In-Class Design Exercise: Scantron Cheating Detection

CSE3461 W08 Prepared by: M. Baljko

Background

Final exams in certain courses consist of multiple-choice questions. Students indicate their answers on Scantron bubble sheets. Thus, each student's answers can be represented as a line:

012345678 2333d32442444444124125534252552d41d35114d41412532dd443dd22234343113415443144124422142511 2.....

The line consists of a student number, tab, and a vector of answers. The numbers 1, ..., 5 map to answers A, ..., E respectively. The symbol "d" indicates that the corresponding question was dropped from the marking scheme. The symbol "?" indicates that scantron could not read the answer on the student's bubble sheet.

Correspondingly, the answer key can be represented the same way:
answerkey 2231D22442443444124335551515322D31D33354D31451113DD532DD554142342221234322422231221425....

The symbol "D" indicates the question was dropped from the marking scheme.
The instructor provides the answer key to CNS, who then runs scantron on all of the exams and then sends back mark for each student, as well as a file containing the scanned data.

If cheating is witnesses, corroborating evidence can be obtained from the scantron data. If a student copies the answers from someone sitting nearby, the two answer vectors may appear suspiciously similar.

One approach is to calculate the *similarity measure* (number of same answers divided by the total number of questions). This measure can be suggestive. Suppose two students provide the same answer, and the answer is not only incorrect, but also incorrect in a way that differs from most of the other students who had an incorrect answer. But overall, the similarity measure can never be conclusive. Why? Because the similarity of two answer vectors is also affected by how many answers are correct. For instance, two students who both get a mark of 90% will have a high likelihood of having highly similar answer vectors.

The *kappa statistic*, on the other hand, tells us what the likelihood is that the similarity between two answer vectors is due to chance. For instance, suppose we suspect student X. We calculate the kappa statistic between X and all other students in the class. Suppose we find for student Y a similarity measure of 0.82, and the kappa statistic tells us that the likelihood that X and Y's similarity measure of 0.82 is due to random chance if 0.001%. This is strong evidence for cheating and can be used to corroborate eye witness accounts. Note that the measure cannot be used to tell whether X cheated from Y or vice versa. The kappa statistic should only ever be used to corroborate eye witness evidence.

The class `KappaAppDataModel` provides services for the calculation of the kappa statistic. It provides the following methods:

```
void setAnswerKeyFile(java.io.File file)
void setStudentDataFile(java.io.File file)
void setSuspect(java.lang.String suspectID)
void generateSimilarityReports()
```

The method writes the similarity reports to text files and places them in the same directory as the student data files. Here is a sample report:

ID1	ID2	Sim Score	|C|	Observed Agr	Expected Agr	Kappa	Standardized Kappa	Critical Val (0.0010)	Critical Val (0.0050)	Critical Val (0.01)	Critical Val (0.025)	Critical Val (0.05)	Critical Val (0.1)	Likelihood (%)
xxxxxxxx	20739xxxx	0.82	78	0.82	0.4006	0.6997	4.81524	3.09	2.576	2.326	1.96	1.645	1.282	0.10%
xxxxxxxx	20703xxxx	0.72	78	0.72	0.4274	0.511	2.25839	3.09	2.576	2.326	1.96	1.645	1.282	2.50%
xxxxxxxx	20635xxxx	0.59	78	0.59	0.3694	0.34983	1.58982	3.09	2.576	2.326	1.96	1.645	1.282	10.00%
xxxxxxxx	20779xxxx	0.55	78	0.55	0.361	0.29577	1.38465	3.09	2.576	2.326	1.96	1.645	1.282	10.00%
xxxxxxxx	20789xxxx	0.45	78	0.45	0.3073	0.20601	1.17189	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20812xxxx	0.42	78	0.42	0.294	0.17847	1.03979	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20655xxxx	0.43	78	0.43	0.3082	0.17606	0.94877	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20854xxxx	0.44	78	0.44	0.331	0.16293	0.79449	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20661xxxx	0.38	78	0.38	0.2856	0.13214	0.75754	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20868xxxx	0.43	78	0.43	0.331	0.14798	0.69482	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20789xxxx	0.4	78	0.4	0.3108	0.12943	0.68811	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20787xxxx	0.36	78	0.36	0.2812	0.10963	0.67077	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20850xxxx	0.41	78	0.41	0.3187	0.13401	0.6679	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20736xxxx	0.5	78	0.5	0.3894	0.18113	0.66785	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20851xxxx	0.4	78	0.4	0.312	0.12791	0.66727	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20837xxxx	0.31	78	0.31	0.2457	0.08524	0.63351	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20862xxxx	0.54	78	0.54	0.425	0.2	0.6323	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20839xxxx	0.45	78	0.45	0.354	0.14861	0.62546	3.09	2.576	2.326	1.96	1.645	1.282	>10%
xxxxxxxx	20840xxxx	0.45	78	0.45	0.3554	0.14676	0.62371	3.09	2.576	2.326	1.96	1.645	1.282	>10%

There is one other aspect of the kappa statistic. The statistic requires that, for each question, the probability distribution be derived over all of the possible answers. The default is that all of the student answers be used to build the probability distribution. But this can be changed.

Suppose the suspect student had x answers correct. A more refined statistic would be to consider only those other students who had a similar number of correct answers, such as $x \pm 2$. The value "2" is referred to as the *conditioning window size*.
It can be a bit tricky to choose a good value. Choose a window size too small, and there will be too few student answer vectors to provide a good probability distribution. Choose a window size too large, and the level of similarity required to satisfy the statistical threshold gets very high.

The class `KappaAppDataModel` keeps track of a list of different window sizes and generates a similarity report for each one.

```
void      addWindowSize(int i)

void      addWindowSizeMax()
```

Desired Features

1. Provide a facility so that the user (an instructor or other course administrator) can derive the kappa statistic reports for any student in a given class.
2. The user should be able to select additional conditioning windows, as desired.

It is important that the app be as fast and easy to use as possible.