Monitoring the Learning Progress In Piano Playing With Hidden Markov Models

Nina Ziegenbein,[1] Jason Friedman,[2] Alexandra Moringen[1]

[1] Neuroinformatics Group, Bielefeld University ,
[2] Department of Physical Therapy, Tel Aviv University
{nziegenbein, abarch}@techfak.uni-bielefeld.de, jason@tau.ac.il

Abstract

Monitoring a learner's performance during practice plays an important role in scaffolding. It helps with scheduling suitable practice exercises, and by doing so sustain learner motivation and a steady learning progress while they move through the curriculum. In this paper we present our approach for monitoring the learning progress of students learning to play piano with Hidden Markov Models. First, we present and implement the so-called *practice modes*, practice units that are derived from the original task by reducing its complexity and focusing on one or several relevant task dimensions. Second, for each practice mode a Hidden Markov Model is trained to predict whether the player is in the *Mastered* or *NonMastered* latent state regarding the current task and practice mode.

Introduction

For decades, a large research field has focused on developing intelligent tutoring systems (ITS) that provide scaffolding to learners in the absence of teachers or experts (Mousavinasab et al. 2021; Almasri et al. 2019). Monitoring a learner's performance as they practice plays an important role in scaffolding (Zydney 2012)[1]. Like most skills that we learn in sports or music, playing piano is a complex skill that consists of numerous subskills that we need to master gradually. These include being able to play correct notes, correct rhythm, being able to play with multiple fingers and, furthermore, coordinate both hands playing different parts of the tune independently. To this end, first, we present multiple practice modes to split up the complexity of a task for the player with a focus on rhythm, pitch or fingering, respectively. Second, we create Hidden Markov Models (HMM) to learn when a practice mode is mastered. This approach is a building block that in the long-run will enable an Intelligent Tutoring System (ITS) to generate an informed practice schedule based on the learner's progress (e.g. as in Mu, Jetten, and Brunskill 2020). Thus, our next goal is to integrate research presented in this paper with a practice schedule optimization that employs Gaussian Processes to provide an optimal practice mode to the learner based on their talents and their skill level (Moringen et al. 2021).

To train the HMMs and evaluate them with respect to a set of practice modes we have conducted a small-scale[2] quantitative study. A video highlighting how the study was run can be found online (vid 2020). For each practice mode, a Hidden Markov Model is trained to predict, given current performance measures and the current latent state, whether the player is still in the *Non-Mastered* latent state or is ready to be moved to the *Mastered* state. Prior to the practice sessions, we allocated a suitable level at which the learners needed to start to practice with exercises, being neither too easy nor too difficult. To this end, the study participants went through a dynamic difficulty adjustment (see Section Dynamic Difficulty Adjustment). This resulted in an assignment of a task on a corresponding complexity level for each study participant. An important feature of our approach is that each HMM is trained for the respective practice mode, but does not take a particular task or complexity level into consideration. With this approach, we need as many HMMs as practice modes which then can be applied to track the learner's state independent of the complexity level or task. In this paper we show preliminary results that we achieved for HMM-training. A more detailed report can be found in the bachelor thesis (Ziegenbein 2021) on which this paper is based.

Related Work

To teach successfully and with the best possible outcome, models for students need to be individualized (Lee and Brunskill 2012) and the student needs to be adequately challenged and promoted or encouraged to achieve maximal engagement (Mu, Jetten, and Brunskill 2020). Learning Systems or Intelligent Tutoring Systems (ITS) are not novel - there exists a variety of different systems (Koedinger et al. 2013). There is research on ITS using Case-Based Reasoning (Soh and Blank 2008) or POMDP (Rafferty et al. 2011), where the learner's knowledge (skill level) is represented in the state. Since we are not only building a computer system,

[1]This paper is part of an already existing project, in which an artificial intelligent tutoring system teaches a novice to play the piano. More information is available under the following links: https://ni.www.techfak.uni-bielefeld.de/node/3637, (Ziegenbein et al. 2021)

[2]Due to Corona the number of study participants was strongly restricted.

but a system that interacts with, learns from and depends on the human user, there are a few constraints. Since it is difficult and costly to generate big data in this domain, we have to learn from a learner's past to predict their future behavior. In addition, we have to be aware that humans are non-stationary and sometimes act irrationally (Brunskill 2018).

