**Slide 2**

I propose a model for response times as a synthesis of ideas developed in the literature. The model simultaneously estimates a probability for each item being compromised and a probability for each examinee having item preknowledge.

The DG-IRT model uses the Rasch model to model response accuracy data, but it adds a twist. It divides the items in the test into two sets: compromised items and uncompromised items. Then, for each examinee, the model estimates a true latent ability based on compromised items and latent cheating ability based on uncompromised items. Finally, it computes the posterior probability of the cheating latent ability being higher than the true latent ability and uses it as a proxy for item preknowledge.

**Slide 3**

The LNRT model is a latent trait model for response times. And it models response times as a function of a latent speed parameter for an examinee (**stop a second**) and two item parameters, time-intensity, and time-discrimination (**stop a second**). The higher the latent speed for an examinee, the lower the expected response time for an item, so examinees with higher latent speed are expected to respond faster to an average item. If an item has a higher time intensity, it takes more time to respond for an average examinee. And, time-discrimination is a parameter for unaccounted variability in response times not due to a common latent speed construct. A lower time-discrimination parameter implies more noise in response times for an item beyond the systematic variance due to the latent speed construct.

- The new model borrows the ideas from the DG-IRT model and applies them to response times through LNRT with additional considerations. The new model doesn’t assume that the sets of compromised and uncompromised items are known, and instead, it estimates the probability that a particular item is compromised. The probability of an examinee having item preknowledge is directly estimated as opposed to DG-IRT, in which the estimate is somewhat indirect. It also uses a neat trick during the estimation by marginalizing the likelihood over the two discrete parameters.

**Slide 4**

The proposed model is designed to detect item preknowledge with the assumption that examinees that have seen a group of items before would tend to respond faster to these items. The model has three parameters for each person and three parameters for each item.

H\_sub\_i (**stop a second**) is an indicator variable with two values and represents the examinee item preknowledge status for an examinee. If H\_sub\_i is 1, it implies that examinee i has item preknowledge, and if H\_sub\_i is 0, it implies that examinee i does not have item preknowledge.

Like DG-IRT, the model hypothesizes two latent speed parameters for each examinee. Tau\_t\_sub\_i (**stop a second**), a true latent speed parameter for examinee i when the examinee responds to uncompromised items, and tau\_c\_sub\_i (**stop a second**), a true latent speed parameter for examinee i when the examinee responds to compromised items.

When it comes to item parameters, C\_sub\_j (**stop a second**) is also another indicator parameter with two values for item i. If C\_sub\_j is 1, it implies that item j is compromised, and If C\_sub\_j is 0, it implies that item j is not compromised. Beta\_sub\_j and alpha\_sub\_j (**stop a second**) are the time-intensity and time-discrimination parameters for item j and have the same interpretation as the LNRT model.

The model assumes that the log of the response times follows a normal distribution. The variance of the distribution (**stop a second**) is defined as a function of the time-discrimination parameter, alpha\_sub\_j.

All other parameters, H\_sub\_i, C\_sub\_j, tau\_t\_sub\_i, tau\_c\_sub\_i, and beta\_sub\_j, play a key role in defining the mean of this distribution. Like DG-IRT, there is a gating mechanism to define the mean. When an examinee with item preknowledge responds to a compromised item; in other words, when both C\_sub\_j and H\_sub\_i are equal to 1 (**stop a second**), then the mean of the distribution is equal to beta\_j – tau\_c\_sub\_i, so the model assumes an examinee with item preknowledge utilizes their cheating speed when responding to compromised items. In all other conditions, the mean of the distribution is equal to beta\_j – tau\_t\_sub\_i. In other words, when an examinee with item preknowledge responds to an uncompromised item or an examinee without item preknowledge responds to either a compromised or an uncompromised item, the model assumes that the examinees utilize their true latent speed.

**Slide 5**

I want to talk about breaking down the density of distribution for the observed log response time. As I was working on this model, this became necessary because I was using Stan to fit the model. I was told that Stan could not handle the discrete parameters such as H\_sub\_i and C\_sub\_j in this model (**stop a second**). Therefore, it is necessary to rewrite the overall density by marginalizing it over discrete parameters.

Since we have two discrete parameters in the model, and each can only take two values, either 0 or 1, we can consider four possible combinations.

In the first possible combination, both H\_sub\_i and C\_sub\_j are equal to 1 (**stop a second**), representing a situation in which an examinee with item preknowledge responds to a compromised item. In this case, the density reduces to **this quantity (stop a second)** multiplied by the probability that H\_sub\_i is equal to 1 and the probability that C\_sub\_j is equal to 1. Notice that the reduced version of density quantity only depends on tau\_c\_sub\_i because, in this condition, we expect the examinee to use the cheating latent speed.

