Enrollment Projection - Requirements Definition

Steven J Zeil:

Last modified: Jan 2, 2021

Contents:

1 Overview

Enrollments in the CS Department are in a continual state of flux. Enrollments are trending upwards as a combined result of economic trends outside the University, improved recruitment of new students, marketing of distance education options, and the creation of new degree programs outside of CS (such as the bachelors degree in cybersecurity and the masters in data science) that require significant numbers of CS courses.

Trying to project the number of students requiring each CS course in an upcoming semester has proven challenging. The Department frequently winds up adding new course offerings throughout the registration period, up to and sometimes even after the start of the new semester.

This document proposes a system to analyze historical and current enrollment data and to project the total number of students needing seats in each CS course.

2 Background

The University catalog defines the set of *courses* that the Department offers. For example, “CS350 Introduction to Software Engineering” is a course.

In the months leading up to a new semester, the Department plans its schedule, laying out an set of courses that it expects to teach. If expected enrollments are high enough, the Department may plan for multiple *offerings* of a course, where an “offering” represents a single course load for one faculty member. For example, in Fall 2020, the CS Department had three offerings of CS361, one taught by J. Sun, one by F. Wang, and one by J. Morris.

For a variety of administrative reasons, an offering may be split input multiple *sections*.

- Some courses are cross-listed between departments, meaning that students can register under a course number in either department. For example, CS620 is also available as DASC620.
- Courses at the 400/500 and at the 700/800 level can have their offerings be split to allow mixed classes of undergraduates and masters students (400/500) or of masters and Ph.D. students (700/800). For example, in Fall 2020, J. Wu taught a single offering of “Web Programming” available in sections as CS418 and CS518.

- Distance learning offerings are divided by "campus", a division that can reflect both physical locations and also a classification by the tuition rate charged. For example, in Fall 2020, J. Brunelle taught a CS411W offering that had eight different sections: four for students attending "live" at the Gornto building on the main Norfolk campus, a room at the ODU Virginia Beach Center, a room at the ODU Peninsula Center, or a room at the ODU Tri-Cities Center, and four sections for students attending by network conferencing, one for students in the Hampton Roads area, one for students elsewhere in the state of Virginia, one for students outside Virginia in the United States, and one for students outside the United States.

- From Spring 2020 through Summer 2021, as a concession to the ongoing pandemic, most "live" course offerings were split into two sections, one for students attending physically in the classroom and one for students attending via network conferencing.

Each section is uniquely identified (for a given semester) by a 5-digit CRN.

Students in different sections of an offering are taught by the same faculty member, and there is a shared overall cap to the total number of students that can enroll in the offering, regardless of which section they might select. Individual sections also have an enrollment cap limiting the number of students in that section alone.

For several years, has tracked its upcoming and current schedules to generate this schedule page. Historical data going back several years is available, and will form the basis for this enrollment projection project.

3 Requirements

3.1 Input

3.1.1 Command Line Parameters

The system will be a non-interactive program accepting the following as command-line parameters:

1. A list of one or more *semester directory* locations, indicating the locations of historical enrollment data.
2. A single directory location, indicating the enrollment data for the semester for which a projection is desired.
3. A path indicating the file location where the detailed projection report will be written.

The "semester directory locations" in items 1 and 2 may be given as either paths to local data or as URLs from which the data may be downloaded.

3.1.2 Semester Directories

Each of these directories will be named with a 6-digit number according to the following scheme:

1. The first four digits identify the *academic year* – the year in which the Fall semester begins.
2. The final two digits will be "10" for the Fall semester, "20" for the Spring semester, and "30" for the Summer semester. Thus, for example, 202010 denotes the Fall 2020 semester, and 202020 denotes the Spring 2021 semester. (2020 being the year in which the Fall semester of that academic year occurred.)

Each of these semester directories will contain

1. A file named `dates.txt` containing two dates, one per line, in YYYY-MM-DD format. The two dates will indicate the first day of pre-registration for the semester and the date of the "add deadline" (the last day on which students may add themselves to a course).

   If the `dates.txt` file is missing, the program should abort by throwing an `IOException` with the explanation "Missing dates.txt".

2. A series of *enrollment snapshots*, described below.

3.1.3 Enrollment Snapshots

These files will be named `YYYY-MM-DD.csv` where the `YYYY-MM-DD` is the date on which that snapshot was taken.

- Snapshots taken prior to the start of pre-registration should be ignored.
- Of the snapshots taken after the add deadline, all but the first should be ignored.

If, as a result of these rules, fewer than two snapshots remain to be processed, the program execution is aborted by throwing an `IOException` with the explanation “Insufficient snapshots”.

Each enrollment snapshot will be in CSV format. The first line of the file contains the column headers. Each subsequent row/line of the file describes a single course section.

Not every column in the snapshot will be relevant to this program. The important ones are:

CRN
: unique identifier for each section

SUBJ
: Identifies the department offering the course (e.g., “CS” for Computer Science)

CRSE
: Course number. The *course name* is the concatenation of the SUBJ and CRSE fields.

XLST CAP
: (cross-list cap) The maximum number of students that can enroll in this section.