Bayesian Knowledge Tracing (BKT) has been shown to be successful in tracing student knowledge in various educational applications (Corbett and Anderson 1994). The BKT model is an HMM with two hidden states "mastered" and "not mastered" and two observable states, "correct" and "incorrect". This model helps to explain that correct answers are not always made because of a mastered state of knowledge but rather might be a "lucky" guess. Conversely, incorrect answers might be caused by a slip in the mastered state and are not evidence for a non mastered knowledge state. In this paper we demonstrate application of this approach to a new domain, learning to play the piano with practice modes.

Practice Modes

Due to its complexity, piano practice is broken down into multiple practice modes as defined in Moringen et al. (2021). Each practice mode is focused on specific aspects a piano player might struggle with. The decomposition into practice modes was chosen based on informal interviews with piano teachers (see acknowledgement in Moringen et al. 2021, for more information).

The idea is that only helpful and necessary practice modes in a limited count will be given to the player, based on their performance or rather their error values. The practice modes used in the study divide the task by only playing with the left hand (*LeftHand*) or only playing with the right hand (*RightHand*), reducing the task to use only one pitch and therefore focusing on rhythm (*PlayRhythm*) and playing the piece slower (*Slower*) with less beats per minute. These modes have been chosen as they cover the timing aspect, as well as fingering, pitch and two-hand coordination. The piece without any adaption or focus is also considered a practice mode and is referenced to as *Identity*.

Curriculum

As with any educational task, it is important to design a curriculum that fits the subject as well as the student. To perform a piece well and play piano at a certain skill level one needs to master prerequisite levels, which in turn might also have prerequisites. Those dependencies are considered as the prerequisite structure, which is built in a tree-like fashion and can be seen in the bottom of Figure 1. The player needs to first master the prerequisite levels before being able to master the levels which depend on them. Since the particular order of levels, as long as the prerequisite dependencies are adhered to, does not matter, the levels are linearly ordered to form a curriculum. In Figure 1, one can see a possible curriculum ordering derived from the prerequisite structure below. The player may move on to the next level when the previous level has been mastered. This order is fixed for every player.

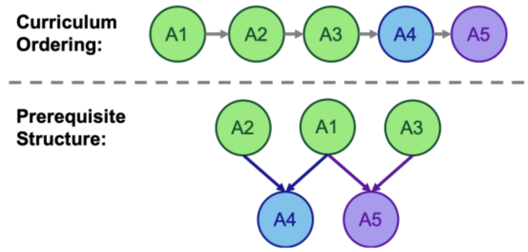

Figure 1: Example of one possible linear curriculum that has been ordered based on the prerequisite structure seen on the bottom. - figure from Mu, Jetten, and Brunskill (2020)[3]

In the case of piano playing, a suitable example would be that a player first plays a level with only the left hand and a level with only the right hand, before moving on to a level in which both hands play together. The single-handed levels would be the prerequisite levels for the both-handed level.

The aforementioned levels are complexity levels in the case of piano playing. The complexity levels are defined on various parameters, for example, note range or values and are learned on the classification of advanced piano pieces by Parmar, Reddy, and Morris (2021). With each level, the complexity increases.

In our approach, there is a variety of scores for each complexity level, as each score is randomly generated based on the task parameters (Moringen et al. 2021), such as rhythmic resolution, number and range of notes, number of fingers, number of notes per bar, number of bars, etc. Each score can be practiced with a multitude of different practice modes with one score-practice mode pair referred to as a task. Each time the player plays a task, it is called a practice opportunity. The player usually plays multiple practice opportunities on the same task, which is referenced to as a sequence. An overview of all these terms can be found in Figure 2.

Dynamic Difficulty Adjustment

Optimal learning by the piano player can be achieved when the player is neither overwhelmed nor unchallenged, but in a state of flow, as described by Csikszentmihalyi (1990). The complexity level that achieves this needs to be individually found for each student, as it should be equal to the current skill level of the student.

