Automatic graph classification

Brian Lukoff[*]

1 Motivation

In educational measurement, the objective is to measure some unknown latent variable or variables that represent a student's underlying ability or abilities in a subject domain. The first step in this process is to collect observable data from the student that will provide a window into these latent variables. Typically, this takes the form of a student's response to a test question (usually called a test "item"). Multiple-choice has proven to be the most popular item format for large-scale assessments because of the ease of automatically scoring student responses. However, multiple-choice items provide only a limited view into a student's knowledge and understanding, and the proliferation of test prep companies shows that multiple-choice items can be easily gamed.

Because of these weaknesses, an ongoing avenue of research in educational measurement focuses on richer environments for assessing student knowledge, such as questions that ask students to highlight a phrase in text, drag items into the proper place in the periodic table, and create concept maps (Zenisky & Sireci, 2002). But many of these methods are simply glorified multiple-choice questions—they do not provide a truly open environment for students to produce responses. When it comes to scoring open-ended responses, the Educational Testing Service has made progress with a trio of projects; the e-rater and c-rater projects score essays and natural-language responses (Attali & Burstein, 2006; Leacock & Chodorow, 2003), and the m-rater project scores equations and graphs (Livingston, 2009).

It is this latter problem—the problem of scoring a student's graphical response to a mathematics question—that we will examine in this paper. A weakness of the m-rater system is that it asks a student to generate a graph using a proprietary graphing tool, and then it tests the correctness of the graph by converting the graph to an equation or testing points (Livingston, 2009). An arguably more natural question to ask a student is to *sketch* the solution to a problem; such sketches are faster for the student to produce, doesn't require them to use an unfamiliar tool, and allows for a broad range of items to be asked. Perhaps more importantly, it allows for items with multiple correct responses.

Traditionally, free-response graphical items on large-scale assessments were scored by hand using human raters. This of course limits the amount and degree to which they can be used, and also introduces interrater unreliability as a source of error in student scores. An automated system would allow free-response items to be introduced and used more widely, and might potentially help improve the reliability of item scoring.

2 Problem Statement

In this paper, we will attempt to build a system that is able to score the following items:

Problem 1 (multiple correct responses) *Sketch the graph of a quadratic function.*

Problem 2 (single correct response) *A car is on a track that stretches from $y = 0$ meters to $y = 500$ meters, and at $y = 500$ meters there is a brick wall. At time 0, it starts at $y = 0$, drives at a constant speed, and after 5 seconds it hits the brick wall. For the next 5 seconds the car doesn't move after crashing into the brick wall. Sketch a graph of position (y) vs time that illustrates the position of the car over time.*

Note that Question 1 has many correct responses; any sketch that "roughly" corresponds to a graph of a function of the form $f(x) = ax^2 + bx + c$ ($a > 0$) should be considered correct. In contrast, Question 2 has a single correct answer; the only correct responses are sketches that "roughly" correspond to a graph of the function

$$g(x) = \begin{cases} 100x, & \text{if } 0 \le x \le 5 \\ 500, & \text{if } 5 \le x \le 10. \end{cases} \tag{1}$$

In both items, it is the definition of "roughly" that leads to interrater unreliability when these items

[*]E-mail: brian.lukoff@stanford.edu

are scored by humans: a student drawing with the mouse on a computer screen will never sketch a graph that exactly follows a simple function. Human raters must decode whether it is a shaky hand or a shaky understanding of the underlying mathematics concept that is driving a response that is not unequivocally correct.

Thus, the machine learning problem is the following: given an image of a student's response to one of these two items, make a binary (right/wrong) classification of the correctness of the response that lines up with human raters (as much as possible). We will evaluate the quality of the hypothesis h generated by each algorithm by using the error rate either on a separate test set or estimated using leave-one-out cross-validation.

3 Data sets and Features

3.1 Problem 1

For Problem 1, I generated simulated data consisting of $m/3$ graphs each of simulated sketches of linear, quadratic, and cubic functions, leading to training and test sets that were each of size m. Each "simulated sketch" of a polynomial of degree d ($d \in \{1, 2, 3\}$) consisted of a plot of the function $s(x) = \sum_{i=1}^{d} a_i x^i + \varepsilon$ in the window $[-M, M] \times [-M, M]$ where $a_1, \ldots, a_d \sim \text{Uniform}([-M, -1] \cup [1, M])$ and the noise variable $\varepsilon \sim N(0, \tau^2)$.[1] For the quadratic or cubic sketches, sketches where $|s(M)| < M$ or $|s(-M)| < M$ were discarded and the parameters reselected, because such a function s cannot be completely graphed in the graphing window. For these experiments, I set $\tau = 1/4$ and $M = 5$; the choice of M is arbitrary (but there is no reason to believe that any other value would make the simulation any more or less realistic) and the choice of τ is a subjective choice that gives what seems to be a realistic "shakiness" in each curve. I generated $m = 3000$ simulated responses for both the training and test sets.

