Knewton ML Challenge

Raiden Hasegawa

Method in a nutshell.

We use an Item Response Theory parameterization with a discrete representation of latent ability. This allows us to estimate our data model as a finite mixture of Bernoulli random variables and maximize the log-posterior using the Expectation Component Maximization (ECM) algorithm. We opt for a semi-Bayesian treatment and introduce priors to what would normally be a maximum likelihood problem due to issues with non-uniqueness of the likelihood maximization problem and numerical difficulties arising from large parameter estimates.

The Problem.

We are given a dataset consisting of ~14,000 astronometrics students with their performance on multiple choice test questions selected from a set of 400 in some fashion, presumably randomly. Tasked with teaching this year's astronometrics class of a similar size with only a few TAs, we decide to construct our final exam by selecting five questions randomly from the pre-approved question bank for each student with hopes of realizing a meaningful ranking of the students at the end of the class. The problem here is that we know that some of the questions are not very good at actually testing the astronometric knowledge of our students. We want to eliminate these questions from our question bank, but the school requires us to use at least half of the questions in our test generation algorithm so as to cover the full range of topics. If we knew which questions were good, we could easily eliminate these. But, unfortunately question quality is unobserved. Additionally, since we don't know what questions are good questions, the scores of last years students are at best a noisy measure of the students' ability. Knowing neither of these exactly makes this a challenging problem.

Hypothesis.

The first thing we must do is define what a good question is. Since we are looking for questions that can help us determine a meaningful ranking of student's knowledge of astronometrics at the end of the course, we can make the reasonable assumption that a good question is one that is good at distinguishing between students of different levels of understanding. How can we identify this quality of the questions in our question bank given our dataset?

A first guess.

One heuristic measure of a good question, as defined above, is the log-odds ratio of answering the question correctly between students with different levels of understanding. Consider students j and k and question i. The log-odds ratio of question i between the two students can be written mathematically as

$$logOdds^i_{j,k} = \log\left(\frac{\frac{P_j(Correct_i)}{P_j(Incorrect_i)}}{\frac{P_k(Correct_i)}{P_k(Incorrect_i)}}\right) \tag{1}$$

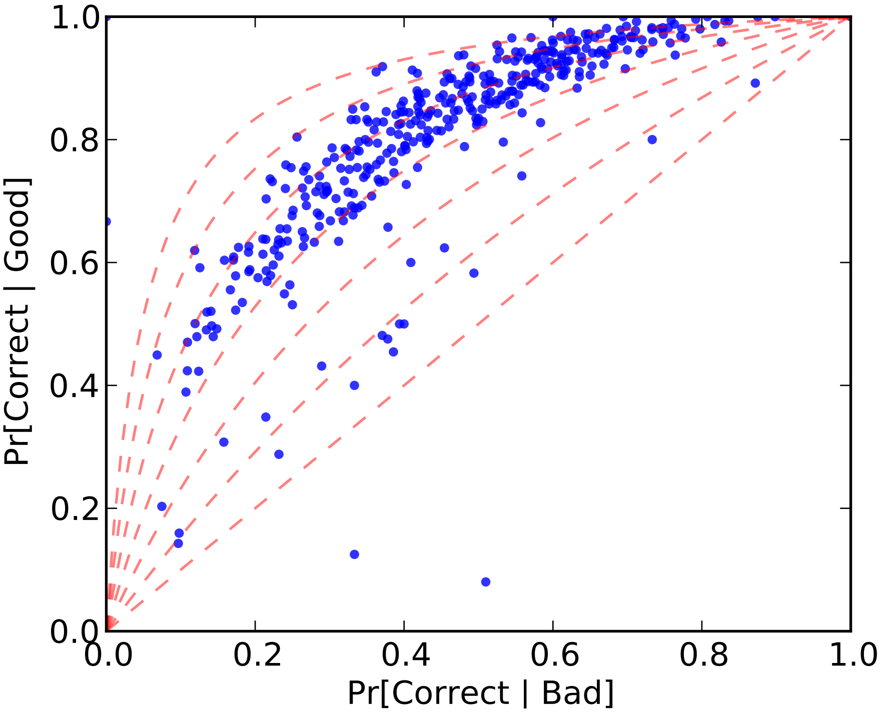
In particular, a good questions should have a high log-odds ratio between students with a high level of understanding (student j) and those with a low understanding (student k). This just means that when answering a good astronometrics question, it is more likely that students who know more astonometrics will answer the question correctly. If the log-odds ratio is near or less than zero, the question is probably not a good gauge of astronometric knowledge.

