Measuring Revealed Student Scheduling Preferences using Constrained Discrete Choice Models

Jacob Bailey, University of Illinois, Urbana-Champaign

Jacob Bailey is an graduate student with a focus on computer science education at the University of Illinois at Urbana-Champaign.

Prof. Matthew West, University of Illinois, Urbana-Champaign

Matthew West is an Associate Professor in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign. Prior to joining Illinois he was on the faculties of the Department of Aeronautics and Astronautics at Stanford University and the Department of Mathematics at the University of California, Davis. Prof. West holds a Ph.D. in Control and Dynamical Systems from the California Institute of Technology and a B.Sc. in Pure and Applied Mathematics from the University of Western Australia. His research is in the field of scientific computing and numerical analysis, where he works on computational algorithms for simulating complex stochastic systems such as atmospheric aerosols and feedback control. Prof. West is the recipient of the NSF CAREER award and is a University of Illinois Distinguished Teacher-Scholar and College of Engineering Education Innovation Fellow.

Prof. Craig Zilles, University of Illinois, Urbana-Champaign

Craig Zilles is an Associate Professor in the Computer Science department at the University of Illinois at Urbana-Champaign. His current research focuses on computer science education and computer architecture. His research has been recognized by two best paper awards from ASPLOS (2010 and 2013) and by selection for inclusion in the IEEE Micro Top Picks from the 2007 Computer Architecture Conferences. He received the IEEE Education Society's Mac Van Valkenburg Early Career Teaching Award in 2010, a (campus-wise) Illinois Student Senate Teaching Excellence award in 2013, the NSF CAREER award, and the Univerisity of Illinois College of Engineering's Rose Award and Everitt Award for Teaching Excellence. Prior to his work on education and computer architecture, he developed the first algorithm that allowed rendering arbitrary three-dimensional polygonal shapes for haptic interfaces (force-feedback human-computer interfaces). He holds 6 patents.

Measuring revealed student scheduling preferences using constrained discrete choice models

Abstract

For constrained student resources with large student populations it is often necessary to implement some form of reservation or scheduling system. Examples of scheduled-access resources can include one-on-one tutoring, machine shops or labs, and computer-based testing facilities. For planning and resource scheduling purposes it is important to be able to forecast demand, and thus it is important to understand what drives student preferences for particular scheduling time slots. Measuring these preferences can be challenging, however, for at least the following three reasons. (1) Revealed preferences (what students actually choose) can differ significantly from stated preferences (what they say they will want at a future time), requiring the use of actual scheduling data to infer preferences or utilities. (2) The utility that students derive from particular choices is multifactorial, so that in a computer-based testing facility, for example, students may prefer to take their exam mid-afternoon, but they may also prefer to take it as close to the end of the exam period as possible, and it can be difficult to disentangle these factors. (3) Capacity constraints will frequently lead to many time slots being fully reserved, making it unclear which slots were actually preferred.

This paper presents a general framework for measuring revealed student preferences from actual reservation or scheduling data. This framework is based on the theory of constrained discrete choice modeling, as used in economics for modeling consumer preferences. A multifactorial random utility model (RUM) is formulated for student scheduling preferences and the model is trained on scheduling data using maximum likelihood estimation (MLE) and cross-validated on multiple rounds of training/test data splits.

Results are presented using scheduling data from a computer-based testing facility with approximately 50,000 student reservations over three semesters (Spring 2015 to Spring 2016, inclusive). We show that this measurement methodology can accurately capture student preferences in real-world scheduling data and can successfully separate out time-in-week preferences from time-within-exam preferences. Errors are quantified using both log-likelihood with per-reservation data and root mean square error (RMSE) with data aggregated to the time slot level. We discuss both estimation and simulation algorithms for constrained discrete choice models and discuss how Monte Carlo simulation can be used to obtain uncertainty predictions for predicting expected usage.

1. Introduction

Computer-based assignment systems have been widely adopted in large STEM courses in recent years[3], due to the benefits for both students and instructors, including immediate feedback, online content integration, as well as reduced grading workloads. The development of computer-based assignments with automatic grading allows instructors to carry out more frequent testing in their courses. Educational research indicates that frequent testing leads to better retention than rehearsal methods like rereading notes or previously solved problems[4;5]. In addition to the retention gained, students require repeated practice in order to achieve skill mastery[1].

