Personalized Grade Prediction: A Data Mining Approach

Yannick Meier, Jie Xu, Onur Atan and Mihaela van der Schaar
University of California, Los Angeles, CA

***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. 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 demonstrate the performance of our algorithm on a dataset obtained based on the performance of approximately 700 undergraduate students who have taken an introductory digital signal processing over the past 7 years. 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; 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 [1]–[3].

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. However, with large class sizes at universities and even larger class sizes in Massive Open Online Courses (MOOCs) it has become impossible for the instructor and teaching assistants to keep track of the performance of each student individually. 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.

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. 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. First, 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. Second, 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; for other students it might take more time to make an equally accurate prediction. This 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 are.

1) We propose an algorithm that makes a personalized and timely prediction of the grade of each student in a class.
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 analyze real data from an introductory digital signal processing course over 7 years and use the data to experimentally demonstrate the performance of our algorithm compared to benchmark prediction methods.
5) 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.

Table I summarizes the comparison between our paper and related work. Due to space limitations, a detailed discussion of related work can be found in the electronic preprint of a longer version of this paper [4].

II. Formalism, Algorithm and Analysis

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

A. Definitions and System Description

Consider a course taught for several years with only slight modifications. Students attending the course have to complete performance assessments such as graded homework assignments, course projects, in-class exams and quizzes throughout

TABLE I
COMPARISON WITH RELATED WORK

	[5], [6]	[7]	[8], [9]	[10]	[11]–[13]	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	Regression

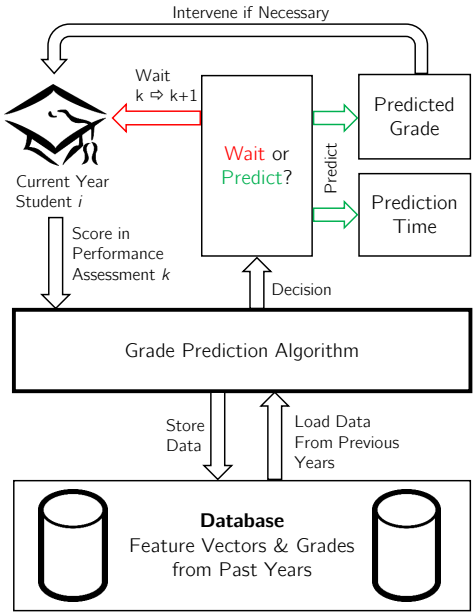


Fig. 1. System diagram for a single student.

the course.[1] Our goal is to predict with a certain confidence the overall performance of a student before all performance assessments have been taken. Fig. 1 illustrates 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. 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 $z_{i,y} = \sum_{k=1}^{K} w_k a_{i,y,k}$ 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.[2] The residual (overall score) $c_{i,y,k}$ of yth year student i after performance assessment k is defined as

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

Using this definition we can write the overall score of yth year student i as $z_{i,y} = c_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}$. 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 $\hat{z}_{i,y,k}$.

For each student i we store the set of feature vectors $\mathbf{X}_{i,y} = \{\mathbf{x}_{i,y,k} | k \in \{1, \ldots, K\}\}$, the set of residuals $\mathbf{C}_{i,y} = \{c_{i,y,1}, \ldots, c_{i,y,K-1}\}$ and the student's overall score $z_{i,y}$. We use $\mathbf{C} = \bigcup_{y=1}^{Y} \bigcup_{i=1}^{I_y} \mathbf{C}_{i,y}$ and $\mathbf{Z} = \bigcup_{y=1}^{Y} \bigcup_{i=1}^{I_y} z_{i,y}$ to denote all residuals and overall scores of all completed years. Let $\mathbf{X}^{k'} = \{\mathbf{x}_{i,y,k} | k = k', \forall i, y\}$ denote the set of feature vectors and $\mathbf{C}^{k'} = \{c_{i,y,k} | k = k', \forall i, y\}$ 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. The objective is to accurately predict the overall score of each student individually in a timely manner.

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 $\hat{z}_{i,y}$. 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

[1]The performance assessments are usually graded by teaching assistants, by the instructor or even by an automated system [14].