To achieve this, Hidden Markov Models are used to predict mastery, which results in giving the next task to the player. In order not to bore the player in the beginning, while going through many tasks that are below the skill level of the player, a dynamic difficultly adjustment is introduced. Dynamic difficulty adjustment is widely used in level-based games (Xue et al. 2017). To find the correct starting level on which the player enters the learning process, they are presented with tasks in practice mode *Identity* and given exactly one practice opportunity for each task. Since this project is generally tailored towards novices, it makes sense to start

[3]The displayed colors bear no meaning in the context of this paper.

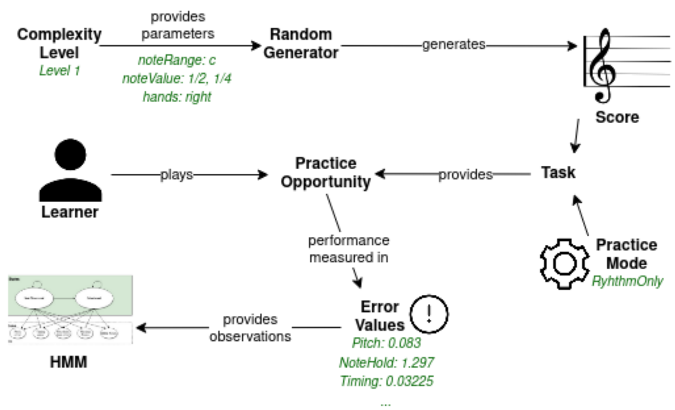

Figure 2: This is an overview figure of how one practice opportunity is generated, played and used to predict the HMM with the corresponding terminology.

with the easiest task. Then, the student subsequently performs tasks of the next highest complexity level, until the performance drops under a certain threshold. At this point the current complexity level is set as the initial complexity level at which the regular training starts with multiple practice modes. The performance will be measured by error values regarding the player's performance, e.g. pitch error, timing error, etc.

Piano Practice HMM

A Hidden Markov Model (HMM) (e.g. Stamp 2004; Lee et al. 2021), is a statistical model that consists of latent states and observable variables. The latent states are not observable, but generate certain observations, which in turn can be used to train the HMM, i.e. generate information on the hidden states and the transitions between them. The transition between hidden states are modeled as a Markov chain.

Let A be the transition probability matrix of the hidden states, and π be the start probability vector for those states. The probability of a certain observation occurring, also called the emission probability, can be any probability distribution with parameters θ. Those can be represented as the observation probability matrix B. A Hidden Markov Model is completely specified by π, A and θ or B.

In order to be able to tutor a player adequately, one needs to know whether a player has mastered a specific task. This is not obvious but needs to be derived from observable and measurable values. Thus, we define a Piano Practice Hidden Markov Model (PPHMM) as shown in Figure 3.

The hidden states should model if a player has mastered a certain task. Thus, two states are needed: *NonMastered* and *Mastered*. Since a mastered task means that the skills of the player now include this task, no transition from state *Mastered* back to state *NonMastered* is possible. Thus, the transition probability between *Mastered* and *NonMastered* should be 0.

The observations of piano playing are performance based measures, in this case the error values. The error values are recorded for both left and right hands and are normalized by

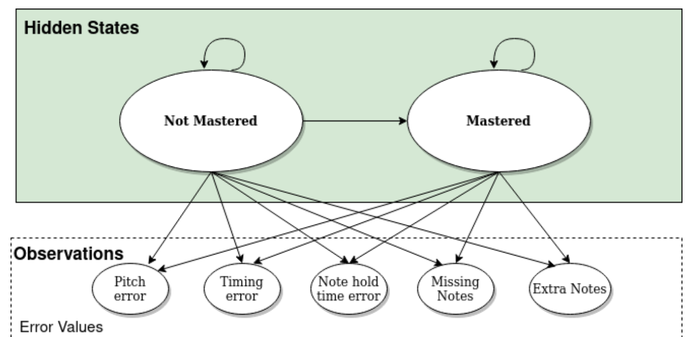

Figure 3: Structure of the PPHMM. The two states encode the current knowledge of the player, while the observations are error values that the player makes while playing.

the number of notes or number of bars. The values describe errors in pitch, rhythm, timing and if the player played more or less notes than required, and can be seen in Figure 3.