At the University of Illinois at Urbana-Champaign, we have employed these computer-based assessments for summative assessment by implementing a "Computer Based Testing Facility" (CBTF) similar to one that was previously proposed[8]. A CBTF is a proctored computer lab dedicated to holding the exams and quizzes of various classes. Students are allowed to choose from a range of possible *exam slots* (e.g., 1PM – 2PM on Thursday) throughout their exam period, which typically lasts 3 to 5 days. Exams slots are open for multiple exams, so students from a range of classes may be taking an exam at the same time in the same room. Exams are typically one hour in duration, except for a small number of courses that use two hour exams. For modeling purposes, two-hour exams are treated as two one-hour exams.

Empirically, when choosing exam times our students do not uniformly distribute themselves. Figure 1 shows that there is significant variation in the number of exams taken on a day compared to others. Figure 2 shows that there is significant variation in the number of exams taken at a time during a week; each of the seven saw teeth regions represent a different day of the week, each with its own unique distribution.

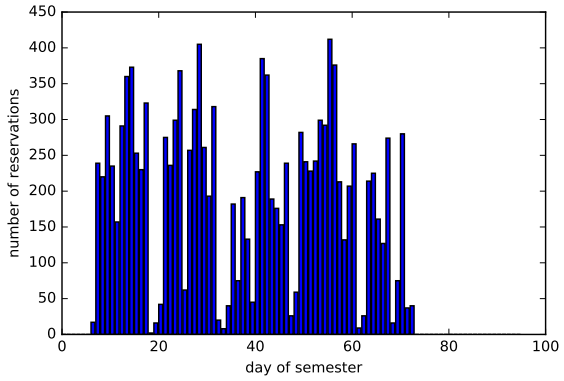


Figure 1: Reservations by day of the semester.

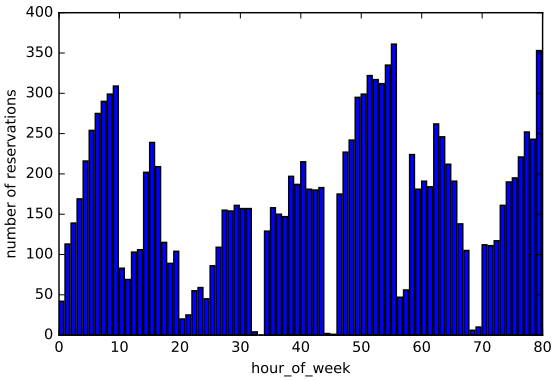


Figure 2: Reservations by day/time during a week.

Given this non-uniformity, it is useful to be able to model the choices students make in selecting exam slots. Having a model that describes this behavior has, for the past 3 semesters, allowed us to better schedule exams (i.e., prevent slots from becoming too popular, or have too much interference from other classes), allowed us to predict the number of proctors needed for slots

ahead of time based on their popularity, and allowed us to reason about the implications of the opening and closing times of CBTF each day.

In Section 2, this paper presents a model, based on discrete choice theory[6]. The model allows us to predict the likelihood that a given student selects each time slot when making an exam reservation. It does so by estimating the *utility* associated with each time slot for a given exam. The two factors used in the model were chosen by studying patterns in the reservation data. First, students' preferences vary with the *time-and-day-of-week*; generally students prefer time slots later in the day, as seen by the daily ramps in Figure 2. Second, students generally prefer taking exams on days later in the exam period (*day-within-exam-period*). By observing past student reservation choices, we can estimate the utility for an "average student" of different choices in each of these dimensions. Having computed these utilities, we can predict student behavior in novel situations through simulation by assigning students to slots in proportion to these observed utilities, modulo availability.

Beyond describing our discrete choice model, this paper makes three contributions:

- We describe (in Section 3) a *fluid limit* simulation algorithm that computes predicted exam slot utilizations from the model and the number of students taking each exam sufficiently efficiently to be interactive.
- We demonstrate (in Section 4) that a model that uses both time-of-day and day-within-exam-period components predicts student preferences significantly better than a uniform distribution or a model that includes only one of the components.
- We demonstrate (in Section 5) through cross validation that the model can effectively predict data that it hasn't previously seen.

In Section 6, we show a visualization that is part of our interactive tool for scheduling exams. We conclude in Section 7 with a discussion of opportunities for future work.

2. Model

Our discrete choice model is based on work presented in a previously-published Work-in-Progress paper[7]. We include here a description of the model with small improvements for clarity.