[2]This assumption is made for simplicity. As we show in the online preprint of a longer version of this paper we can apply our algorithm to settings where different instructors using different weights for each performance assessment teach the course [4]. A prediction across courses with a different number of performance assessments is possible as well, for example by combining the scores of two or more performance assessments to a single score.

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 $q_{i,y}(k) > q_{th}$. The problem of finding the optimal prediction time for yth year student i is formalized as follows:

$$\begin{aligned} &\underset{k}{\text{minimize}} \quad k \\ &\text{subject to} \quad q_{i,y}(k) > q_{th} \end{aligned} \tag{2}$$

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

C. Grade Prediction Algorithm

In this section we propose an algorithm that learns to predict a student's overall score based on data from classes held in past years and based on the student's results in already graded performance assessments.

At time k we predict the residual $c_{i,y,k}$ and calculate the prediction of the overall score with $z_{i,y} = c_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}$. 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 $\langle ., . \rangle_k$ is a distance metric defined on the feature space $\mathbf{X}^k$ and r is a parameter. Different feature spaces can have different definitions of the distance metric; we are going to define the distance metrics we use in Section III-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. $v^k\left(C^k|\mathbf{x}\right)$ 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 $v^k\left(C^k|\mathbf{x}\right)$ 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

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

where $\mathbf{1}$ is the indicator function. Intuitively $p^k_{\mathbf{x}_c,r}(\mathbf{x})$ 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

$$f^k_{\mathbf{x}_c,r}\left(C^k\right) := \int_{\mathbf{x} \in \mathbf{X}^k} v^k(C^k|\mathbf{x}) dp^k_{\mathbf{x}_c,r}(\mathbf{x})$$

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

$$\begin{aligned} \mu^k(\boldsymbol{x}_c, r) &= \mathbb{E}_{\mathbf{x} \sim p^k_{\mathbf{x}_c,r}}\left[\mathbb{E}\left[C^k|\mathbf{x}\right]\right] = \mathbb{E}_{\mathbf{x} \sim p^k_{\mathbf{x}_c,r}}\left[\mu^k(\mathbf{x})\right] \\ &= \int_{\mathbf{x} \in \mathbf{X}^k} \mu^k(\mathbf{x})\, dp^k_{\mathbf{x}_c,r}. \end{aligned}$$

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

$$\hat{\mu}(C^k(B(\mathbf{x}_{i,y,k}, r))) = \frac{\sum\limits_{\mathbf{x} \in B(\mathbf{x}_{i,y,k}, r)} c_{\mathbf{x},k}}{|B(\mathbf{x}_{i,y,k}, r)|} \tag{3}$$

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 $\text{Var}\left(C^k(B(\boldsymbol{x}_{i,y,k}, r))\right)$ 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 $f^k_{\mathbf{x}_c,r}\left(C^k\right)$. 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

$$P\left[|C^k(B(\boldsymbol{x}, r)) - \mu^k(\boldsymbol{x}, r)| < \epsilon\right] > 1 - \frac{Var\left(C^k(B(\boldsymbol{x}, r))\right)}{\epsilon^2}. \tag{4}$$

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 $\text{Var}\left(C^k(B(\boldsymbol{x}, r))\right)$ of the distribution are unknown, we estimate the two values through the sample mean from (3) and the sample variance $\widehat{Var}\left(C^k(B(\boldsymbol{x}, r))\right)$ given by

$$\widehat{Var}\left(C^k(B(\boldsymbol{x}, r))\right) = \frac{\sum_{\mathbf{x} \in B(\mathbf{x}, r)} \left(c_{\mathbf{x},k} - \hat{\mu}^k(\boldsymbol{x}, r)\right)^2}{|B(\mathbf{x}, r)| - 1}. \tag{5}$$

In the following we use $Var^k(\boldsymbol{x}, r) := \text{Var}\left(C^k(B(\boldsymbol{x}, r))\right)$ to denote the variance and $\widehat{Var}^k(\mathbf{x}, r) := \widehat{Var}\left(C^k(B(\boldsymbol{x}, r))\right)$

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 $q\left(B(\mathbf{x}, r)\right) = 1 - \widehat{Var}^{k}\left(\boldsymbol{x}, r\right)/\epsilon^2$.

