Predicting Students Performance in Educational Data Mining

Bo Guo*, Rui Zhang†, Guang Xu‡, Chuangming Shi§ and Li Yang§
School of Computer
Hubei University of Education, Wuhan, Hubei, China
Email: guobo.chris@gmail.com*
Email: zhangr2013@gmail.com†
Email: xuguang@hue.edu.cn‡
Email: shichuangming@hue.edu.cn§
Email: yangli@hue.edu.cn§

***Abstract*—Predicting student academic performance has been an important research topic in Educational Data Mining (EDM) which uses machine learning and data mining techniques to explore data from educational settings. However measuring academic performance of students is challenging since students academic performance hinges on diverse factors. The interrelationship between variables and factors for predicting performance participate in complicated nonlinear ways. Traditional data mining and machine learning techniques may not be applied directly to these types of data and problems. In this study we develop a classification model to predict student performance using Deep Learning which automatically learns multiple levels of representation. We pre-train hidden layers of features layerwisely using an unsupervised learning algorithm sparse auto-encoder from unlabeled data, and then use supervised training for fine-tuning the parameters. We train model on a relatively large real world students dataset, and the experimental results show the effectiveness of the proposed method which can be applied into academic pre-warning mechanism.**

I. Introduction

Applying data mining and machine learning methods in education is an emerging interdisciplinary field, which forms a new research field called Educational Data Mining (EDM) [1]. EDM uses these machine learning and data mining techniques to explore data from educational settings to find out predictions and patterns that characterize students behaviors and performance. Its goal is to better identify the settings in which they learn to improve educational outcomes and to gain insights into educational phenomena.

In EDM, predicting the performance of a student is a great concern to the education managements. For example, it could give an appropriate warning to students those who are at risk by forecasting the grade of students, and help them to avoid problems and overcome all difficulties in study. However measuring of academic performance of students is challenging since students academic performance hinges on diverse factors or characteristics such as demographics, personal, educational background, psychological, academic progress and other environmental variables. The interrelationship between these variables participating in the complex and multi-faceted problem of academic performance are not clearly understood, and they are often related in a complicated nonlinear way.

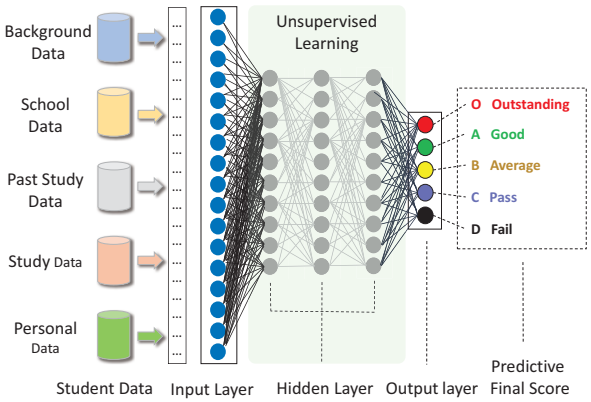

Fig. 1. **Students performance prediction system overview**. We learn multi-level representations by unsupervised training hidden layers, then neural network is fine-tuned using back-propagation. The input of network is a flat vector of different kinds of student information, the output is a multiple classification of softmax which indicates the student final examination score.

Therefore using machine learning techniques in EDM to explore educational data and discover latent meaningful patterns for predicting students marks prevails recently. The performance of machine learning methods is heavily dependent on the choice of data representation. A variety of machine learning related approaches has been proposed to predict student performance. Romero et.al [2] used multiple linear regression model and support vector machine (SVM) to predict overall and individual student academic performance. Jia et.al [3] predicted students retention by combining SVM and a shallow neural network to improved the classification accuracy. Musso et.al [4] applied traditional artificial neural networks to predicting general academic performance. Kotsiantis [5] used regression method to predict the students marks in a distance learning system. Wolff et.al [6] developed a predictive model using decision trees and SVM with data from several Open University to forecast students pattern. These methods are all based on shallow architectures [7] which implement one or two

978-1-4673-7370-8/15 $31.00 © 2015 IEEE
DOI 10.1109/ISET.2015.33

layer feature representation. Shallow model can not capture all relationships among factors especially when the data is relatively large and correlated. Traditional data mining and machine learning techniques may not be applied directly to these types of data and problems.