As noted above, the model makes use of two variables: the number of days left in the exam period and the time and day of the exam slot. The model associates a distinct utility for every exam slot during the week (i.e., each ¡day, time¿ combination). Independently, the model associates a utility associated with the (integer) number of days remaining between an exam slot and the end of the exam period.

These utilities are defined as

$$V_{ni} = \sum_{h=1}^{N_h} \beta_h x_{ih} + \sum_{r=1}^{N_r} \lambda_r w_{nir}, \tag{1}$$

where V_{ni} represents the utility of decision n for a given exam slot i. β_h is the utility of hour h, and x_{ih} is either 0 or 1, depending on if slot i is at hour h. λ_r is the utility of an exam slot with r days remaining in the exam period, and w_{nir} is either 0 or 1, depending on if slot i has r remaining days for decision n. This definition is done as a summation such that the generation of all of V can be done at once, using x and w to "filter out" the correct utilities for each element.

While it is likely that each student values the time slots differently, the model attempts to estimate the utility of time slots as perceived by the "average" student. By computing these average utilities from the whole student population, we can simulate the behavior of the whole population by probabilistically assigning students to exam slots in proportion to these utilities. Specifically, during simulation, these utilities are used to pick an exam slot from a list of available slots. As slots fill up, they are no longer available for additional reservations. As such, the model is a discrete choice model with capacity constraints[2].

The probability that decision n chooses slot i is then

$$p_{ni} = \frac{a_{ni} e^{V_{ni}}}{\sum_{j=1}^{N_i} a_{nj} e^{V_{nj}}}, \tag{2}$$

where N_i is the number of slots and a_{ni} is 1 if slot i is available when student n is making their reservation.

2.1. Computing Utilities

Given a data set of student reservations, the utilities β and λ can be computed by finding their values that maximize the likelihood of producing that data set.

The log-likelihood, L, of producing a given assignment of student reservations given a set of utilities is

$$L = \sum_{n=1}^{N_n} \sum_{i=1}^{N_i} y_{ni} \log p_{ni}, \tag{3}$$

where y_{ni} is 0 or 1, depending on if slot i was chosen in decision n in the data set. In our case, this was done with numpy's BFGS optimizer, where the argument being optimized was an array containing both β and λ.*

Figures 3 and 4 show representative utilities for days remaining in exam period (λ) and by day and time of week (β), respectively.

*The variables actually optimized in our implementation force one value in both β and λ to be zero. This is valid since we can define a "zero" value in each, and the relative utility of each possibility is preserved. This also reduces the number of variables being optimized by two, which helps decrease the runtime at least a little.

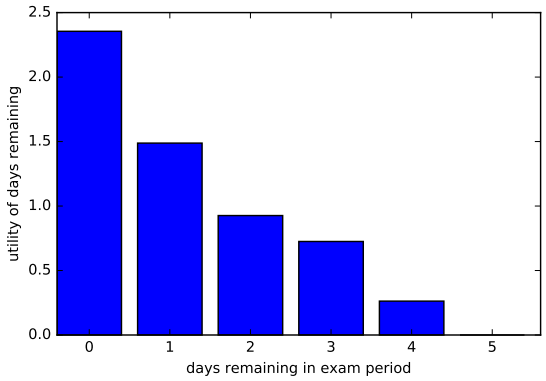


Figure 3: Utilities by days remaining in an exam period.

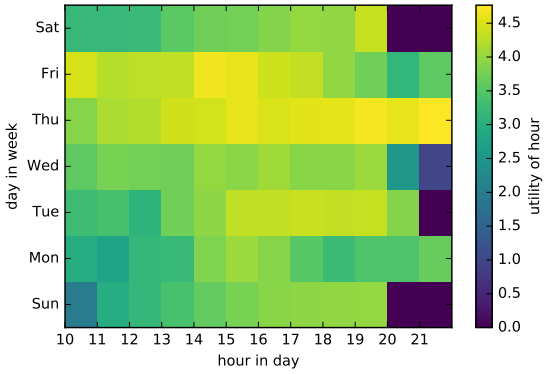


Figure 4: Utilities by day and time in a week.

3. Fluid-Limit Simulation

The primary application of the model is to predict utilization of a set of exam slots for a given set of exams. Exam slots are specified by their date and time and their capacity; one exam slot is allocated for each distinct hour that the CBTF will be open during the semester.