Empirical log-odds.

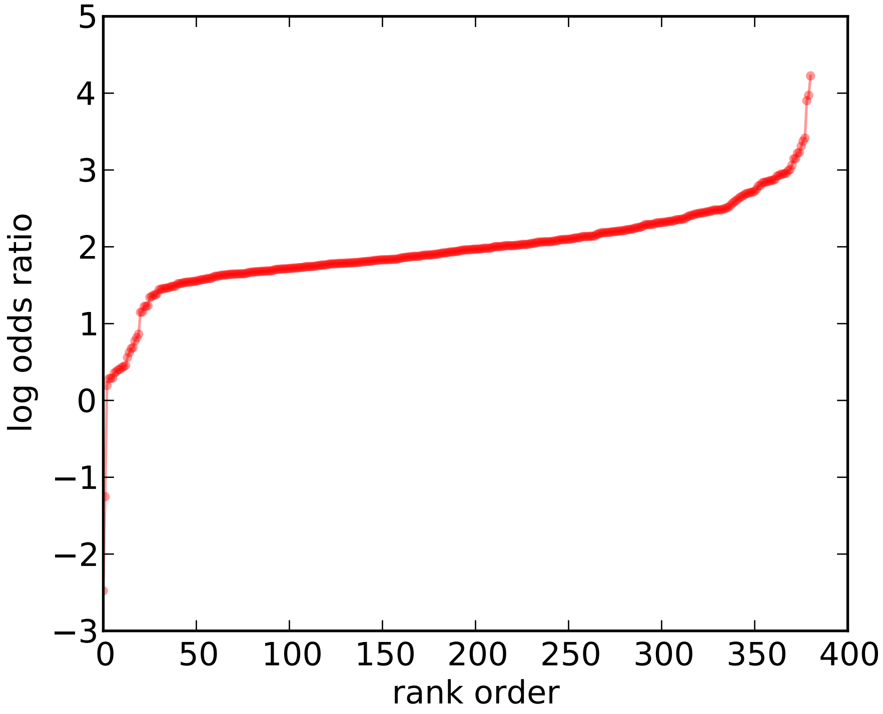
Before performing more sophisticated analyses, we can get a sense of what the log-odds ratio tells us about our dataset. Consider a world where there are only two types of students: good students and bad students. A simple rule to partition the students in our dataset into these two groups is to label the top 50% of the students in terms of empirical probability of answering a question correctly as good students and the bottom 50% as bad students. We can then look at each question and compare the probability of answering it correctly for both groups. The empirical probability that student n in our dataset answers question i correctly can be written as

$$\hat{P}_{i,k} = \frac{1}{|D_n|} \sum_{i \in D_n} \mathbb{I}_{n,i} \tag{2}$$

where $\mathbb{I}_{n,i}$ takes the value 1 if student n answers i correctly and 0 otherwise and D_n is the set of questions that we have data for on student n. Below is a plot of each question in our dataset where the x-axis is the empirical probability that a bad student (as defined by our rule) will get the question correct and the y-axis is the empirical probability that a good student will get the question correct. The red dotted lines are log-odds iso-contours starting at a log-odds ratio of 0 on the 45 degree line and going up in increments of $\frac{1}{2}$ toward the northwest corner.



The data exhibits a pretty nice pattern with most questions falling between the 1.5 and 2.5 log-odds ratio contours. The striking feature of this chart is the outlying questions that have log-odds ratios of less than 1.5 (there are even some below zero!). You can see this even more clearly by rank ordering the log-odds ratio for each question and noticing the dramatic drop off at the left tail (see below).