ENR
: Number of students enrolled in this section

LINK
: Used to associate labs and recitations to a lecture. All link codes consist of an upper-case alphabetic letter followed by a single digit number. The number is 1 for a lecture section, 2 for a recitation, and 3 for a lab.

XLST GROUP
: (cross-list group) Identifies which sections of a course belong to a single offering. If this is empty, the section is the only one in its offering. If non-empty, all sections with the same SUBJ, CRSE, and XLST GROUP comprise a single offering.

OVERALL CAP
: The maximum number of students that can enroll in the offering to which this section belongs.

OVERALL ENR
: The total number of students enrolled in the offering to which this section belongs.

3.2 Projected Enrollments

The ultimate goal of the program is to project enrollments expected for each course on the future add deadline date of current semester. For this purpose, we will focus only on lecture sections. Sections of labs or recitations will be ignored.

Consider mapping the range of dates comprising the enrollment period as a number in the range 0.0 … 1.0, so that 0 represents the start of pre-registration, 1.0 represents the add deadline, 0.5 represents the date midway between those, and so on.

Let $c(d)$ and $h(d)$ denote current and historical enrollments, respectively, on date d. Let d' denote the last day of the current semester on which a snapshot was taken. Then we might expect that

$$\frac{c(1)}{h(1)} = \frac{c(d')}{h(d')}$$

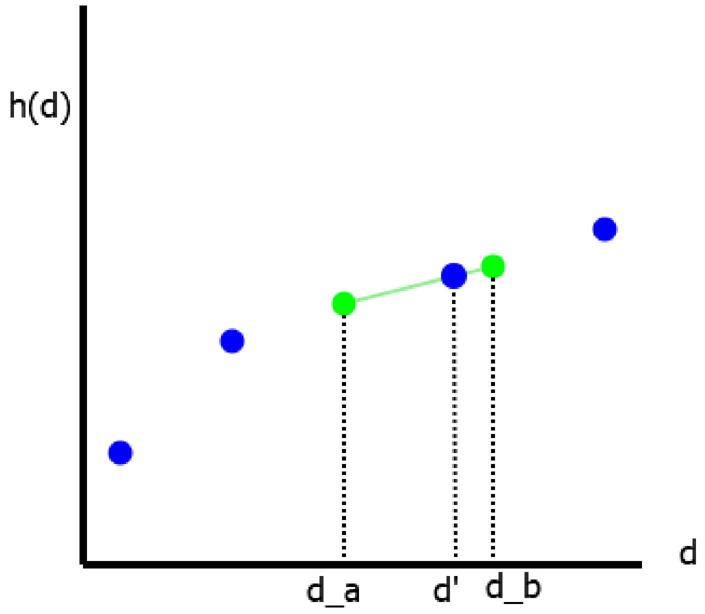
We don’t know $c(1)$, the enrollment on the add deadline of the current semester. But we do know the remaining three quantities in that equation.

3.2.1 Interpolation

Suppose that we have taken a snapshot for the current semester on date d. We can map d, as described earlier, onto a $0 \ldots 1$ scale, but it is extremely unlikely that value of d will map back onto an exact snapshot date in the historical semester. It will almost certainly fall between two dates, d_a and d_b, of existing snapshots.

We can estimate $h(d)$ by interpolation:

$$h(d) = h(d_a) + \frac{d - d_a}{d_b - d_a}(h(d_b) - h(d_a))$$



3.2.2 Curve Smoothing

Although the above formulas would suffice for making projections, it may yield results overly sensitive to single day evens (e.g., freshman orientations/previews, in which an entire block of freshmen register in a single day.)

A more stable prediction could be obtained by looking at the “shape” of enrollment curves over multiple semesters. If multiple historical semesters have been specified in the command line, then we can rewrite the projection equation as

$$\frac{c(1)}{h(1)} = r$$

where r is the average value of $\frac{c(d')}{h(d')}$ over all the historical semesters.

This curve smoothing is strongly recommended, but not required, for this project.

3.3 Output

There are two outputs from the program, the summary projection report and the detailed projection report.

3.3.1 Summary Projection Report

The summary projection report is written to standard output. It consists of two header lines followed by a line for each course offered during the current semester (item #2 in the input).

The first header line announces the percentage of the enrollment period (the time between the start of pre-registration and the add deadline) that has passed as of the date of the last enrollment snapshot. Negative percentages should be printed as zero, and percentages greater than 100% should be printed as 100%.

The second header line gives the headers for the following lines.

Following the headers, the courses are listed in order by course name. Each line prints the following:

1. A single character marker. This is an asterisk if the projected enrollment is greater than the total cap.

2. The course name
3. The current enrollment in that course (as of the last snapshot), summed over all offerings.
4. The projected enrollment for that course on the add deadline.
5. The total cap for that course, the sum over all offerings of the overall caps.

The second header line and the following lines should be arranged into left-justified columns.