To derive the features of the classifier, I first scaled the 300×300 pixel images down to a 15×15 pixel image (where the (i, j) pixel in the smaller image consists of the sum of the grayscale values in the box $[1 + in/d, (i + 1)n/d] \times [1 + jn/d, (j + 1)n/d]$. Each of the grayscale values of the $15^2 = 225$ pixels in the smaller image was used as a feature.

[1]The somewhat odd choice of distribution for the coefficients $\{a_i\}$ is because I wanted to avoid degenerate quadratic or cubic functions where the term of the quadratic or cubic term was near zero.

3.2 Problem 2

For Problem 2, I collected real samples of human-generated responses to the problem. Using Mechanical Turk, $m = 54$ subjects participated by reading the problem statement and then sketching a solution with their mouse using a system based on an existing JavaScript-based painting solution (Vock, 2006). A subject's solution was saved in PNG format and then imported into Matlab for analysis. Subjects were paid a token amount for their participation, but were informed that they were paid for their effort and not for the correctness of their solutions.

I manually scored each participant's response on a binary (right/wrong) scale, and trained an SVM classifier using these labels.

In addition to using the 15×15 pixels of the shrunken image as features, an additional feature extracted from the data for this problem was the number of "soft matches" between pixels on the shrunken (15×15 pixel version) response image and a canonical correct solution generated by mechanically plotting the function given in equation (1) and then reducing that image to 15×15 pixels.[2] In other words, this feature is the number of the 225 pixels where the (scaled-down) student response are both nonwhite (i.e., grayscale value 0).

A final feature for this problem was the number of soft matches between the left and right halves of the response image and each of two "partial solutions" that consist of the one of the two components of the correct solution image (the diagonal line representing the car traveling at a constant speed and the horizontal line representing the car after it has hit the wall).

4 Methodology

Because of previous success using Support Vector Machines to solve the problem of handwriting recognition (LeCun, Bottou, Bengio, & Haffner, 1998), and because of the conceptual similarity between handwriting recognition and the problem considered here, I elected to use SVM to build the classifiers. SVM^{light} (Joachims, 1999) was used to train and test the models.

I compared the following sets of classifiers:

[2]Note that this is not a feasible feature for Problem 1 because in that problem there was no single "correct" response.

Features	Kernel	Augmented?	P1 (quadratic function)	P2 (car position)
Baseline			67%	57%
Pixels only	Linear	no	96%	85%
Pixels only	Polynomial	no	>99%	85%
Pixels and correct solution	Linear	no		91%
Pixels and correct solution	Polynomial	no		93%
Pixels, correct solution, partial solutions	Linear	no		91%
Pixels, correct solution, partial solutions	Polynomial	no		91%
Pixels and correct solution	Linear	yes		89%
Pixels and correct solution	Polynomial	yes		93%

Figure 1: *Experimental results*

- A baseline classifier that consisted of predicting the most frequent class for every training case.
- A SVM with a linear kernel, setting $C = 1$.
- A SVM with a polynomial kernel, setting $C = 1$ and $d = 2$ (i.e., a quadratic kernel).

For P1, a separate test set of $m = 3000$ cases was used to estimate error. For P2, since the training set was so small to begin with, leave-one-out cross-validation was used to estimate error.

For P2, an augmented training set of 11 additional training cases—the correct solution and 11 almost-correct responses consisting of two almost-correct line segments[3]) was also used in some experiments. These additional training cases were *not* left-out in the leave-one-out cross-validation, to avoid inflating the estimated error.

5 Results

Figure 1 shows the results of the experiments. The current results are very encouraging. In Problem 1, SVM with a quadratic kernel classified all but one of the 3000 test cases correctly! Unsurprisingly, the results were not quite as good in Problem 2: there were fewer test cases ($m = 54$ for P2 versus $m = 3000$ for P1), and of course in P1 the test cases were generated through simulation and not actual student responses (as they were in P2).

Given that there were only 54 cases in the entire data set for P2, and the best SVM (pixels and correct solution used as features, polynomial kernel, no augmentation of training set) classified $50/54 = 93\%$ of the cases correctly, it is interesting to examine the 4 cases that the SVM classified incorrectly. Figure 2 shows the four responses that were classified incorrectly. There seem to be three problems with the classifier:

1. Cases 24 and 52 have the first part (the upward-sloping line segment) correct, but these participants neglected to add the second part of the graph (the horizontal line segment).
2. Case 42 was labelled as correct because the participant's intent—to draw a horizontal line segment from $t = 5$ to $t = 10$—is clear. To a human grader, it's clear that they simply didn't consistently hold the mouse button down the entire time as they were drawing the segment; however, the SVM seems to be relying on those missing pixels and mistakenly classifies this case as incorrect.
3. Case 54 has the correct graph—but also extends the horizontal line segment to the left, making the response clearly incorrect. In this case, the SVM may be marking it as correct based on the fact that the correct graph is essentially embedded inside the response as a subgraph.