Study

To collect suitable data to train the HMM and in turn predict and trace the current knowledge or skill state of the player, a study was conducted with 12 study participants, 1.5 hours each. Due to the corona pandemic the study had to be conducted at a relatively small scale.

The corresponding code can be found at the Github repository on branch *PractiseModes* (Ziegenbein et al. 2021). The technical details on how to run the code to reproduce the study can be found in the README file. A video highlighting the software and the conduction of the study can be found online (vid 2020).

Each participant had time to familiarize themselves with the study equipment and setup, especially the keyboard. Every participant started by going through the dynamic difficulty adjustment, as described above. The piece that was generated at the last complexity level of the dynamic difficulty adjustment stays the same, but is thereafter trained with various practice modes.

Due to the limited number of study participants, the order in which the practice modes are to be trained is fixed. This order should be selected in order to maximize learning success, which is why - based on piano books for beginners (Thompson 1959; Willard A. Palmer 1994; Frances Clark 2002) and interviews with piano teachers - the following order was selected: *Right Hand - Left Hand - Play Rhythm - Slower - Identity*. Complexity levels which do not include both hands will start with practice mode *Play Rhythm*.

Each practice opportunity is rated by the participant and the experimenter on a scale: *very bad - bad - okay - good- very good*. The task is repeated until both the study participant and the study conductor agree that the task is mastered. This is important to make sure that each sequence contains a transition to the *Mastered* state. The study participant continues through all practice modes on that complexity level and then moves on to the next complexity level. This is repeated for roughly 90 minutes for each study participant.

Participants who previously played an instrument performed significantly better. They did not start at a higher complexity level in the dynamic difficulty adjustment compared to the non-musicians, as most of them have not played piano but rather a different instrument. They did, however, master tasks faster, which in turn meant they ran through multiple complexity levels, showing a higher learning velocity. Only one participant played piano in their childhood, which lead to a higher starting complexity level by the difficulty adjustment.

Another interesting thing to observe was the handling of errors. Most knew when they had played a wrong note or made a timing error. Often that led to louder following notes, as the key was pressed faster and harder (i.e. post-error speeding, Paas et al. 2021). It would be interesting to see if the key velocity as a measure can identify errors as well, especially for novices.

Training of Models

All sequences of the recorded study data with the same practice mode were used to train one HMM for that practice mode. This was done with the hmmlearn library (Lee et al. 2021).

To train a Hidden Markov Model based on observations, the Baum-Welch algorithm is used. It is an Expectation-Maximization algorithm, which means that it might end up in a local optimum, instead of the global optimum. Because of this it is important to perform the algorithm multiple times and try different initial values to find the best possible model.

For each practice mode, a separate HMM with Gaussian emissions, a diagonal covariance matrix and two states was created. The models were initialized with a start probability vector π, as well as an initial transition matrix A. A PPHMM with two states has a start probability vector π of (1.0, 0.0), since the player always starts in the *NonMastered* state. The initialization for the 2x2 transition matrix A were varied.

We generated models based on the study data with two states, but only the model for practice mode *Left Hand* was suitable. The models for the rest of the practice modes unfortunately did not bear any merit.

In an attempt to generate more sensible models, we tried to produce simplified data with a better defined division in error values between *Mastered* and *NonMastered* states. The simplified data was recorded by intentionally playing very good or very bad. This lead to a very clear divide in error measures. These practice opportunities were then ordered in sequences similar to those recorded in the study. That means that first bad practice opportunities were given and then the last two or three practice opportunities were good. The number of bad practice opportunities in the sequences ranged from 3 to 13.

This method was very successful as the best performing model predicted all sequences of the simplified data correctly, when trained on all error measures.

Analysis of the Model

The model describe above using simplified data will now be applied to the recorded study data.