If I was the professor of this year's class and I had very little time to figure out which questions to eliminate I might use the log-odds ratio heuristic and drop a number of these outlying questions from our test generation algorithm. This method has a few very obvious drawbacks. First, we are pretty certain that the quality of a student can't be represented sufficiently by our good/bad binary model of ability. Second, even if the world did follow our model, since each student only answers a randomly sampled subset of our questions, our rule for figuring out which student belongs to which type is potentially very noisy. In the same vain, some questions are not attempted by many students at all so our log-odds ratio measure may also be very noisy. This may not be a great algorithm in practice but it does help us clarify what we are looking for: questions that can reliably discriminate between students of different ability.

Motivation.

The purely graphical analysis above can actually be motivated by some useful and well studied theory: Item Response Theory (IRT). Briefly (since I am sure you know more about this than me), the two parameter logistic IRT model parameterizes the probability that a student n with some latent (unobserved) ability θ_k answers question i, which can be characterized by its difficulty α_i and its discrimination δ_i, as

$$P(X_i = 1|\theta_k, \delta_i, \alpha_i) = p_{ki} = \frac{e^{\delta_i(\theta_k - \alpha_i)}}{1 + e^{\delta_i(\theta_k - \alpha_i)}} \tag{3}$$

where X_{ni} is a random variable that takes value 1 when student n gets question i correct and 0 otherwise. With this parameterization in mind, one can see that the log-odds ratio for each question i we looked at in the previous section can be written as

$$logOdds^i_{good,bad} = \delta_i(\theta_{good} - \theta_{bad}) \propto \delta_i\,. \tag{4}$$

The discrimination parameter is directly related to our idea of log-odds. Graphically, it corresponds to the slope of the logit function. With this model in tow, we might be able to do a better job than our graphical analysis above.

Method.

To make the analysis simpler, we assume that latent ability can be modelled as a discrete variable that we know ahead of time (this is an obvious weakness of the model). Although this assumption might not be optimal, it may be detailed enough to assign students to the right letter grade (we use 8 latent classes... I guess we should have used 11 or 12 to map to an A-F scale). We predetermine that θ_k can take values -2, -1.5, -1,-.5, .5, 1, 1.5 or 2. Additionally, we assume that items are independent in the sense that, conditional on latent ability θ_k, responses to any two items are independent. The goal is to estimate δ_i and α_i for each question. In particular, we want estimates of the δ_i's in order to rank the questions in our question bank based on how well they discern the ability of a student. To do this, we maximize a log-posterior distribution of the parameters given the data X_{ni} using the Expectation Component Maximization (ECM) algorithm.