In the second combination, H\_sub\_i is equal to 1 while C\_sub\_j is equal to 0 (**stop a second**), a situation in which an examinee with item preknowledge responds to an uncompromised item. In this case, the density reduces to **this quantity (stop a second)** multiplied by the probability that H\_sub\_i is equal to 1 and the probability that C\_sub\_j is equal to 0. Notice that the reduced version of density quantity now only depends on tau\_t\_sub\_i because, in this condition, we expect the examinee to use the true latent speed. Although the examinee has item preknowledge on some items, because this is an uncompromised item, the model assumes that the examinee still have to use the true latent speed.

**Slide 6**

The other two combinations are very similar, and both covers the conditions in which an examinee without item preknowledge responds to either a compromised or uncompromised item. The density reduces to a form that depends on true latent speed and multiplication by the respective probabilities (**stop five-to-ten seconds**).

**Slide 7**

In the end, we can write the overall density as a sum of these four terms that cover all possible combinations of discrete parameters. This allows us to eliminate the discrete parameters in the model and instead directly sample the relevant quantity the probability these parameters are equal to 1.

Here, I introduced a very simplified version of the model statement in Stan that can be used to represent the overall density in a few lines of code. Note that this is a very simplified version assuming there is no missing data. Since I have missing data, I had to tweak this a little bit to handle missing data. Another thing to note is that we are working on the log scale while fitting the model, so you see the log of these quantities in the syntax. For instance, normal\_lpdf (**stop a second**) represents the log of normal density. Or log(pC[j]) (**stop a second**) represents the log of probability that C\_sub\_j is equal to 1. And, log1m(pC[j]), (stop a second) represents one minus the log of probability that C\_sub\_j is equal to 1, which is technically equal to the log of probability that C\_sub\_j is equal to 0.

You also see that we use either p\_sub\_t or p\_sub\_c as the mean of the distribution depending on the condition. In three conditions, we use p\_sub\_t, which is a function of tau\_t\_sub\_i, and in one condition, we use p\_sub\_c, which is a function of tau\_c\_sub\_i.

The most important feature of this representation and syntax is that we can directly sample C\_sub\_j and H\_sub\_i, which can be interpreted as the probability of item j being compromised and the probability of examinee i having item preknowledge.

**Slide 8**

To test whether this model can be successfully fitted to a real dataset and yield reasonable meaningful parameters, I used a random sample of 1000 examinees from Form A in this dataset. Form A has 171 items in total, and 50 of these 171 items were operational items that all 1000 examinees responded to, and 121 of these 171 items were pilot items that only 100-150 examinees responded to. Each examinee was given a subset of 15 pilot items in addition to the 50 operational items.

Due to the missing data by design for the pilot items, the dataset was relatively sparse, and it literally looked like this in wide format. About 38% of the data matrix was complete, and the rest were missing due to not administered pilot items.

To deal with missingness in Stan, I transformed the data from wide format to long format and analyzed 65,000 individual responses (1000 examinees x 65 items). That’s why I had to tweak the simplified Stan code I showed in earlier slides to deal with long format data instead of wide-format data.

**Slide 9**

I fit the model using Stan. There were four sampling chains with 1000 iterations each. The first 250 iterations were used as a warm-up, and the remaining 750 iterations were used to calculate the posterior means for parameter estimates. This may sound somewhat low number of iterations for people using other Bayesian software such as WinBugs or JAGS. However, because Stan is allegedly using a more efficient sampling method, it typically requires fewer iterations to get quality parameter estimates.

I don’t have time to get into the details of estimation. However, I put an informal blog post about all the details of the model fitting process. I also put all the code used for this presentation in a Github repo. If you are interested, please send me an email, and I can send you all these links.

**Slide 10**

The model convergence was checked by visual inspection of the sampling chains and the split-chain R-hat statistic. You can see below the distribution of R-hat statistics for all model parameters separated by type of parameters. The split-chain statistic was less than 1.01 for most parameters in the model, with a maximum value of 1.032 for one of the item parameters, the time-intensity parameter for Item 30. Overall, there was no indication of non-convergence or red flags.

**Slide 11**

I also inspected how well the chains were mixing by looking at the trace plots. Below, I present a trace plot of a randomly selected parameter from each type. Again, there was no indication of any pathological behavior during sampling. So, the model was successfully fitted without any issues.

**Slide 12**

I want to focus on two parameters in this analysis. As I mentioned, a unique feature of the proposed model is that it estimates a probability of being compromised for each item and a probability of having item preknowledge for each examinee. In the below plot, I provide the distribution of estimates for the probability of being compromised for 50 operational items (at the top) and 121 pilot items at the bottom. The model indicated a clear separation between pilot and operational items, yielding much higher estimates for 50 operational items. The highest probability estimate for pilot items (considered secure by design) was 0.91. Therefore, if I use 0.91 as a cut-off value to obtain beyond chance and apply this cut-off to operational items, the model would indicate that 47 out of 50 operational items were potentially compromised.