The performance of a model is judged by multiple criteria. The transition matrix (TransMatrix) should be reasonable, which means there should be no transition possible from state *Mastered* to state *NonMastered*. The probability to stay in state *NonMastered* should be significantly higher than the probability to transition out of it into the *Mastered* state. Another measure to analyze models are the means of a state (Mean) or the mean observations in a state. In this case, these are the mean error values of those practice opportunities that have been assigned to that state. The means should be consistent, meaning that all error values in state *NonMastered* should be significantly higher than in state *Mastered*. In the study, both the experimenter and the study participants ranked each practice opportunity, which were transferred to a numerical scale with 1 being ‘very good’ and 5 being ‘very bad’. The distribution of these labels were depicted in a beanplot and judged by how distinct the difference in mean between the two states and the general distributions of the states are. This criterion is referenced as LabelDistr below. The HMM is used to predict in which state a person is at each practice opportunity of a sequence, so it only makes sense to look at the predicted states in a sequence as well. Since there is no easily available ground truth there is no way to say if a sequence is correctly classified or not, but it is possible to see how plausible the predicted states are given the way the study was conducted. If the experimenter and the study participant felt that they had mastered a given task, a new task would be issued. That means that the *Mastered* states should be at the end of the sequence and should be the last one to four practice opportunities. The percentage of plausible sequences are listed in the table below and referred to as PredStates.

The model on the simplified data converged to a sensible transition matrix, and means of the states as seen in Equation 1 and Table 1 respectively. Compared to other models generated on the same data it also has the highest score, which is the summed log likelihood over all observations $o \in O$.

$$A_{SimplifiedData} = \begin{bmatrix} 0.8293 & 0.1707 \\ 0.0 & 1.0 \end{bmatrix} \quad (1)$$

state	pitch	hold_time	timing	missing_notes	extra_notes
0	0.2682	3.3372	0.2315	0.1674	0.0455
1	0.0	1.3777	0.0242	0.0	0.0024

Table 1: Means of states of model trained on simplified data

The natural question to ask next is if the model trained on the simplified data would perform well also on the recorded study data. The trained model was applied to predict the states of the recorded study data separated by practice mode. The performance of this model applied to each practice modes in the study data can be seen in Table 2.

While these results are not bad, they are not optimal as became clear when looking at the predicted states. For practice modes *PlayRhythm* and *Slower* not even half of all sequences are plausible. Further training of the existing model on the study data might improve results. This is exactly what has been done: The model trained on the simplified data was taken with all its current parameters and a second training was carried out on the respective study data of each practice mode. When further training on the study data, all models for all practice modes performed worse than the non-further trained models. The performance split up by the different criteria can be seen in Table 2. It becomes apparent that the models, that have been further trained on the study data perform worse than the original model.

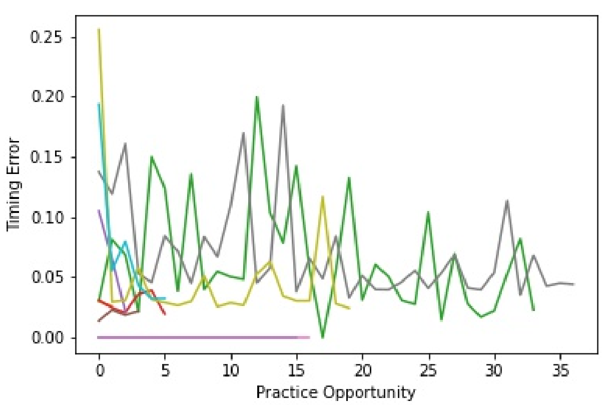

Figure 4: The *Timing Error* over 10 randomly selected sequences of practice opportunities in practice mode *RhythmOnly*.

Another attempt to improve model performance was made by only using a limited amount of error values. For example it seems plausible that for practice mode *PlayRhythm* only the *Timing Error* is needed to make meaningful predictions. This is not the case, as all practice modes performed worse when trained on only a small subset of errors. This might become clearer when looking at Figure 4 which shows a subset of sequences of the *Timing Error*. While the general trend shows a reduction in error, there is a lot of variability between the single practice opportunities. But even a low value does not correspond to a well played performance. This might be because if the player missed a lot of notes, then there than is no timing error on those notes. This would lead to a low *Timing Error*, but would result in a high *Missing Notes Error*. This is why a multidimensional error value is important and necessary for a well-performing model.