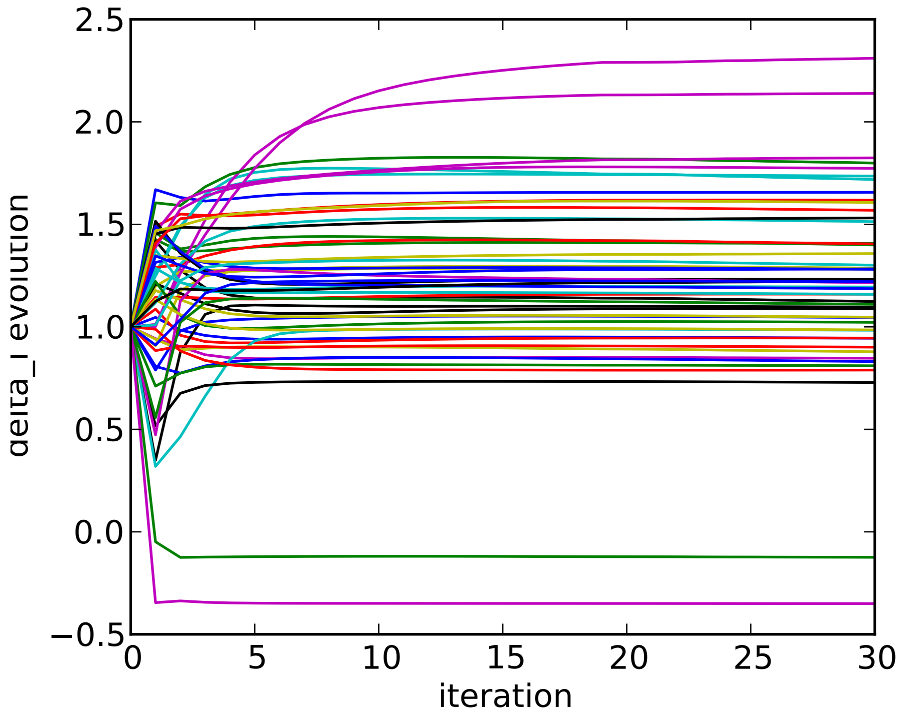
Originally, I wanted to put the model in a maximum likelihood framework but because of questions in our dataset with only a few student observations, there are some issues with large parameter estimates that caused numerical overflow (i.e. $e^{\delta_i(\theta_k-\alpha_i)}$ was too big). Another issue is that the maximum likelihood is not unique since $\delta_i(\theta_k - \alpha_i) = C$ for any constant C yields solution curves rather than a unique solution for δ_i and α_i. Thus, we introduce normal priors on δ_i and α_i. The mean and standard deviations of the priors were chosen somewhat arbitrarily (another weakness) but we have some reasoning for the choices. We choose a larger standard deviation ($s.d. = 2$) for the $\delta_i's$. Since we are most interested in their ranking, we want to allow them to vary enough to generate a useful ranking. We keep the prior on the α_i's smaller ($s.d. = 1$) so our algorithm (hopefully) converges to a unique solution. The means of the priors on δ_i and α_i are 1 and 0, respectively. These means reduce our 2 parameter model to a model that explains the observed data as a function only of student ability and thus seem like reasonable choices.

Because we are using a discrete representation of latent ability, we can express our log-likelihood as a mixture of Bernoulli random variables which lends itself to the use of the EM algorithm to maximize the log-posterior (see Bishop, 2006.). The latent variables in this model, $Z = \{z_1, z_2, \cdots, z_n\} \in \mathbb{R}^K$, are unit vectors of length K (the number of latent ability classes) with one element equal to one and all other elements equal to zero. The latent variables assign each student to a latent ability. For the Bernoulli mixtures model, the E-step of the EM algorithm calculates the expectation of the latent variables, $z_{nk} \forall n \leq N, k \leq K$, conditional on the data. The M-step then maximizes the expectation of the complete data (X, Z - observed and latent) log-posterior with respect to Z (see the appendix for mathematical details). Since our parameter space is so large, we split our maximization step into two components: with respect to α and with respect to δ. The parameter space is still large in both cases but because local independence is assumed our hessians for both components end up being diagonal which is computationally friendly (See Meng & Rubin, 1993. for why ECM works). The mathematical appendix gives the formulae for the log-posterior, expected log posterior, gradients, hessians, etc.