It is also interesting to note that beyond adding the correct solution soft-match feature, neither additional features nor the augmentation of the training set helped improve the cross-validation error. Several points about this can be made:

1. The sample size for P2 was relatively small; any additional improvement beyond the current results would be at risk of involving features that are cherry-picking the specific attributes of the

[3]In other words, drawing two connected line segments, but with slopes slightly different than the correct slopes of 100 and 0.

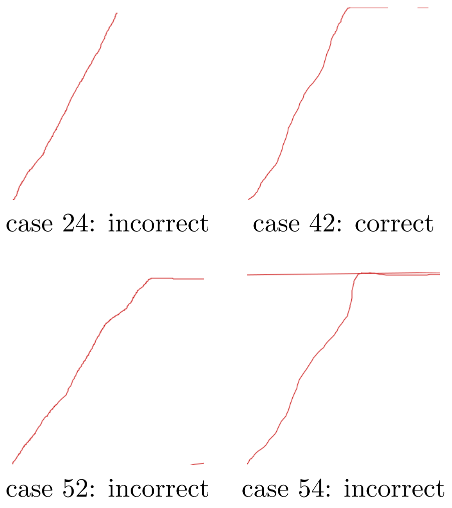

Figure 2: *P2 responses that were classified incorrectly by the best classifier*

training cases that failed to match, and thus might not be generalizable.

2. It is not necessarily the case that even two human raters would agree on 100% of the cases. For items like these, raters might disagree on how much "sloppiness" in the student's response is acceptable; for example, how close to $t = 5$ the pivot from a slope of 100 to a slope of 0 must be.

3. The highest cross-validation error estimate was obtained by using as features only the pixels of the shrunken image and a single additional feature (soft-matches of the response image with the correct solution). This is nice because it suggests that this technique is likely generalizable to other types of test items that require a graphical response (e.g., other types of position-vs-time car scenarios like the one used here) due to the fact that there was no "fine-tuning" of the features for the particular problem used.

6 Limitations and Future Work

The results here point to potential applications of this technique to using graphical response items in computer-based testing. However, there are a number of areas where future work is needed before such items could be used in an operational setting. First, more work is needed to see whether these results generalize to different kinds of items. In this study we presented two different items and found that an SVM performed well in either case, but the universe of possible graphical response items is of course much larger and so it is necessary to see whether the technique generalizes.

A future study with a larger data set might investigate whether there are other features that would further reduce generalization error. Although it is encouraging that in P2 one of the classifiers attained only 7% error in cross-validation, a larger training set might contain other patterns of incorrectness, not seen in the small data set used here, that might require additional features to correctly classify. Similarly, although in P1 the cross-validation error was quite low, the data set consisted only of computer-generated "sketches" of random linear, quadratic, and cubic functions; if P1 were presented in a real test-taking environment, some students would inevitably respond by sketching graphs of higher-order polynomials, non-polynomials, or even non-functions. The only way to test this would be to gather large data sets of actual student responses to train a (potentially more complex) model.

Finally, one general limitation of this technique is that it requires a training set of human-scored data; compare this to a typical multiple-choice question, which can be asked of students and scored automatically simply by supplying the scoring algorithm with the correct answer. In practice, though, for a variety of reasons[4] test items that appear on large standardized tests are always rigorously pretested on a relatively large group of students before making it onto a real test; the response images from these pretests could be manually scored by human raters and then used to train a SVM for scoring the item operationally.

References

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater v.2. *Journal of Technology, Learning, and Assessment*, *4*(3), 1–31.

Holland, P., & Wainer, H. (1993). *Differential item functioning.* Lawrence Erlbaum.

Joachims, T. (1999). Making Large-Scale SVM Learning Practical. Advances in Kernel Methods-Support Vector Learning. MIT Press.

[4]For example, looking for Differential Item Functioning (Holland & Wainer, 1993).

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*(4), 389–405.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998, November). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Livingston, S. A. (2009, September). Constructed-response test questions: Why we use them; how we score them. *ETS R&D Connections*, *11*.

Vock, R. (2006). *Ajax paint.* `http://smir.andaloria.de/ajaxpaint/`.

Zenisky, A., & Sireci, S. (2002). Technological innovations in large-scale assessment. *Applied Measurement in Education*, *15*(4), 337–362.