Deep Queue-Learning: A Quest to Optimize Office Hours

Avoy Datta, Dian Ang Yap, Zheng Yan
CS 229 | Autumn 2018

Introduction

• OHs often suffer from overcrowding and long wait times, stressing both students and instructors.

• If we could accurately predict the expected workload at a given OH, TAs can be better allocated.

• QueueStatus, Carta, and course syllabi provide a wealth of information that can be used.

• We trained a neural network model that predicts student load influx (expected serve time * # sign-ups) at OH on an hourly basis, for any course.

• With these predictions, we now optimize TA scheduling given realistic constraints.

Methodology

• We defined a new loss (Shrug loss) and used smoothing on labelled data to reduce penalty on outliers.

\[ L_I = \begin{cases} \frac{1}{2} (y - \hat{y})^2, & \text{if } |y - \hat{y}| \leq \delta \\ \sqrt{|y - \hat{y}| - \frac{1}{2} \delta}, & \text{otherwise} \end{cases} \]

• To reduce contributions by outliers, we smooth out the data through convolution with a Hann window:

\[ w(n) = \frac{1}{2} \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \]

• Both classification (SVMs, Random Forest) and regression (fully connected nets, LSTMs) were experimented for predictions.

• Shrug yields significantly lower RMSE on test set, but poorer convergence during training.

Experiments

• Using data scraped off of Stanford course resources, a fully connected NN, and Gibbs sampling, we have come up with a system that schedules TA hours (within realistic constraints) that appears to correlate well with student demand.

• Major challenges for inference: figuring out a model that balanced bias with variance and coming up with a loss that didn't penalize outliers excessively.

• This model can serve as a recommender system for office hours for newly introduced courses. We tested it on one quarter of a course not used in the train set and found correlation between assigned hours and predicted influx were similar to actual load influx and server correlation.

Scheduler

• We use Gibbs Sampling to assign TAs to each individual time slot.

The Gibbs sampler optimizes:

\[ P(X_j = x) = T_{\text{assigned}} \cdot T_{\text{predicted}} | X_j = x \]

\[ T_{\text{assigned}} \cdot T_{\text{predicted}} \text{ measures the cosine similarity between the number of TAs assigned each office hour and the predicted loads} \]

• Weight of sampling is proportional to increase in cosine similarity of the full assignment for each value assigned.

Results

• Smoothing reduces spikes in erratic data.

• Less effective in predicting actual magnitude.

• Shrug yields significantly lower RMSE on test set, but poorer convergence during training.

Cosine similarity, actual schedule

0.794

Cosine similarity, optimized schedule

0.789

Features and Preliminary Statistics

• Load influx is significantly and positively correlated with:

  - Week number \((r = 0.07)\) and Number of servers \((r = 0.32)\)

  - Significantly and negatively correlated with:

  - Days left until assignment due \((r = -0.08)\), Hour of day \((r = -0.10)\), Weekday \((r = -0.09)\), Days until next exam \((r = -0.06)\)

Class Statistics

<table>
<thead>
<tr>
<th>Class</th>
<th>Quarter &amp; Year</th>
<th>#OH-Active TAs</th>
<th># Students</th>
<th>Total OH Hours</th>
<th>Total Served</th>
<th>Total Load Influx</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS107</td>
<td>Spring 2017</td>
<td>15</td>
<td>184</td>
<td>415</td>
<td>1722</td>
<td>21873.09</td>
</tr>
<tr>
<td>CS161</td>
<td>Spring 2017</td>
<td>6</td>
<td>93</td>
<td>204</td>
<td>875</td>
<td>15380.68</td>
</tr>
<tr>
<td>CS110</td>
<td>Spring 2018</td>
<td>20</td>
<td>187</td>
<td>223</td>
<td>1174</td>
<td>35459.1</td>
</tr>
<tr>
<td>CS229</td>
<td>Autumn 2018</td>
<td>17</td>
<td>634</td>
<td>359</td>
<td>1390</td>
<td>31733.7</td>
</tr>
</tbody>
</table>

Summary

• Using data scraped off of Stanford course resources, a fully connected NN, and Gibbs sampling, we have come up with a system that schedules TA hours (within realistic constraints) that appears to correlate well with student demand.

• Major challenges for inference: figuring out a model that balanced bias with variance and coming up with a loss that didn’t penalize outliers excessively.

• This model can serve as a recommender system for office hours for newly introduced courses. We tested it on one quarter of a course not used in the train set and found correlation between assigned hours and predicted influx were similar to actual load influx and server correlation.