Using this confidence measure the radius of the optimal neighborhood after performance assessment k is given by $r^* = \arg\max_r q\left(B\left(\mathbf{x}_{i,y,k}, r\right)\right) = \arg\min_r \widehat{Var}^{k}\left(\boldsymbol{x}, r\right)$. 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 $\hat{m}_k(\mathbf{x}_{i,y,k}) = \arg\max_m q\left(B(\mathbf{x}_{i,y,k}, r_m)\right)$. In the following we use $\hat{m}_k := \hat{m}_k(\mathbf{x}_{i,y,k})$ to denote the best neighborhood. Let $\hat{c}_{i,y,k} := \hat{\mu}^k\left(\mathbf{x}_{i,y,k}, r_{\hat{m}_k}\right)$ denote the estimated residual of the best neighborhood at time k and $\hat{z}_{i,y,k}$ denotes the corresponding estimated overall score $\hat{z}_{i,y,k} = \hat{c}_{i,y,k} + \sum_{l=1}^{k} w_l a_{i,y,l}$. If the confidence bound for the best neighborhood $q_{i,y}(k) = q\left(B\left(\mathbf{x}_{i,y,k}, r_{\hat{m}_k}\right)\right)$ is above a given threshold $q_{i,y}(k) \geq q_{th}$, 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. Due to space limitations, an illustration of the neighborhood selection process and a formal description of the grade prediction algorithm in pseudocode can be found online in the preprint of a longer version of this paper [4].

To conclude the discussion of the grade prediction algorithm, we derive a bound for the probability that the prediction error is larger than a value ϵ. Before we state 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

$$m_k^*(\mathbf{x}) = \underset{1 \leq m \leq M}{\arg\min}\ Var^k(\mathbf{x}, r_m). \tag{6}$$

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 is calculated with the known sample variance of residuals $\widehat{Var}(\mathbf{x}, r)$ and not with the unknown true variance $Var^k(\mathbf{x}, r)$ used in (6).

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

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

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

$$\Delta_k(\mathbf{x}) = \sqrt{Var^k(\mathbf{x}, r_{m_{k,2}^*})} - \sqrt{Var^k(\mathbf{x}, r_{m_k^*})}. \tag{7}$$

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 our algorithm. The probability that the absolute error the prediction exceeds ϵ is bounded by*

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

Proof: Due to space limitations, the proof can be found online in the preprint of a longer version of this paper [4]. ■

This theorem illustrates two important aspects of our algorithm. 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. Second, the term $Var^k(\mathbf{x}, r_{m_k^*})/\epsilon^2$ 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.

III. Experiments

A. Data Analysis

Our experiments are based on a dataset from an undergraduate digital signal processing course 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 assessment 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.

To understand the predictive power of the scores in different performance assessments, Fig. 2a 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.

B. Our Algorithm

In this section we discuss three important details of the application of the grade prediction algorithm 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 $a_{i,y,k} = \left(a^*_{i,y,k} - \hat{\mu}_{y,k}\right)/\hat{\sigma}_y$, 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.[3]

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 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle_k = \sum_{l=1}^{k} w_l \left| x_{i,l} - x_{j,l} \right| / \sum_{l=1}^{k} w_l$ 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 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.

C. Benchmarks

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

[3]Note that our algorithm does not require a specific normalization and it does not matter that the normalized scores we use will not be in the interval $[0, 1]$ as assumed in Section II for simplicity.

- 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.
- 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.
- The k-Nearest Neighbors algorithm with 7 neighbors. This number provided the best results with training data from the first year.

The advantage of the method we use in our algorithm over linear 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 regression where a single parameter per performance assessment has to fit all students.