Exams are specified by: (1) which day of the semester the exam period starts, (2) the number of days in the exam period, (3) the number of students who need to take it, and (4) the length of the exam in slots[†].

Because multiple exams are scheduled in the CBTF in parallel, we find it useful to additionally define the concept of a *group*. A group is a set of students that are all taking a specific exam (e.g., "Course 3 Exam 5"). Each group has a given size (as specified by the exam) and a set of utilities associated with each exam slot (as per the average student assumption in Section 2), which we write as V_{gi}, the utility of the group g picking slot i. We also define *group feasibilities* f_{gi}, which is 1 if group g can take an exam in slot i, or 0 if it cannot.

The group feasibilities are generated with the previously described exam specifications. Additionally, we define capacity fractions C, which outline the capacity of the CBTF for each slot (e.g., 48 students from 10AM to 10PM, except on weekends where the CBTF is closed after 6PM, and over breaks when no exams should be scheduled).

Given our assumption that students select exam slots probabilistically and uniformly based on the model's utilities, we can efficiently perform the simulation by treating the groups of students as a continuous fluid rather than discrete individuals. This approach "pours" a group of students into exam slots in proportion to the probabilities that students in that group would choose those exam slots.

Specifically, our *fluid limit* algorithm (Algorithm 1) computes an aggregate rate that each exam

[†]We currently support 1 hour and 2 hour exams during the semester. For the purpose of the model, we treat N students taking a 2-hour exam as $2N$ students taking a 1-hour exam. Given that 2-hour exams are a minority of our exams, we haven't found this to significantly affect the predictive power of the model.

slot fills based on the contributions from each group, assuming that each group fills slots at a rate proportional to the size of the group. With these aggregate rates, we can compute which slot fills to capacity first and at what percentage of the student reservations that occurs. At that point, the aggregate rates can be re-computed with the updated slot availability and the process can be repeated until all of the reservations have been made.

This algorithm can be extended to emit exact counts for each slot.

Algorithm 1 Expected-value "fluid limit" simulation algorithm

Input parameters: group feasibilities f, group fractions H, group utilities V, and capacity fractions C
$\tilde{p}_{gi} \leftarrow 0$ for all $g = 1, \ldots, N_g$ and $i = 1, \ldots, N_i$
$a_{gi} \leftarrow f_{gi}$ for all $g = 1, \ldots, N_g$ and $i = 1, \ldots, N_i$
$S_i \leftarrow C_i$ for all $i = 1, \ldots, N_i$
$t \leftarrow 0$
repeat
  $p_{gi} \leftarrow \dfrac{a_{gi} e^{V_{gi}}}{\sum_{j=1}^{N_i} a_{gj} e^{V_{gj}}}$ for all $g = 1, \ldots, N_g$ and $i = 1, \ldots, N_i$
  $D_i \leftarrow \sum_{g=1}^{N_g} H_g p_{gi}$ for all $i = 1, \ldots, N_i$
  $\Delta t_i \leftarrow \begin{cases} S_i / D_i & \text{if } D_i > 0 \\ 2 & \text{otherwise} \end{cases}$ for all $i = 1, \ldots, N_i$
  $j \leftarrow \underset{i \in \{1, \ldots, N_i\}}{\operatorname{argmin}} \Delta t_i$
  $\Delta t \leftarrow \Delta t_j$
  if $t + \Delta t > 1$ **then**
    $\Delta t = 1 - t$
    $t \leftarrow 1$
  else
    $t \leftarrow t + \Delta t$
    $a_{gj} \leftarrow 0$ for all $g = 1, \ldots, N_g$
  end if
  $\tilde{p}_{gi} \leftarrow \tilde{p}_{gi} + \Delta t \, p_{gi}$ for all $g = 1, \ldots, N_g$ and $i = 1, \ldots, N_i$
  $S_i \leftarrow S_i - \Delta t \, D_i$ for all $i = 1, \ldots, N_i$
until $t = 1$
Output group choice fractions $\tilde{p}$

The algorithm is fast enough for interactive use. Simulations for a whole semester involving tens of thousands of reservations can be completed in about a second on a commodity laptop. [‡]

[‡]While simulating a semester takes about a second, computing the utilities for the simulator as described in Section 2.1 takes about 20 minutes on the same machine. Thankfully, this only needs to be done once to allow the simulator to work.