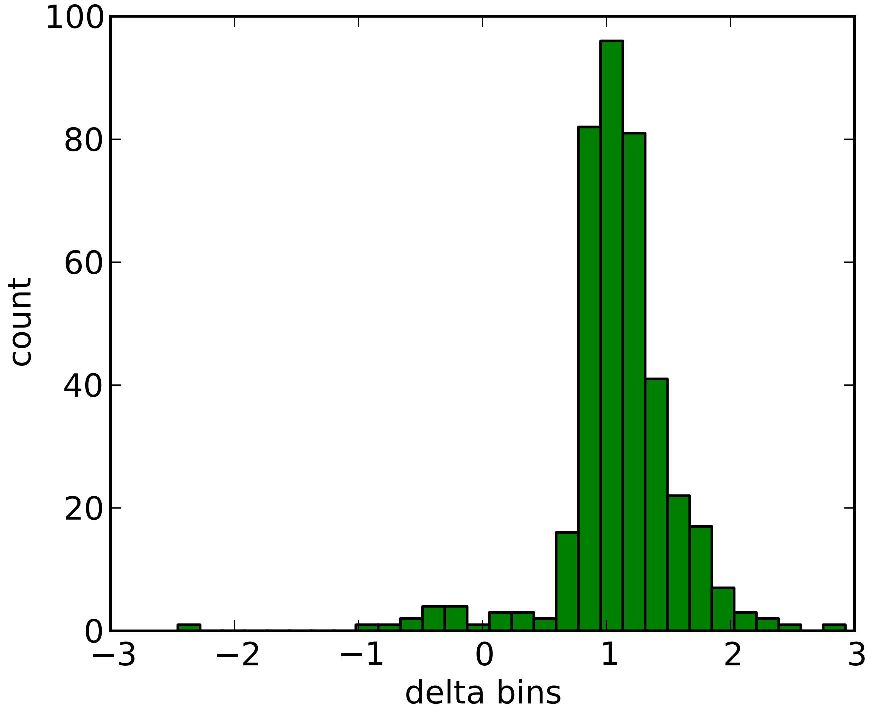
Once we estimate our model, we can identify the questions with the smallest δ_i's and eliminate them from the question bank. A reasonable cutoff for inclusion in the question bank may be $\delta_i > 1$. In addition to discovering and eliminating bad questions, this estimation will allow us to assign students in this upcoming year to latent ability classes (grades) based on the new parameter estimates post-exam (we can't determine exactly which grade a student deserves (their z_n) but can determine which latent ability most likely explains the result of their test).

Results.

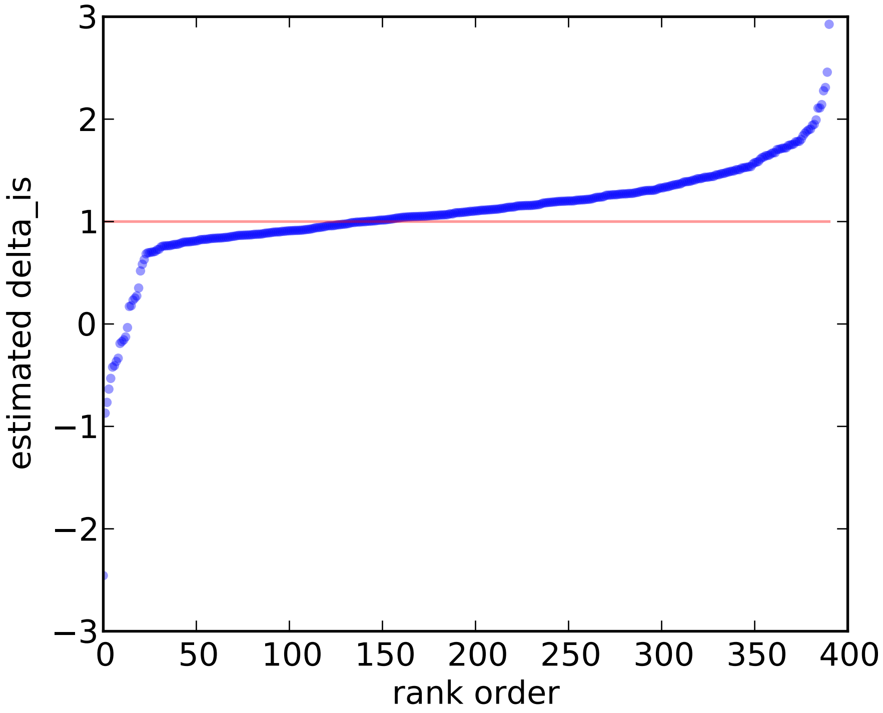
The list of questions removed by the algorithm can be found in 'results/emMartianRemove2013_03_18_15_06_32.txt'. Our ECM algorithm ran for 110 iterations until the max parameter change for any δ_i was less than 1e-5. But, upon viewing the evolution of the δ_i it is clear that convergence to a useful ordering happens much earlier on. Except for δ_i very close to our inclusion cut-off of 1, we know which questions we will eliminate well before the 110th iteration. You can see this in the plot below which show the parameter evolution of 50 random δ_i over the first 50 iterations.



