Mining Student Time Management Patterns in Programming Projects

Dale E. Parson and Allison Seidel
Kutztown University of PA
FECS’ 14 (#FEC2189)
Outline

• The problem – What are the distinguishing attributes of poorly performing programming students?
• Related work.
• Data collection using student make actions.
• Analysis finds correlations to procrastination, length of work sessions and time of day.
• Some students can procrastinate successfully.
• Future work includes an automated warning system that may take the form of a game.
The Problem

• Computer science professors cite procrastination as a primary cause of poor student performance in programming assignments.
  – However, some procrastinators can do very well.

• There are many conflicting demands on student time. What other factors contribute?
  – Other course projects & exams, jobs, teams/clubs.
Approach for spring 2013 Java Programming

• Programming projects use **make** to compile, to test, and later to turn in student work.
• A student can type “make” many times while repairing compilation and testing errors.
• Make actions capture details of a student’s project directory & the source file contents.
• Participation is voluntary, no grade effect except for 2 bonus points. Data analysis occurs only after end of semester and grading.
Related work

• Edwards (2009) analyzed projects across 5 years, found correlation between procrastination & poor results, but did not consider consistently performing students.
  – Ours captures more work detail for all students.

• Other cited studies find weak correlations to procrastination or number of lines of code changed per session, with very little detail.
Data archives, listings, source files

idN_2014-02-12-12-38-48-EST_BUILD.zip
idN_2014-02-12-12-41-50-EST_BUILD.zip
idN_2014-02-12-12-41-53-EST_BUILT.zip
idN_2014-02-12-12-41-53-EST_TESTING.zip
idN_2014-02-12-12-41-55-EST_TESTED.zip

TESTED 2013-03-24-21-45-38-EDT
~idN/JavaLang/FillWord2
idN 5836 Mar 3 15:43 FillWordTest.java
idN 1183 Mar 13 17:33 IFillWord.java
idN 3474 Mar 14 12:59 FillWordGrows.java
idN 8849 Mar 24 21:33 FillWordBasic.java
idN 5027 Mar 24 21:37 FillWordHelper.java
91 Data Attributes Collected

• Temporal data include date and time of day of work, session length, time between sessions, distance of first and final work session from handout and deadline, dates minus missed days.

• Size data include file numbers and sizes, bytes & lines added/changed/deleted per file.

• Student ID, year & track, conflicting project & exam survey, categories of emails to professor, project grade and centile rank.

• Incoming CS GPA and number of credits.
Finding attributes:
Basic OneR projections

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jstr</td>
<td>hours from student start until due deadline</td>
</tr>
<tr>
<td>Mavg</td>
<td>average minutes of a work session</td>
</tr>
<tr>
<td>Mdev</td>
<td>sample standard deviation of Mavg</td>
</tr>
<tr>
<td>Snum</td>
<td>number of work sessions</td>
</tr>
<tr>
<td>Mtot</td>
<td>total minutes spent on the project</td>
</tr>
<tr>
<td>Cgpa</td>
<td>computer science GPA at semester start</td>
</tr>
<tr>
<td>GprjRank</td>
<td>centile ranking of project grade</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{If } Jstr < 24.0 & \text{ then } 0 \leq \text{GprjRank} \leq 13.2 \\
\text{Elseif } Jstr < 79.0 & \text{ then } \text{GprjRank} > 90.4 \\
\text{Elseif } Jstr < 181.0 & \text{ then } 80.7 < \text{GprjRank} \leq 90.4 \\
& (36/111, 32.4\%, \text{instances correct}) \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>Gprj</th>
<th>project grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Cgpa &lt; 2.24 then 0 &lt;= Gprj &lt;= 0.625</td>
<td></td>
</tr>
<tr>
<td>ElseIf Cgpa &lt; 2.635 then 0.625 &lt; Gprj &lt;= 0.815</td>
<td></td>
</tr>
<tr>
<td>ElseIf Cgpa &lt; 2.945 then 0.815 &lt; Gprj &lt;= 0.925</td>
<td></td>
</tr>
<tr>
<td>ElseIf Cgpa &gt;= 2.945 then 1.015 &lt; Gprj &lt;= 1.04</td>
<td></td>
</tr>
<tr>
<td>(28/111, 25.2% instances correct)</td>
<td></td>
</tr>
</tbody>
</table>
Simple K-means clusters for 6 most predictive attributes for grade