In this study a prediction system, called Students Performance Prediction Network (SPPN), is proposed to predict student performance using emerging trend Deep Learning approach [8] which is demonstrated to be a very effective method to predict outcomes with a high level of accuracy, especially when large data sets are available. Deep learning algorithm automatically discovers abstraction with the belief that more abstract representations of data tend to be more useful and learns multiple levels of representation. SPPN involves millions of parameters to train, which require massive computation power. The learning can be made more efficient by using a layer-by-layer pretraining phase that initializes the weights sensibly. The pretraining also allows the variational inference to be initialized sensibly with a single bottom-up pass. Thus graphical processing unit (GPU) [9] is being used due to its parallel architecture for the fast execution and training. One of the best advantages of GPU over the Central Processing Unit (CPU) is its lower cost to create parallel threads on blocks due to its efficient hardware implementation, whilst the CPU incurs an overhead to switch to another program. For this reason, the GPU hardware architecture allows the improvement of computational performance in massive data scenarios. GPU seems to be a natural candidate for massive processing of educational data on forecasting applications. To the best of our knowledge, SPPN is the first GPU implemented deep learning system in educational prediction.

In SPPN we use a six layer neural networks to implement deep learning algorithm as shown in Fig.1. Networks consist of 1 input layer, 4 hidden layers and 1 output layer. We learn multi-level representations by greedily pre-training hidden layers of features, one layer at a time, using an unsupervised learning algorithm sparse auto-encoder [10]. It can learn good feature representations and better initialize the network parameters. After the pre-training, neural network is then fine-tuned using back-propagation of error derivative. We train proposed model on a 120,000 students dataset with two Tesla K40 12GB GPUs, and the experimental results show the effectiveness and efficiency of the proposed method which can be applied into academic pre-warning mechanism.

II. Educational Data

Educational data can be collected from multiple sources coming in different formats and granularity. We collected real world data from 100 junior high schools in Hubei province. Each school samples 1200 grade-9 students for recent three years (400 grade-9 students per one year). Grade-9 student will have a high school Entrance Examination, therefore it's meaningful for management to predict entrance examination score and help the students at risk to improve the education quality. As shown in Fig.1, the training data is a composite of different kinds of information:

Background and demographic data: gender, age, health status, family status etc.
Past study data: junior high school entrance score, GPA of primary school etc.
School assessment data: school type, school ranking etc.
Study data: every course score in junior high school(middle-term exam, final-term exam, average)
Personal data: personality, attention, psychology related data, etc.
All collected raw data is transferred into numerical values, then we normalized and scaled the data values by subtracting the mean and dividing by the standard deviation of its elements to make sure that each value varies within the same range. After normalizing each input vector, the entire dataset is whitened [11] to make the input less redundant.

III. Algorithm

After pre-processing, we concatenated student data consisting of different information described in the Section.II into a flatten vector x_1. As shown in Fig.1, x_1 will be the input of the networks. Then an unsupervised learning algorithm sparse auto-encoder is used to discover features from the unlabeled data. We train an auto-encoder for K hidden nodes in hidden layer l using back-propagation algorithm to minimize squared reconstruction error among all m examples with a penalty term that forces the units to output a low average activation:

$$\arg\min_{W^l,b^l} \frac{1}{m}\sum_{j=1}^{m}\left|\left|a^{(l-1)} - recons\left(W^l, b^l, a^{(l-1)}\right)\right|\right|^2 + \lambda S\left(a^l\right) \tag{1}$$

$$recons\left(W^l, b^l, a^{(l-1)}\right) = f\left((W^l)^T f\left(W^l a^{(l-1)} + b^l\right) + b^l\right) \tag{2}$$