**Slide 13**

I also double-checked my findings by simulating data with no item preknowledge. This was to ensure that the separation I see between operational and pilot items in probability estimates of being compromised is not due to any data structure or characteristics such as large amounts of missingness or something else. I simulated response time data for 1000 examinees and 171 items using the same parameters (true latent speeds, time-intensity, and time-discrimination parameters) estimated from the model. There was no item preknowledge. I also replaced 62% of values in the data with NAs for Item positions from 51 to 171. So, I mimicked the exact same data structure. Then, I fitted the model with the same specifications. The plot below shows the distribution of the same parameter estimates for operational and pilot items. As you can see, there is no separation as we saw in the real data. So, the model was most likely picking up some signal in the real dataset.

The distributions are centered around 0.5 because I use a Beta (1,1) prior. It is a uniform distribution between 0 and 1. So, it is expected that the center of the estimates when there is no item preknowledge is aligned with the mean of the prior distribution. You may be wondering why the distribution is more peaked for pilot items and a little more spread for operational items. I think it is related to sample size. Because pilot items have fewer observations, around 100-150, than pilot items with 1000 observations, their estimates are more strongly drawn to the center of the prior distributions.

**Slide 14**

When it comes to the other parameter of interest, the probability that an examinee has item preknowledge, there was also something fishy in the distribution.

The top plot below provides the distribution of the probability estimates of having item preknowledge for 1000 examinees from the IT certification data. Notice the uptick on the right side of this distribution, and it is a bit unusual.

Similarly, I looked at the corresponding distribution from the same simulated data with the null condition. I also used a beta(1,1) prior for this parameter. So, when there is no item preknowledge, it is centered around 0.5 with a maximum value of 0.64.

So, values on the higher end of the distribution from the IT certification data are certainly unusual. If we use, for instance, 0.9 as a cut-off point to flag examinees, the model would flag 96 out of 1000 examinees as potentially having item preknowledge.

For instance, I could use a lower threshold, such as 0.65, based on the distribution from null data, but I want to be overly cautious, so I think 0.9 is a reasonable choice.

**Slide 15**

I also looked a little deeper into this subgroup of 96 examinees identified by the model based on a threshold of 0.9.

One striking difference was the difference in response time between this subgroup of examinees and the rest of the sample for a model-identified subset of items and other items. As seen in the table at the top, there is only about a 6-second difference in response time between 47 flagged items by the model and the remaining 124 items for 904 examinees not flagged by the model. On the other hand, the average response time is reduced from 86 seconds to 33.5 seconds, a 61% drop for the 96 examinees flagged by the model.

So, a certain subgroup of examinees unusually responded faster to a specific subset of items. The model picked this signal in the data and separated the examinees into two classes and items into two classes. This is not surprising since the model is purposefully designed to pick such a signal in the data.

I also observed unusual similarities in this subgroup. Based on the information available in the dataset, this subgroup of examinees was different from the rest of the group in certain aspects. For instance, 64.5% of the model-identified examinees were from Country X. In contrast, only 18.8% of the remaining sample were from this particular country. So, the model-identified subgroup of examinees mainly was from the same country. All examinees in this model-identified subgroup were marked as online-proctored, whereas only 61.5% of the sample was online-proctored. 78.1% of the examinees in the model-identified subgroup was flagged for voucher misuse, while the rate of voucher misuse was only 11.2% for the remaining examinees. Finally, 93.8% of examinees in the model-identified subgroup were also flagged by the company due to independent investigation based on the response similarity analysis.

All this information reinforced my opinion that the proposed model was responsive to some systematic variance in the dataset.

**Slide 16**

I want to conclude with a few remarks:

I am confident to argue that a certain subgroup of examinees responded significantly faster to a specific subset of items, almost all operational items, than the rest of the group.

The proposed model is designed to pick such a signal and was successfully fitted to a random sample from the dataset.

The model successfully separated a particular group of examinees in the data from the rest of the group by estimating the probability of item preknowledge for each examinee.

The model also successfully separated a subset of items by estimating the probability of being compromised for each item.

In the context of this presentation, I tend to interpret faster response times as an indication of item preknowledge. If there are other plausible explanations for faster response times in operational items for this subgroup of examinees, this inference is void.

The idea can be extended and used for response accuracy data (work in progress!)

Response time and response accuracy pieces can be combined. It becomes a very complex model but can potentially yield the highest performance (work in progress!)

**Slide 17**

Thank you for listening. Please reach out if you have any questions.