x}) \neq m_k^*(\mathbf{x})]$$
$$\leq \frac{4\mathrm{Var}^k(\mathbf{x}, r_{m_k^*(\mathbf{x})})}{\epsilon^2} + 2\exp\left[-\epsilon^2 \min_{1 \leq m \leq M} \frac{|B(\mathbf{x}, r_m)|}{2}\right]$$
$$+ 2M\exp\left[-\Delta_k(\mathbf{x})^2 \min_{1 \leq m \leq M} \frac{|B(\mathbf{x}, r_m)| - 1}{8}\right],$$

which concludes the proof. ∎

References

[1] R. Baraniuk, "Open education: New opportunities for signal processing," in *Plenary Speech, 2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[2] Openstax college [Online]. Available: http://openstaxcollege.org/, accessed: 2015-05-07

[3] C. Tekin, J. Braun, and M. van der Schaar, "Etutor: Online learning for personalized education," in *Proc. 2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[4] G. Marjanovic and V. Solo, "$l\_\{q\}$ sparsity penalized linear regression with cyclic descent," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1464–1475, Mar. 15, 2014.

[5] S. Marano, V. Matta, and P. Willett, "Nearest-neighbor distributed learning by ordered transmissions," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5217–5230, Nov. 1, 2013.

[6] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, May 15, 2010.

[7] N. R. Kuncel and S. A. Hezlett, "Standardized tests predict graduate students success," *Science*, vol. 315, pp. 1080–1081, 2007.

[8] E. Cohn, S. Cohn, D. C. Balch, and J. Bradley, "Determinants of undergraduate gpas: Sat scores, high-school GPA and high-school rank," *Econ. Ed. Rev.*, vol. 23, no. 6, pp. 577–586, 2004.

[9] E. R. Julian, "Validity of the medical college admission test for predicting medical school performance," *Acad. Med.*, vol. 80, no. 10, pp. 910–917, 2005.

[10] P. A. Gallagher, C. Bomba, and L. R. Crane, "Using an admissions exam to predict student success in an ADN program," *Nurse Ed.*, vol. 26, no. 3, pp. 132–135, 2001.

[11] W. L. Gorr, D. Nagin, and J. Szczypula, "Comparative study of artificial neural network and statistical models for predicting student grade point averages," *Int. J. Forecasting*, vol. 10, no. 1, pp. 17–34, 1994.

[12] N. T. Nghe, P. Janecek, and P. Haddawy, "A comparative analysis of techniques for predicting academic performance," in *Proc. 37th Annu. Frontiers In Education Conf.—Global Engineering: Knowledge Without Borders, Opportunities Without Passports (FIE'07)*, 2007, pp. T2G–7, IEEE.

[13] R. D. Goldman and R. E. Slaughter, "Why college grade point average is difficult to predict," *J. Ed. Psychol.*, vol. 68, no. 1, p. 9, 1976.

[14] S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," *Comput. Ed.*, vol. 61, pp. 133–145, 2013.

[15] E. Osmanbegović and M. Suljić, "Data mining approach for predicting student performance," *Econ. Rev.*, vol. 10, no. 1, 2012.

[16] B. K. Baradwaj and S. Pal, "Mining educational data to analyze students' performance," 2012 [Online]. Available: http://arxiv.org/abs/1201.3417, unpublished

[17] S. K. Yadav, B. Bharadwaj, and S. Pal, "Data mining applications: A comparative study for predicting student's performance," 2012 [Online]. Available: http://arxiv.org/abs/1202.4815, unpublished

[18] A. B. E. D. Ahmed and I. S. Elaraby, "Data mining: A prediction for student's performance using classification method," *World J. Comput. Appl. Technol.*, vol. 2, no. 2, pp. 43–47, 2014.

[19] P. Cortez and A. M. G. Silva, "Using data mining to predict secondary school student performance," 2008.

[20] L. H. Werth, "Predicting sttudent performance in a beginning computer science class," in *Proc. 17th SIGCSE Technical Symp. Computer Science Education (SIGCSE '86)*, 1986, vol. 18, no. 1, pp. 138–143.