A possible output would be:

```
 76% of enrollment period has elapsed.
  Course Enrollment Projected Cap
  CS120G 46         104       120
  CS121G 32         86        100
 *CS170G 55         88        75
     ⋮
```

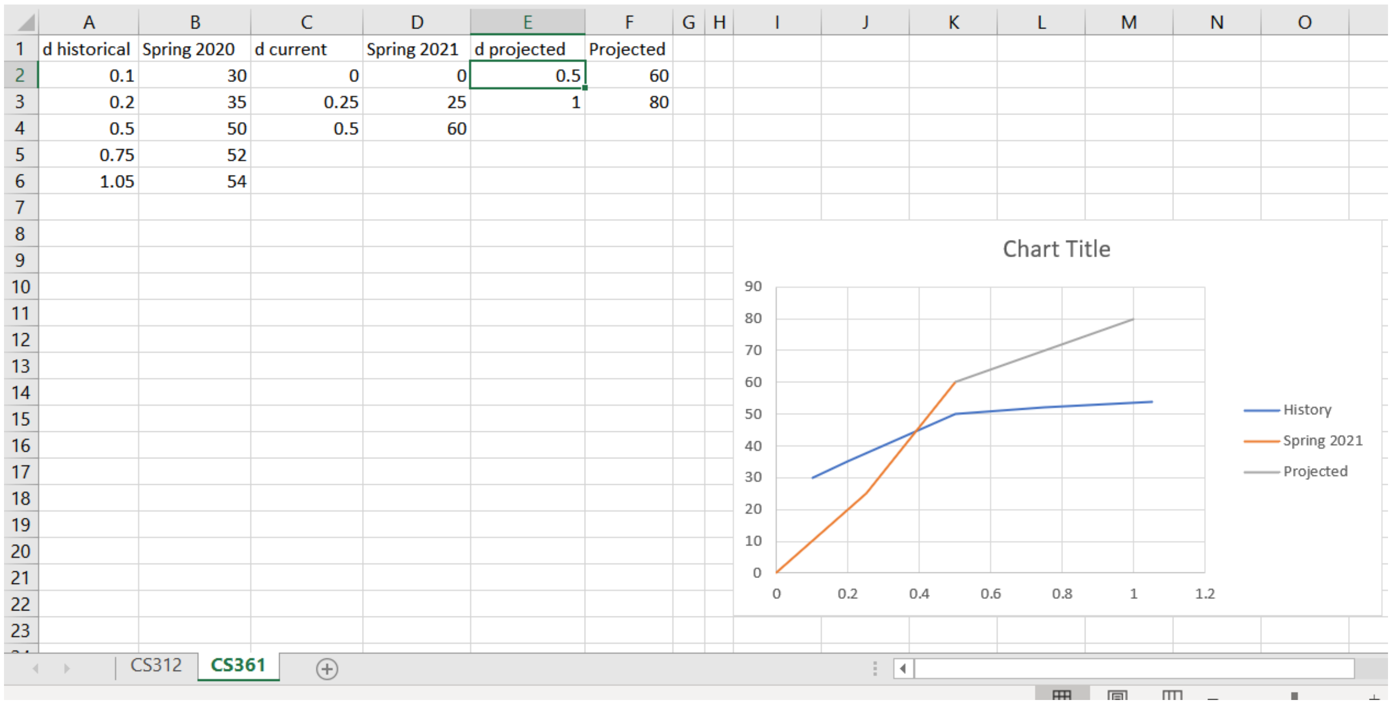
3.4 Detailed Projection Report

The detailed projection report will be an Excel workbook (spreadsheet) written to the path indicated in the command line parameters.

The workbook will contain one sheet per course, each sheet named with a course name and arranged in ascending order of course name.

Each sheet will show a graph (x/y scatter plot) of the historical enrollment, current enrollment, and projected future enrollment. It may contain any other data required to generate that plot.

An example of such a sheet is shown here:



	A	B	C	D	E	F
1	d historical	Spring 2020	d current	Spring 2021	d projected	Projected
2	0.1	30	0	0	0.5	60
3	0.2	35	0.25	25	1	80
4	0.5	50	0.5	60		
5	0.75	52				
6	1.05	54				

The horizontal axis maps the fraction of time elapsed between the start of pre-registration (0.0) and the add deadline (1.0) for the semesters involved.

The History line plots the historical enrollment. If curve smoothing is not implemented, this should be the actual enrollments recorded in the snapshots of the historical semester. If curve smoothing is implemented, this line should show the average enrollments for $d = [0, 0.05, 0.10, \ldots, 0.95, 1.00]$.

The current semester line plots the enrollments observed in the current semester snapshots.

The Projected line represents the linear projection from $(d', c(d'))$ to (1.0, c(1.0))$.

3.5 Additional Requirements

The program will be implemented in Java 1.11.

It will be delivered as a `EnrollmentProjection.jar` file that can be executed via a "`java -jar`" invocation. This Jar will include:

- Compiled code for the enrollment projector.
- Any necessary data files (e.g., a spreadsheet template)
- Compiled code from any third-party libraries required for execution of the system.

It shall not, however, include compiled versions of unit or integration test drivers and the libraries for their support.

The system should run on Windows, Linux, and MacOS systems equipped with an appropriate Java JRE.