While small subsets of error values do not bear any merit, it is important to not use error measures which encode aspects of piano playing which are strategically left out by the used practice mode. For example, the *PlayRhythm* practice mode should not be evaluated with *Pitch Error*, as playing the correct pitch is strategically left out of this practice mode.

Further Improvements

The results from this study lead to the question of how (more) plausible, good-performing models can be generated based on real data, recorded from novices learning to play piano.

The first approach would be to gather more data points, in order to generate more robust models.

There were a few models with a high score, compared to other models on the same data, which suggests that the HMM has found a well-fitting model on some underlying pattern that was not the ones it should have learned. This leads us to conclude that a supervised approach with (hidden)-state labels is necessary and will hopefully yield better results. There is an semi-supervised learning approach to learn Hidden Markov models (Tamposis et al. 2019) which might prove very helpful, as labeled and non-labeled data can be used together to train the HMM. This will reduce the amount of expert hours needed to label the performances. There could be an initial model with labeled sequences, labeled by a piano teacher and then the model can be further personalized while the user is training using unlabeled sequences.

Another idea would be to record more data of the same person to have more data points for an individualized model. If this would work in an unsupervised fashion, it would even work as an application for a lot of people. Although having to put some time in until the individualized learning starts to have merit, if it would train itself it would definitely yield a benefit in the long term.

A different approach would be to take a step back and redefine the error values, which are used as observations. It might make sense to calculate errors for each bar separately. We observed a correlation between key stroke velocity and errors, so taking the velocity into account might make sense. Another idea is to introduce an error measure that counts the number of error clusters or even takes out errors of consequential failure. For example when missing a note and then playing it fast after and therefore also shortening the note after and playing it too late.

Finally, a different model, a Gaussian mixture Hidden Markov model could be used. This has not been done in the scope of this work, as the hmmlearn library does not support learning multiple sequences for Gaussian mixture HMMs.

Further work should include experiments to compare this model with existing Bayesian Knowledge Tracing and Intelligent Tutoring models. It would also be interesting to find

Model	PM	TransMatrix	Mean	LabelDistr	PredStates	Overall
Simplified	RightHand	+	+	+	100%	✓
	LeftHand	+	+	+	100%	✓
	SingleHand	+	+	+	100%	✓
	PlayRhythm	+	+	+	44%	okay
	Slower	+	+	+	48%	okay
	Identity	+	+	+	63%	okay
Further Trained	RightHand	o	o	o	100%	✗
	LeftHand	+	+	+	100%	✓
	BothHands	-	+	o	100%	✗
	PlayRhythm	+	o	-	32%	✗
	Slower	+	+	-	48%	✗
	Identity	+	+	-	58%	✗

Table 2: Summary-table of analysis for the model trained on simplified data and applied to the study data, as well as the model that has been further trained on the study data. A + shows a sensible model based on that criterion, while a - shows an unreasonable model and o encodes a model that works okay, but not well. It becomes apparent that the models, that have been further trained on study data perform worse than the original model.

a way to automatically determine those error values which should not be used to train the HMMs for a particular practice mode.

Summary

The contributions of this work are three-fold: 1) We introduced and implemented various practice modes to focus on different aspects of piano playing. 2) We discussed and implemented a dynamic difficulty adjustment based on a curriculum structure of complexity levels. 3) We conducted a study to record performance data on which we generated Hidden Markov Models to predict the current state of a player. We generated a model for the different practice modes on simplified data and tested them on the data recorded in the study. We analysed the model and showed that it is suitable and performs well. Models trained directly on the study data performed significantly worse. We discussed possible solutions to create better working models.

Further work is necessary to have an accurate model that can predict the current state of the learner, for example, a semi-supervised learning approach or a Gaussian mixture model. Additionally, it would be interesting to see how based on performance one could predict if a certain type of practice mode is even needed and to order the practice modes optimally, by trying out different optimization strategies to give the practice mode next, that yields the biggest learning progress.

Acknowledgement

This work is supported by a Grant from the GIF, the German-Israeli Foundation for Scientific Research and Development.