4. Model Accuracy

We first validate that our model performs better than a uniform distribution over the slots (i.e., by simply making all slots equally as likely to be chosen) and that the two variable model (*time-and-day-of-week* and *day-within-exam-period*) performs better than models with only one of those parameters.

4.1. Experimental Method

For this test, we used data from Fall 2015, with various "odd" exams removed (like one-off or makeup exams). The data contains all of the reservation data for that semester, including which exam was taken (the "group"), which exam slots were available when the decision was made, and which of those slots were actually chosen.[§] For cases where students change their reservations, we considered only their final reservation.

Four models were considered: the two-component model (the "full model") that is the focus of this paper, a model that only considered *time-and-day-of-week*, a model that only considered *day-within-exam-period*, and a uniform model. For each of the one-component models, the model was trained using only that component and simulated using zeroes for the other component. The uniform model was implemented by setting all of the utilities to the same value.

We simulated all of the models using the fluid-limit algorithm and compared the results against the true reservation data from Fall 2015.

4.2. Results

Table 1 shows the errors per slot in terms of the number of reservations compared to the actual reservations seen, while Table 2 shows the error per day. Results are presented as root-mean-square-error (RMSE), mean-average-error (MAE), and median-average-error (Median AE).

Given true data x and predicted data $\hat{x}$, we define RMSE, MAE, and Median AE as

$$\mathsf{RMSE}(x, \hat{x}) = \sqrt{\frac{\sum_{i=1}^{n} (\hat{x}_i - x_i)^2}{n}} \tag{4}$$

$$\mathsf{MAE}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} |\hat{x}_i - x_i| \tag{5}$$

$$\text{Median } \mathsf{AE}(x, \hat{x}) = \mathsf{median}|\hat{x} - x|. \tag{6}$$

[§]Some filtering needs to be done if a proctor forces a student into a slot that wouldn't normally be available, which can occur if a student misses an exam and takes an empty seat at another time. Not doing so means that the model encounters a decision for which there is no available exam slot, which then brings (rightfully) the likelihood of that condition to zero.

Table 1: Errors per slot.

	Uniform distribution	Days Remaining Only	Hour Only	Full Model
RMSE	9.567	8.748	7.075	5.300
MAE	6.526	5.564	4.575	3.237
Median AE	4.304	2.981	2.647	1.658

Table 2: Errors per day.

	Uniform distribution	Days Remaining Only	Hour Only	Full Model
RMSE	61.58	50.56	52.54	31.76
MAE	39.54	34.44	35.78	17.79
Median AE	22.38	26.22	27.40	5.81

By these errors, the proposed two-component model (the "full model") produces the least error out of those tested. The uniform distribution performs the worst, with the proposed two-component model's individual factors performing somewhere in the middle.[¶]

Figure 5 shows a visual comparison of the four outputs for each exam slot in the week, summed over all weeks in the semester. The *uniform* model is very even per day, but varies by day because some days (e.g., Thursdays) had more exams that could be taken that day. The *day-within-exam-period*-only model is similarly very even during each day, but more accurately predicts each day's load. The *time-and-day-of-week*-only model shows more accurately shows the trends within each day, but doesn't correctly allocate load by day. For example, too much load is allocated to Thursday causing it to run at full capacity starting earlier in the afternoon. The proposed two-component model (the "model predicted" panel) combines the two to get something even closer to the true data.

[¶]As a side observation: *time-and-day-of-week* seems to better predict the *errors per slot* overall, while the *day-within-exam-period* seems to better predict the number of students taking an exam in a given day.