A look at the histogram of estimated δ_i show that many of the $\delta_i < 1$ present themselves as outliers with respect to the rest of the distribution which suggests that our criterion for inclusion isn't unreasonable. The only thing that we need to check is that our cutoff criterion doesn't shrink our question bank to less than $\frac{1}{2}$ its original size (it doesn't).

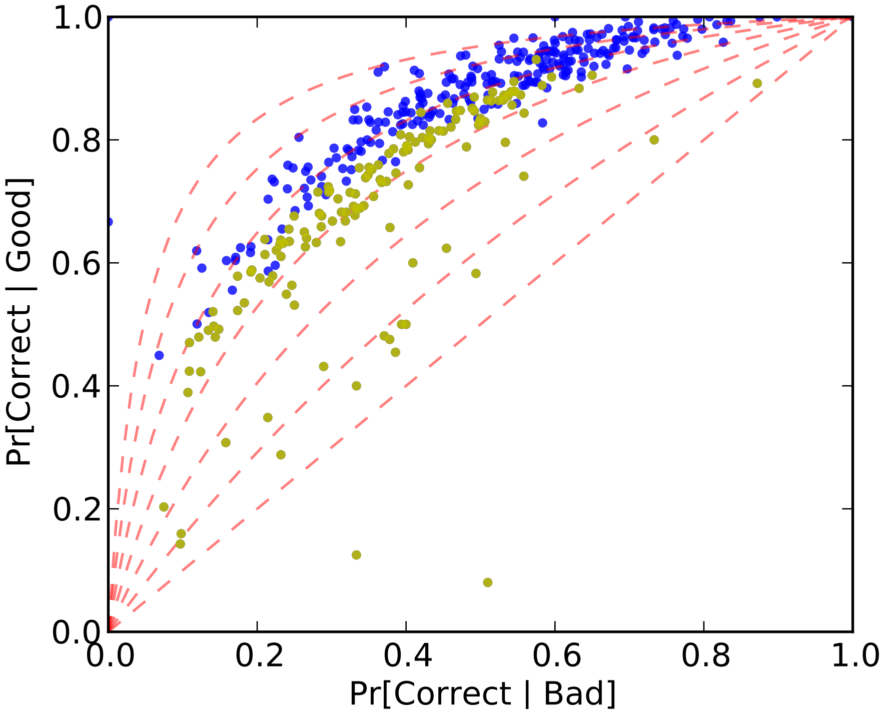


Plotting the rank-ordered δ_i gives rise to a figure that is similar to the rank-ordering we saw in our initial exploratory analysis where the questions tail off dramatically in there ability to discern the ability of a student at a certain point.



Notice in the rank-ordering though, that the majority of δ_i span a narrow range of values. We don't extract a posterior measure of variance for each parameter since we are essentially just mode-finding. Thus, we should be careful about the certainty with which me make statements about the quality of questions. Given the time constraints, this method probably yields a pretty good estimate, but a MCMC method may be a better option since you can sample the whole posterior distribution with this method and generate estimates for the variance of our estimated parameters. We could loosen our priors to see if we get our δ_i's to span a larger wider range, but this most likely will have the effect of increasing the posterior variance which leaves us dealing with the same issue. It might be wise to only eliminate the questions in the lower tail (see the δ_i rank order plot) from the question bank rather than using our $\delta_i < 1$ cutoff rule.

This issue aside, let's see if our ECM results are in line with our initial exploratory analysis. Below is the same log-odds contour plot we saw earlier with the questions that our ECM determined as having $\delta_i < 1$ marked with yellow dots. Our algorithm appears to eliminate the questions that reside in the lower log-odds contour areas which is what we had expected.



This is intuitively satisfying but makes one wonder if our heuristic approach would have been enough? My guess would be that the answer is no. There's a lot of subjectivity in this approach and using the IRT framework and estimating our model can provide useful theoretical motivation for our inclusion rule. Introducing a more explicit modeling and estimation framework can help us infer more information about our students and compare students abilities. You can also see from the plot that we have a finer understanding of how to differentiate questions: sone excluded questions are very close in proximity to included questions. Using the heuristic approach may yield a similar set of questions for our question bank but gives us a systematic way to assign students into ability classes (i.e. grades) by comparing the estimated likelihood a student answers the questions the way they did given they are in a particular latent ability class (see appendix for derivation of this term, $\gamma_{nk} = E[z_{nk}]$).