References

2020. Link to video of using software. https://uni-bielefeld.sciebo.de/s/rDWmlW3UxEuodyC.

Almasri, A.; Ahmed, A.; Almasri, N.; Abu Sultan, Y. S.; Mahmoud, A. Y.; Zaqout, I. S.; Akkila, A. N.; and Abu-Naser, S. S. 2019. Intelligent tutoring systems survey for the period 2000-2018. *International Journal of Academic Engineering Research*, 3: 21–37.

Brunskill, E. 2018. Stanford Seminar - Human in the Loop Reinforcement Learning. https://www.youtube.com/watch?v=nj5t1Q_ANlw.

Corbett, A. T.; and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4): 253–278.

Csikszentmihalyi, M. 1990. *Flow: The psychology of optimal experience*. Harper & Row New York.

Frances Clark, S. H., Louise Goss. 2002. *The music tree - A plan for musical growth at the piano*. Summy-Birchard Inc, english edition edition. ISBN 1-58951-020-8.

Koedinger, K. R.; Brunskill, E.; Baker, R. S.; McLaughlin, E. A.; and Stamper, J. 2013. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3): 27–41.

Lee, A.; et al. 2021. hmmlearn. https://pypi.org/project/hmmlearn/.

Lee, J. I.; and Brunskill, E. 2012. The Impact on Individualizing Student Models on Necessary Practice Opportunities. *International Educational Data Mining Society*.

Moringen, A.; Rüttgers, S.; Zintgraf, L.; Friedman, J.; and Ritter, H. 2021. Optimizing piano practice with a utility-based scaffold. *arXiv preprint arXiv:2106.12937*.

Mousavinasab, E.; Zarifsanaiey, N.; R. Niakan Kalhori, S.; Rakhshan, M.; Keikha, L.; and Ghazi Saeedi, M. 2021. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1): 142–163.

Mu, T.; Jetten, A.; and Brunskill, E. 2020. Towards Suggesting Actionable Interventions for Wheel-Spinning Students. *International Educational Data Mining Society*.

Paas, A.; Novembre, G.; Lappe, C.; and Keller, P. E. 2021. Not all errors are alike: modulation of error-related neural responses in musical joint action. *Social Cognitive and Affective Neuroscience*, 16(5): 512–524.

Parmar, P.; Reddy, J.; and Morris, B. 2021. Piano Skills Assessment. *arXiv preprint arXiv:2101.04884*.

Rafferty, A. N.; Brunskill, E.; Griffiths, T. L.; and Shafto, P. 2011. Faster teaching by POMDP planning. In *International Conference on Artificial Intelligence in Education*, 280–287. Springer.

Soh, L.-K.; and Blank, T. 2008. Integrating case-based reasoning and meta-learning for a self-improving intelligent tutoring system. *International Journal of Artificial Intelligence in Education*, 18(1): 27–58.

Stamp, M. 2004. A revealing introduction to hidden Markov models. *Department of Computer Science San Jose State University*, 26–56.

Tamposis, I. A.; Tsirigos, K. D.; Theodoropoulou, M. C.; Kontou, P. I.; and Bagos, P. G. 2019. Semi-supervised learning of Hidden Markov Models for biological sequence analysis. *Bioinformatics*, 35(13): 2208–2215.

Thompson, J. 1959. *Easiest Piano Course - Part One*. The Willis Music Co.

Willard A. Palmer, A. V. L., Morton Manus. 1994. *Adult all-in-one Course - Lesson, Theory, Technic*, volume 1. Alfred Publishing Co.

Xue, S.; Wu, M.; Kolen, J.; Aghdaie, N.; and Zaman, K. A. 2017. Dynamic difficulty adjustment for maximized engagement in digital games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 465–471.

Ziegenbein, N. 2021. Monitoring The Learning Progress In Piano Playing With Hidden Markov Models. https://uni-bielefeld.sciebo.de/s/6rtoNIGUTQpTgdr.

Ziegenbein, N.; et al. 2021. Learning to play piano. https://github.com/abarch/comp_scaffold/tree/PractiseModes.

Zydney, J. M. 2012. Scaffolding. In Seel, N. M., ed., *Encyclopedia of the Sciences of Learning*, 2913–2916. Boston, MA: Springer US. ISBN 978-1-4419-1428-6.