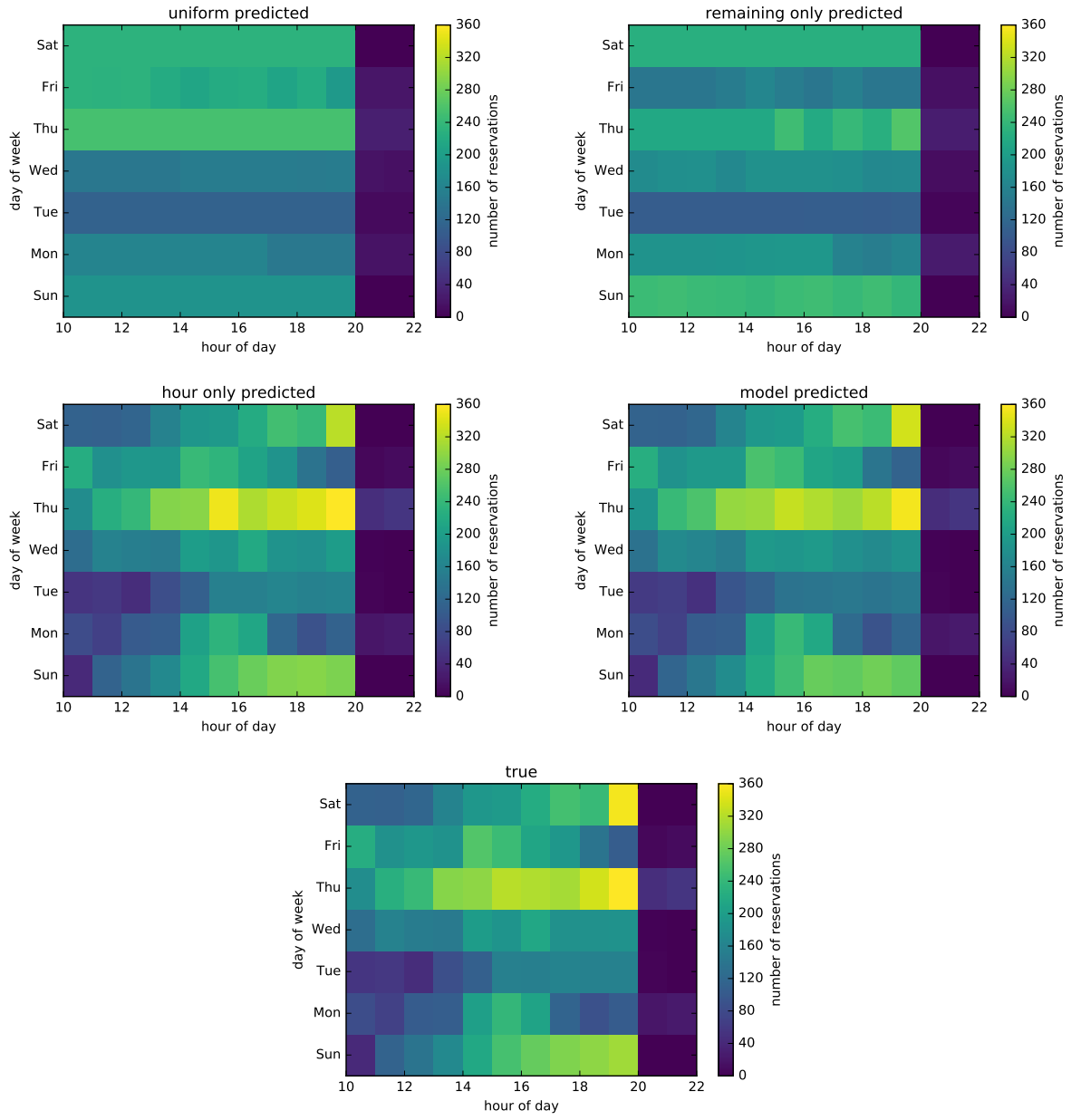


Figure 5: A visualization of the four models tested in Section 4, with the true reservations.

5. Cross validation

In the previous section, we demonstrated that the two-component model provides significant accuracy improvements over alternative simpler models. That experiment was performed using the same semester's data for both the training and test set due to methodological challenges (e.g., every semester our CBTF has been open slightly different hours in the week). One should always be suspect of using the training set as the test set, as there is the potential for over-fitting the training set, especially as model complexity is increased.

5.1. Experimental Method

While it may seem straight-forward to split a semester's data into a separate train and test inputs (e.g., using the first half of the semester as the train input and the second half of the semester as the test input), the continuous overlapping of exam periods means that some exams would span the test/train boundary, causing the two data sets to not be independent.

As a result, we chose to use a cross-validation strategy where we randomly assign 70% of the student reservations to the train set and the remaining 30% to the test set. When we pre-process the data before training, we compute the exam slots that are available at the time the reservation was made. In this way, all of the training examples are independent, so we can train from any subset. When we test, we reduce the capacity in each exam slot by the number of training examples that selected that slot. In this way, a slot that filled in the original data set will fill in our simulation when the same number of reservations from the test set are assigned to that slot as actually chose that slot.

For the data that follows, sixteen randomized 70-30 train-test splits were done on the Fall 2015 data, processed in the same way as the previous section. Each split was trained against its unique train data, then tested against its test data. For each split, the uniform model was also tested. Like in the previous section, the fluid-limit algorithm was used to simulate the decisions based on the optimized utilities. The errors were then compared with averaging and standard deviations.