<table>
<thead>
<tr>
<th>attribute</th>
<th>full data</th>
<th>cluster 0</th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
<th>cluster 4</th>
<th>cluster 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 records</td>
<td>1 records</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jstr</td>
<td>167.95</td>
<td><strong>57.88</strong></td>
<td>155.22</td>
<td>249.78</td>
<td>238</td>
<td><strong>36.19</strong></td>
<td>245.25</td>
</tr>
<tr>
<td>Mavg</td>
<td>58.15</td>
<td><strong>68.51</strong></td>
<td>138.89</td>
<td>42.50</td>
<td>63.02</td>
<td><strong>36.69</strong></td>
<td>49.44</td>
</tr>
<tr>
<td>Mdev</td>
<td>48.41</td>
<td>36.80</td>
<td>116.75</td>
<td>44.72</td>
<td>78.82</td>
<td>21.94</td>
<td>47.46</td>
</tr>
<tr>
<td>Yavg</td>
<td>5153.66</td>
<td>4435.50</td>
<td>20419.47</td>
<td>2605.75</td>
<td>5061.46</td>
<td>5635.86</td>
<td>3066.78</td>
</tr>
<tr>
<td>Snum</td>
<td>5.43</td>
<td>2.92</td>
<td>4.44</td>
<td>5.74</td>
<td>14.14</td>
<td>2.63</td>
<td>7.04</td>
</tr>
<tr>
<td>Mtot</td>
<td>302.05</td>
<td><strong>174.13</strong></td>
<td>589.78</td>
<td>242.30</td>
<td>884.86</td>
<td><strong>100.69</strong></td>
<td>346.18</td>
</tr>
<tr>
<td>Cgpa</td>
<td>3.06</td>
<td><strong>3.34</strong></td>
<td>3.34</td>
<td>3.62</td>
<td>3.14</td>
<td><strong>2.26</strong></td>
<td>2.61</td>
</tr>
<tr>
<td>Gprj</td>
<td>90.01%</td>
<td><strong>93.83%</strong></td>
<td>99.56%</td>
<td>100.89%</td>
<td>102.29%</td>
<td><strong>57.19%</strong></td>
<td>88.86%</td>
</tr>
</tbody>
</table>
Pruned M5P Model Tree

- Cgpa <= 3.205:
  - Jstr <= 66:
  - | Mavg <= 36.167: **LM1** (5/29.636%)
  - | Mavg > 36.167: **LM2** (18/106.448%)
  - Jstr > 66: **LM3** (40/57.247%)
- Cgpa > 3.205: **LM4** (48/31.383%)

**LM num 1:** Gprj =
- 0.0012 * Jstr + 0.0009 * Mavg
- + 0.0214 * Snum - 0.0002 * Mtot
- + 0.2908 * Cgpa - 0.2251

**LM num 2:** Gprj =
- 0.0001 * Jstr + 0.0009 * Mavg
- + 0.0214 * Snum - 0.0002 * Mtot
- + 0.2278 * Cgpa + 0.0515

**LM num 3:** Gprj =
- 0.0001 * Jstr + 0.0007 * Mavg
- + 0.028 * Snum - 0.0001 * Mtot
- + 0.226 * Cgpa + 0.0893

**LM num 4:** Gprj =
- 0.0008 * Mavg + 0.0143 * Snum
- - 0.0001 * Mtot + 0.1052 * Cgpa + 0.5199
Mean project grade as a function of cumulative start-before-deadline
Mean project grade as a function of cumulative work session time
Other patterns from spring 2013 Java Programming dataset

• Time of day of work sessions loosely correlated with project grade.
• Students with low grades tended to work exclusively after 8 PM.
• Students with higher grades tended to work periodically throughout the time of day.
• Office hour interaction with the professor & class attendance not measured, likely contributors.
Semesters after spring 2013

• Fall 2013 Operating Systems data discarded because projects consisted primarily of analysis that did not use make or other tools.
  – Most student time not in collected data set.
• Spring 2014 two additional sections of Java Programming, and one of Programming Lang.
  – Linear Regression has become slightly more predictive than more complex M5P model tree.
  – Subtracting unused days from start date improves accuracy.
  – Contributing attributes remain consistent.
Active versus Passive Procrastination (new for us)


• Good time management among active procrastinators.
  – Manage time by prioritizing work, not ASAP!
  – Increase flow (engagement) by increasing scheduling pressure!
  – We may be able to sort active vs. passive from our data.
Future work – Automated Nag or Interactive Game? (Try both.)

Time until deadline

Comparative grade monitoring as Snakes & Ladders
Use M5P Model Tree or Linear Regression to set slopes
Conclusions

• Procrastination is a factor in success, but:
• Some procrastinators can succeed.
  – High incoming GPA correlates, but it is no guarantee. Some high GPAs do not procrastinate well.
  – Are the successes self-identifying active procrastinators?
• Some non-procrastinators do poorly.
  – Length of work sessions and time of day are two strong contributing factors.
  – More analysis is upcoming.
• Results point direction towards automated assistant for at-risk students.