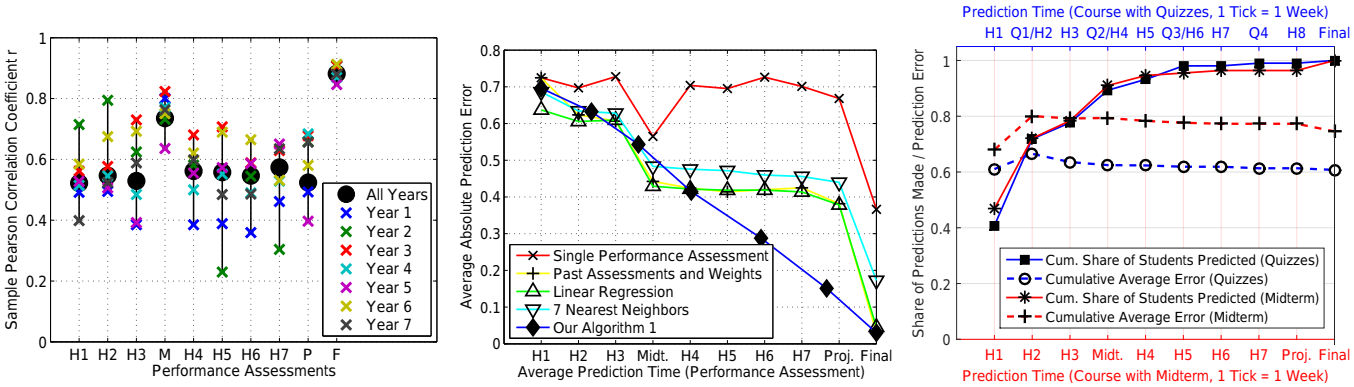
D. Results

In this section we evaluate the performance of our algorithm in different settings and compared to benchmarks.

As a performance measure 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.

1) Performance Comparison with Benchmarks: Fig. 2b visualizes the performance of the algorithm we presented in Section II-C and of benchmark methods. We generated Fig. 2b 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. 2b. To generate the curve of our algorithm, 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. 2b shows the trade-off between timeliness and accuracy; the later we predict the more accurate our prediction gets. 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



(a) Correlation coefficients overall score (b) Performance Comparison (c) Course with Early Quizzes

Fig. 2. 2a: Sample Pearson correlation coefficient between individual performance assessments and the overall score. Note that we use the abbreviations Hi (homework assignment i), M (midterm exam) and F (final exam) in the figure. 2b: Performance comparison of different prediction methods. 2c: Comparison of prediction time and accuracy between the course which contains a midterm exam, and the course 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 the course with quizzes there are weeks in which both a homework and a quiz take place.

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 outperforms linear regression by up to 65%.

2) Performance Comparison with Course Containing Early Quizzes: The results in both the data analysis section (Fig. 2a) and Section III-D1 (Fig. 2b) 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 a course which contains four in-class quizzes in course weeks 2, 4, 6 and 8 instead of a midterm. Fig. 2c 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. 2c by using our algorithm 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.

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.

IV. Conclusion

This paper proposes a systematic method for personalized grade prediction. 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.

References

[1] C. Tekin, J. Braun, and M. van der Schaar, "etutor: Online learning for personalized education," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015.

[2] M. A. Evans and A. Johri, "Facilitating guided participation through mobile technologies: designing creative learning environments for self and others," *Journal of Computing in Higher Education*, vol. 20, no. 2, pp. 92–105, 2008.

[3] "Openstax college," http://openstaxcollege.org/, accessed: 2015-05-07.

[4] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Predicting grades," *arXiv preprint arXiv:1508.03865*, 2015.

[5] L. H. Werth, *Predicting student performance in a beginning computer science class*. ACM, 1986, vol. 18, no. 1.

[6] 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," *Computers in Human Behavior*, vol. 18, no. 2, pp. 151–163, 2002.

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

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

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

[10] C. G. Brinton and M. Chiang, "Mooc performance prediction via clickstream data and social learning networks," in *34th INFOCOM IEEE*. 2015, To appear.

[11] 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 *Frontiers in education, 2003. FIE 2003 33rd annual*, vol. 1. IEEE, 2003, pp. T2A–13.

[12] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, "Data mining algorithms to classify students." in *EDM*, 2008, pp. 8–17.

[13] D. Garcıa-Saiz and M. Zorrilla, "A promising classification method for predicting distance students performance." *EDM*, pp. 206–207, 2012.

[14] V. Aggarwal, A. Minds, S. Srikant, and V. Shashidhar, "Principles for using machine learning in the assessment of open response items: Programming assessment as a case study," in *NIPS Workshop on Data Driven Education*, 2013.