5.2. Results

Table 3 and Table 4 show the results of the tests. The error is similar to that of the previous section, accounting for the reduced number of decisions (and therefore slot usages) and has a low deviation, which means that the model does not over-fit the data and mispredict on non-training data. Again, each error metric shows that the model performs better than a uniform model. The fact that MAE is higher than Median AE suggests that there are still some outliers which may be better predicted by extending the model.

Table 3: Errors per slot over multiple train-test splits.

	Uniform distribution	Model
RMSE	3.349 (stddev 0.04454)	2.930 (stddev 0.1027)
MAE	2.254 (stddev 0.02226)	1.768 (stddev 0.04346)
Median AE	1.437 (stddev 0.08095)	0.9138 (stddev 0.04186)

Table 4: Errors per day over multiple train-test splits.

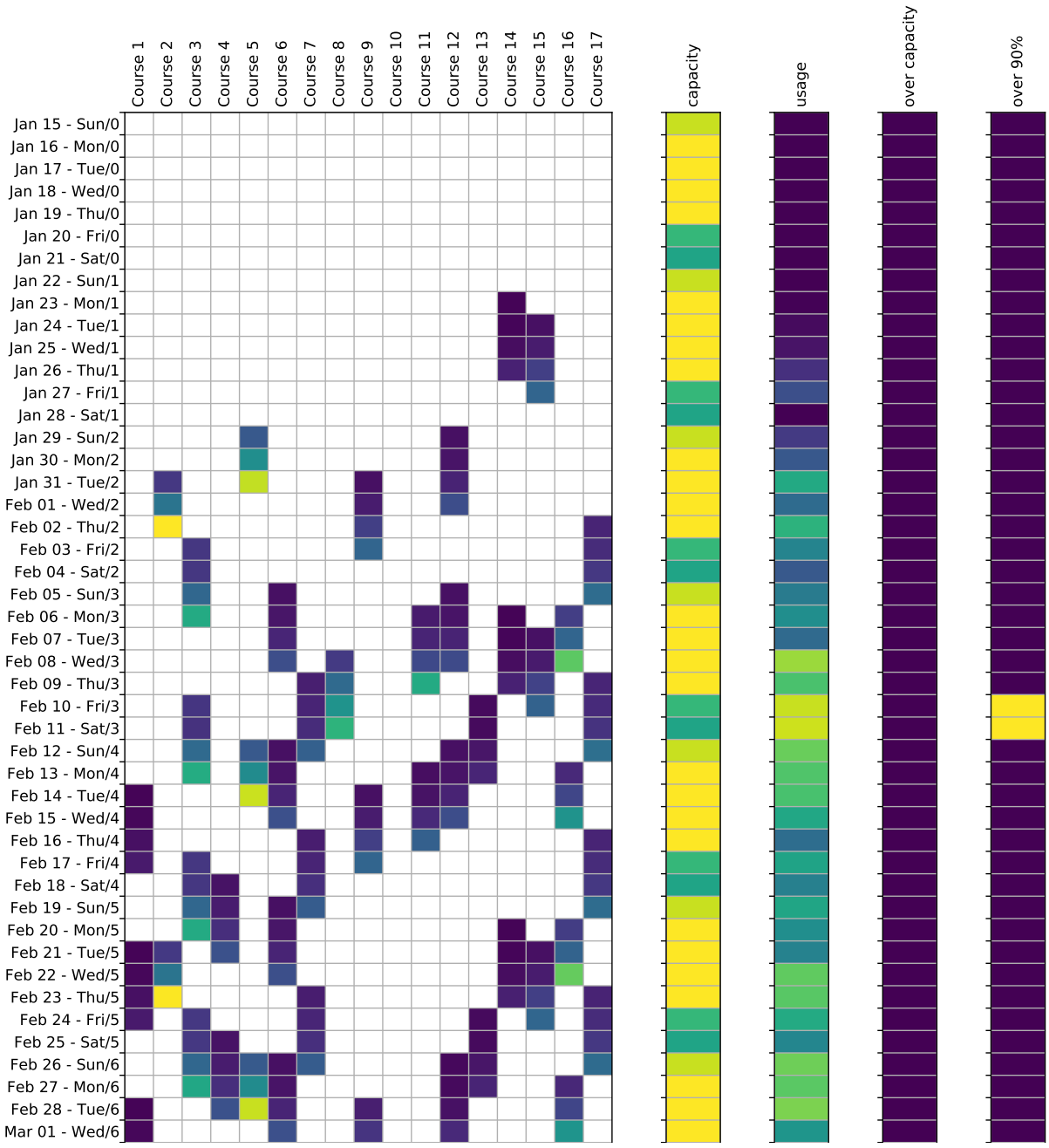
	Uniform distribution	Model
RMSE	19.12 (stddev 0.5253)	13.74 (stddev 0.7631)
MAE	12.35 (stddev 0.3037)	7.749 (stddev 0.4301)
Median AE	6.992 (stddev 0.8363)	2.939 (stddev 0.4564)