Other Remarks.

- Local independence could be a pretty unreasonable assumption since several questions may address different but related (or not even different) concepts in astronometrics. Scientific discovery and knowledge builds upon itself so it would be difficult to believe that one could construct a set of ~400 questions that are collectively exhaustive (in terms of material covered) but mutually exclusive.
- In hindsight, it may have made sense to eliminate questions from our dataset that had very few student attempts. This may have allowed us to avoid using restrictive priors, although the issue of non-uniqueness may still have been an issue.
- We could have also split the M-step into a component for each item given the assumption of local independence. The hessians would no longer be diagonal since δ_i and α_i are correlated, but each hessian would only be 2×2 and thus easily invertible (unless singular...). The upside is that we might get faster convergence since we can explicitly take into account this correlation. The downside is that the convergence speed up might be negated by increased computation (just a guess).
- With more time, it would be good to study the sensitivity of our results to changes in the the specification to θ_k (both to the number of ability classes and the θ value of each class) as well as to our choice of priors. An even more useful next step may be to estimate the θ_k values as well in the maximization step, but this introduces another level of indeterminacy (i.e. $(\theta_k + c) - \alpha_i = \theta_k - (\alpha_i - c)$)

which we would have to deal with with another prior, which although less subjective than choosing the θ_k values explicitly, is still a subjective decision.

References.

1. Bishop, C.M. (2006). *Pattern recognition and machine learning* (Vol 4., No. 4), 423-459. New York: springer.
2. Nocedal, J., & Wright, S. J. (1999). *Numerical optimization,* chapter 6. Springer verlag.
3. Bafumi, J., Gelman, A., Park, D. K., & Kaplan, N. (2005). Practical issues in implementing and understanding Bayesian ideal point estimation. *Political Analysis*, 13(2), 171-187.
4. Meng, X. L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2), 267-278.
5. Companioni, A. (2012). The Mismeasure of Students: Using Item Response Theory Instead of Traditional Grading to Assess Student Proficiency, Knewton's N Choose K blog.

Appendix.

All formulae are for items i, students n and latent abilities k. These derivations draw from Bishop, 2006. pretty heavily.

Priors

$$\alpha_i \sim \mathcal{N}(0,1) \; i.i.d., \forall i \tag{5}$$

$$\delta_i \sim \mathcal{N}(0,2) \; i.i.d., \forall i \tag{6}$$

Probability a student in ability class k answers question i correctly

This is the two parameter IRT model.

$$_{ki} = \frac{e^{\delta_i(\theta_k - \alpha_i)}}{1 + e^{\delta_i(\theta_k - \alpha_i)}} \tag{7}$$

Derivatives (useful for gradient and hessian derivation):

w.r.t. α_i:

$$\frac{\partial p_k i}{\delta \alpha_i} = -\delta_i p_{ki}(1 - p_{ki}) \tag{8}$$

w.r.t. δ_i

$$\frac{\partial p_k i}{\delta \delta_i} = (\theta_k - \alpha_i) p_{ki}(1 - p_{ki}) \tag{9}$$

Student level likelihood

The probability that student n receives the scores that we observed ($x_n = \{x_1, x_2, \cdots, x_n\}$) given that they are in ability class k.

$$P(x_n|\theta_k, \alpha, \delta) = \prod_{i \in D_n} p_{ki}^{x_{ni}} (1 - p_{ki})^{(1 - x_{ni})} \tag{10}$$

Posterior probability that ability class k is "responsible" for student n's results

This is the expectation of z_{nk}, which is equivalent to the probability (likelihood) that the student n gets the scores we observed ($x_n = \{x_1, x_2, , x_n\}$) given being in latent ability class k. This is evaluated

$$\gamma_{nk} = E[z_{nk}] = \frac{\pi_k P(x_n|\theta_k, \alpha, \delta)}{\sum_{k=1}^{K} \pi_k P(x_n|\theta_k, \alpha, \delta)} \tag{11}$$

Unconditional probability of being in a particular latent ability class

This result is derived by maximizing the Lagrangian of the log-posterior with the condition that these probabilities, π_k's, add up to one.