[21] J. L. Turner, S. A. Holmes, and C. E. Wiggins, "Factors associated with grades in intermediate accounting," *J. Accounting Ed.*, vol. 15, no. 2, pp. 269–288, 1997.

[22] A. Y. Wang and M. H. Newlin, "Predictors of web-student performance: The role of self-efficacy and reasons for taking an on-line class," *Compute. Human Behav.*, vol. 18, no. 2, pp. 151–163, 2002.

[23] S. Kotsiantis, C. Pierrakeas, and P. Pintelas, "Predicting students' performance in distance learning using machine learning techniques," *Appl. Artif. Intell.*, vol. 18, no. 5, pp. 411–426, 2004.

[24] C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," in *Proc. IEEE 34th INFOCOM*, 2015, to be published.

[25] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura, "Predicting students' final performance from participation in on-line discussion forums," *Comput. Ed.*, vol. 68, pp. 458–472, 2013.

[26] M. I. Lopez, J. Luna, C. Romero, and S. Ventura, "Classification via clustering for predicting final marks based on student participation in forums.," presented at the Int. Conf. Educational Data Mining, 2012.

[27] M. D. Calvo-Flores, E. G. Galindo, M. P. Jiménez, and O. P. Pineiro, "Predicting students marks from moodle logs using neural network models," *Current Devel. Technol.-Assisted Ed.*, vol. 1, pp. 586–590, 2006.

[28] D. García-Saiz and M. Zorrilla, "A promising classification method for predicting distance students performance," *Proc. 5th Int. Conf. Educational Data Mining*, pp. 206–207, 2012.

[29] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, "Data mining algorithms to classify students," in *Proc. Int. Conf. Educational Data Mining*, 2008, pp. 8–17.

[30] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. F. Punch, "Predicting student performance: An application of data mining methods with an educational web-based system," in *Proc. 33rd Annu. Frontiers in Education Conf.*, 2003, vol. 1, pp. T2A–13.

[31] Y. Xiao, F. Dörfler, and M. van der Schaar, "Incentive design in peer review: Rating and repeated endogenous matching," in *Proc. Allerton Conf.*, 2014.

[32] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.

[33] A. Maurer and M. Pontil, "Empirical bernstein bounds and sample variance penalization," in *Proc. Int. Conf. Learning Theory*, 2009.

Yannick Meier received the B.Sc. degree in information technology and electrical engineering from ETH Zurich, Zurich, Switzerland (Swiss Federal Institute of Technology Zurich) in 2014.

In 2014 and 2015 he conducted research visits at University of Pennsylvania, Philadelphia, PA, USA and at University of California, Los Angeles, CA, USA. He is currently pursuing the M.Sc. degree in information technology and electrical engineering at ETH Zurich.
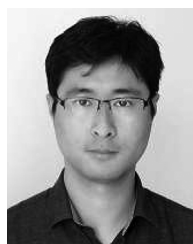
Jie Xu is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Miami. He received his B.S. and M.S. degrees in electronic engineering from Tsinghua University in China in 2008 and 2010, respectively, and a Ph.D. degree in electrical engineering from University of California, Los Angeles (UCLA) in 2015. Dr. Xu's research interests are in game theory and learning theory with applications to education, communication, signal processing and network security. He received the Distinguished Ph.D. Dissertation Award in Signals & Systems at UCLA.

Onur Atan received the B.Sc. degree in electrical engineering from Bilkent University, Ankara, Turkey in 2013 and the M.Sc. degree in electrical engineering from the University of California, Los Angeles in 2014. He is currently pursuing the Ph.D. degree in electrical engineering at the University of California, Los Angeles. He received the best M.Sc. thesis award in Electrical Engineering at the University of California, Los Angeles. His research interests include online learning and multi-armed bandit problems and their applications to medical informatics and education.

Mihaela van der Schaar (F'09) is Chancellor's Professor in the Electrical Engineering Department at UCLA. Her research interests include machine learning for medical informatics and education, online learning, stream mining, networks, network science, social networks and game theory. She received numerous awards, including the NSF Career Award, three IBM Faculty Awards, several best paper awards including the Darlington Best Paper Award. She has also 33 U.S. patents.