where $S(a^l)$ is a sparsity cost function which penalizes layer l output a^l for being far from zero. W^l is layer l weight parameters matrix, b^l is biases, $recons$ is reconstruction function which uses W^l and b^l to construct output with last layer $l-1$ output a^{l-1}. A hidden neuron layer l output is a^l:

$$a^l = f(W^l a^{l-1} + b^l) \tag{3}$$

A Rectified Linear Units(ReLUs)[12] function:

$$f(a) = \max(0, a) \tag{4}$$

is used to model hidden neuron's activity function. We used this unsupervised training method in all hidden layers layerwisely to initialize the whole networks weights W to appropriate values.

In the output layer l_o, we used softmax to make a multiple classification. l_o has 5 output neuron units, each unit corresponds to a final-exam grade, 5 units indicate student final-exam grade $g \in \{O, A, B, C, D\}$ ($O: 90\% - 100\%, A: 80\% - 89\%, B: 70\% - 79\%, C: 60\% - 69\%, D :< 60\%$, final grade is calculated in the format of 100% mark).

After unsupervised pre-training, the whole deep network is subsequently fine-tuned using backpropagation of error derivatives. The recently-introduced technique called dropout [13] is used in training SPPN. Dropout consists of setting

TABLE I
TRAINING PARAMETERS

Parameter	Value
learning rate	0.00025
momentum	0.9
minibatch size	512
weight decay	0.0005

zero to the output of each hidden neuron with probability 0.5. The neurons which are dropped out do not participate to the forward pass and back propagation. Every time the neural network samples a different architecture but sharing same weights. This technique prevent the units from co-adapting too much since a neuron cannot rely on the presence of particular other neurons. Dropout forces to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. Without above techniques, SPPN will suffer from greatly overfitting and be stuck on poor local optima.

SPPN is trained on a GPU platform based on the Nvidia CUDA API. The GPU pipeline is well-suited for parallelism attaining high performances in matrix and vector operations. Unlike CPUs which use the paradigm SISD (Single Instruction Single Data), GPUs are optimized to perform floating-point operations (on large data sets) using the paradigm Single Instruction Multiple Data (SIMD). The parallelism of a GPU is fully utilized by accumulating a lot of input feature vectors and weight vectors, then converting the many inner-product operations into one matrix operation. GPUs enormous computational potential is particularly valuable for neural networks which are complex [9], placing high demands on memory and computing resources. CPUs are simply not powerful enough to solve them quickly in a feasible running time period.

The whole processing is demonstrated in Fig.1.

IV. EXPERIMENT AND RESULTS

SPPN is implemented in ANSI C and Theano [14], a python library that allows transparent use of GPU, and runs on a 2x Intel E5-2680 CPU, 64GB RAM with 2x Nvidia Tesla K40 12GB GPU. Although a single Tesla K40 GPU has 12GB of memory, it's still not enough to have whole net to fit on one GPU over 120,000 training examples. Therefore we spread the net across two K40 GPUs since current GPUs are particularly suitable on cross-GPU parallelization (SLI), which is able to access each another GPUs memory directly without interference with host computer's memory.

SPPN is trained on a about $120,000$ labeled students dataset with training parameters demonstrated in Tab.I. Note that it does not exist a public benchmark or dataset in educational prediction because of the sensitivity and confidentiality, so a direct comparing of different methods is not feasible. Therefore we implemented 3 different classification algorithms: NaiveBayes, Multilayer Perception (MLP) and SVM to compare results with SPPN on our own dataset. The overall accuracy of classifiers performance with correctly classified ratio on our dataset are shown in the Table II. SPPN acquires the best accuracy between those algorithms with the average accuracy 77.2%. Traditional neural networks MLP greatly suffers from substantial overfitting. The other two shallow models SVM, Naive Bayes are not capable to be comparably discriminative as SPPN. The experimental results show that our approach is practical to use in educational setting for identifying particular events such as pre-warning for students at risk.