6. Visualization

Figure 6 shows the visualization of the model's simulation for the Spring 2017 semester, trained on data from Fall 2016. Our CBTF's administrators modify the exams schedule until no exam slot is predicted to be over capacity and as few slots as possible are over 90% capacity.

The Y-axis labels the date, including a 0-indexed week number. White spaces represent dates without exams. Dark spaces have less usage, while lighter spaces have more.

Figure 6: Spring 2017 predictions based on Fall 2016 data.



7. Conclusion and Future Work

Through multiple semesters of use, this model, using discrete choice with utilities based on the number of remaining days in an exam period and the time and date of a slot, has proven effective at predicting students choices of exam slots. Coupled with the fluid-limit algorithm, it provides an interactive capability for capacity planning at our CBTF (Computer-Based Testing Facility). The efficiency of the fluid-limit algorithm is derived from the fact that it doesn't have to consider each student's decision individually, instead treating them as a deterministic, homogeneous continuum.

This determinism is, however, also a shortcoming of the algorithm. We know that, while the model is predictive, the students aren't going behave exactly as the model predicts. We'd like to be able to compute error bars on the predicted utilization and probabilities of over-subscribing a given day. This is not straight-forward with a deterministic simulation.

An alternative to the fluid-limit algorithm is to use a discrete choice simulation and allocate, for example, individual students to exam slots randomly using probabilities derived from the model. With such a non-deterministic approach, Monte Carlo methods could be used compute the aforementioned error bars and probabilities.

Additionally, the model could be expanded and refined to handle special cases, like final exams. While the proposed model produces good results for exams during the semester, the behavior of students around finals seems to be different. Specifically, some students appear to prefer to take exams as early as possible such that they can leave campus early, or prefer a weekend time when no traditional exams are running (traditional final exams aren't scheduled on weekends, but students can choose to take exams in the CBTF on those days). Similarly, student behavior may differ around breaks, as students may want to start break early or not take exams so close to when classes resume. These behaviors may be captured within the variables we're already considering, but new factors may be required to describe them.

Acknowledgements. This work was supported by the College of Engineering at the University of Illinois at Urbana-Champaign as part of the Strategic Instructional Initiatives Program (SIIP), as well as by the National Science Foundation (NSF) awards DUE- 1347722 and CMMI-1150490.

References

[1] J. R. Anderson. *Learning and memory: An integrated approach*. John Wiley and Sons, second edition, 2000.

[2] A. de Palma, N. Picard, and P. Waddell. Discrete choice models with capacity constraints: An empirical analysis of the housing market of the greater Paris region. *Journal of Urban Economics*, 62(2):204–230, 2007. doi: 10.1016/j.jue.2007.02.007.

[3] K. A. Lack. Current status of research on online learning in postsecondary education. Ithaka S+R, 2013.

[4] H. L. Roediger and A. C. Butler. The critical role of retrieval practice in long-term retention. *Trends in Cognitive Sciences*, 15:20–27, 2011.

[5] D. Rohrer, K. Taylor, and B. Sholar. Tests enhance the transfer of learning. *Journal of Experimental Psychology Learning Memory and Cognition*, 36:233–239, 2010.

[6] K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, second edition, 2009.

[7] M. West and C. Zilles. Modeling student scheduling preferences in a computer-based testing facility. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, pages 309–312, 2016. doi: 10.1145/2876034.2893441.

[8] C. Zilles, R. T. Deloatch, J. Bailey, B. B. Khattar, W. Fagen, C. Heeren, D. Mussulman, and M. West. Computerized testing: A vision and initial experiences. In *2015 ASEE Annual Conference & Exposition*, 2015. doi: 10.18260/p.23726.