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_{nk}}{N} \tag{12}$$

Log-posterior

This is what we are maximizing using the ECM algorithm (the second and third terms are due to the priors on α and δ)

$$\mathcal{P}(\delta, \alpha|\theta, \pi, X) \propto \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k P(x_n|\theta_k, \alpha, \delta) \right\} - \sum_{i=1}^{D} \frac{\alpha_i^2}{2} - \sum_{i=1}^{D} \frac{(\delta_i - 1)^2}{8} \tag{13}$$

Expectation of complete-data log-posterior w.r.t Z

This is what we explicitly maximize in the CM-step. Complete data means both the students results, X_{ni}, $\forall n, i$ and the latent Z_n vectors.

$$E_Z[\mathcal{P}(\delta, \alpha|\theta, \pi, X)] \propto \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ \log \pi_k + \sum_{i \in D_n} x_{ni} \log p_{ki} + (1 - x_{ni}) \log (1 - p_{ki}) \right\} - \sum_{i=1}^{D} \frac{\alpha_i^2}{2} - \sum_{i=1}^{D} \frac{(\delta_i - 1)^2}{8} \tag{14}$$

Gradients

Of the expectation the complete-data log-posterior w.r.t. Z.

w.r.t. α_i:

$$\frac{\partial E_z}{\partial \alpha_i} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ -\delta_i (x_{ni} - p_{ki}) \right\} - \alpha_i \tag{15}$$

w.r.t. δ_i:

$$\frac{\partial E_z}{\partial \delta_i} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ (\theta_k - \alpha_i)(x_{ni} - p_{ki}) \right\} - \frac{(\delta_i - 1)}{4} \tag{16}$$

Hessians

Of the expectation of the complete-data log-posterior w.r.t. Z (Diagonal elements).
w.r.t. α_i:

$$\frac{\partial^2 E_z}{\partial \alpha_i^2} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ -\delta_i^2 p_{ki}(1-p_{ki}) \mathbb{I}_{i\in D_n} \right\} - 1 \tag{17}$$

where$\mathbb{I}_{i\in D_n}$ takes value 1 if student n attempted question i and 0 otherwise.
w.r.t. δ_i:

$$\frac{\partial^2 E_z}{\partial \alpha_i^2} = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ -(\theta_k - \alpha_i)^2 p_{ki}(1-p_{ki}) \mathbb{I}_{i\in D_n} \right\} - \frac{1}{4} \tag{18}$$

where$\mathbb{I}_{i\in D_n}$ takes value 1 if student n attempted question i and 0 otherwise.

What was used.

Versions

- python 2.7.3
- matplotlib 1.2.0
- numpy 1.6.2
- scipy 0.11.0

What's here.

Code (in main directory)

- martians_final.py (main code that runs ECM algorithm)
- emfunctions_final.py (contains functions for ECM: log-posterior, expected log-posterior, gradients, hessians, etc.)
- plotting.py (generates plots for pre-estimation data exploration and post-estimation analysis - You need to manually change the ECM output file names to produce the corresponding post-estimation analysis and plots)

Data (in data directory)

- aStudentData.csv

Plots (in figures directory)

- logOddsContour.pdf
- logOddsContour_post.pdf
- logOdds.pdf
- delta_rankOrder.pdf
- delta_hist.pdf
- delta_evol.pdf

Output (in results directory)

- emMartianRemove<date>.txt: questions with delta < 1 to be removed [MAIN RESULT]
- emMartianDelta<date>.txt: delta from each iteration of ECM algorithm
- emMartianAlpha<date>.txt: alpha from each iteration of ECM algorithm
- emMartianPi<date>.txt: pi from each iteration of ECM algorithm
- emMartianOutput<date>.txt: negative log posterior from each iteration of ECM algoritm