TABLE II
COMPARISONS OF ALGORITHMS ACCURACY(%)

Algorithm	O	A	B	C	D	Average
Naive Bayes	4.7	15.5	20.8	26.9	35.4	20.7
MLP	33.4	35.6	**53.8**	40.3	44.7	41.6
SVM	**34.7**	**39.8**	42.5	**55.8**	**69.3**	**48.4**
SPPN	<u>86.5</u>	<u>63.2</u>	<u>77.6</u>	<u>70.4</u>	<u>88.4</u>	<u>77.2</u>

<u>underline</u> and **bold** indicate best and second best performance respectively

TABLE III
TRAINING PARAMETERS

	SPPN-g	SPPN-c
training time	376 minutes	3382 minutes
average precision	77.2%	78.4%

SPPN-g is trained on GPU
SPPN-c is trained on pure CPU

We also compare the training efficiency between GPU and CPU, the training time period is shown at Tab.III. SPPN-g is the network trained on GPU, which is about 9 times faster than SPPN-c trained purely on CPU in the training procedure. Although the result of SPPN-c is slightly better than SPPN-g, SPPN-c took almost 2 and half days to train, SPPN-g just took 6 hours. If the training set becomes larger in the future, it would be necessary to use GPU parallel architecture for converging.

V. CONCLUSION

Data mining technologies have been recently used in the education for predicting students academic performance. However measuring academic performance of students is challenging since diverse factors and variables correlate in complicated nonlinear ways. In this study, we present a deep learning architecture for predicting students performance, which takes advantages of unlabeled data by automatically learning multiple levels of representation. We pre-train hidden layers of features layerwisely using sparse auto-encoder, and then use supervised training for fine-tuning the parameters. We train model on a relatively large real world students dataset, and the experimental results show the effectiveness of the propose method. Future work will aim at optimizing the networks

architecture, gathering more training samples, and using temporal information in the sequential data.

ACKNOWLEDGMENT

The research was supported by the National Natural Science Foundation of China (NSFC) via Grant 61402155 and Natural Science Project of Hubei Education Department via Grant B2015024.

REFERENCES

[1] R Baker and George Siemens. Educational data mining and learning analytics. *Cambridge Handbook of the Learning Sciences:*, 2014.

[2] Cristóbal Romero, Manuel-Ignacio López, Jose-María Luna, and Sebastián Ventura. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, 68:458–472, October 2013.

[3] Ji-Wu Jia and Manohar Mareboyana. Machine Learning Algorithms and Predictive Models for Undergraduate Student Retention. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, 2013.

[4] Mariel F Musso, Eva Kyndt, Eduardo C Cascallar, and Filip Dochy. Predicting general academic performance and identifying the differential contribution of participating variables using artificial neural networks. *Frontline Learning Research*, 1(1):42–71, 2013.

[5] Sotiris B Kotsiantis. Use of machine learning techniques for educational proposes: a decision support system for forecasting students grades. *Artificial Intelligence Review*, 37(4):331–344, 2012.

[6] Annika Wolff, Zdenek Zdrahal, Drahomira Herrmannova, and Petr Knoth. Predicting Student Performance from Combined Data Sources. In *Educational Data Mining*, pages 175–202. Springer, 2014.

[7] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[8] Yoshua Bengio. Deep learning of representations: Looking forward. *Statistical Language and Speech Processing*, 2013.

[9] Laszlo Bako, A Kolcsar, S Brassai, L Marton, and Lajos Losonczi. Neuromorphic Neural Network Parallelization on CUDA Compatible GPU for EEG Signal Classification. In *Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on*, pages 359–364. IEEE, 2012.

[10] Adam Coates, AY Ng, and H Lee. An analysis of single-layer networks in unsupervised feature learning. *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[11] a Hyvärinen and E Oja. Independent component analysis: algorithms and applications. *Neural networks : the official journal of the International Neural Network Society*, 13(4-5):411–30, 2000.

[12] V Nair and GE Hinton. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, (3), 2010.

[13] N Srivastava. Improving neural networks with dropout. 2013.

[14] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3, 2010.