Multi-alignment in the RNA Proficiency Model

Introduction and Summary

Motivation
We urgently need to model two new scenarios: (1) LO's have several sub-LOs, and (2) some items may be multi-aligned to different sub-LOs. This document lays the general theory of multi-alignment (MA) in the RNA proficiency model, with an eye towards solving both of these modeling challenges. Happily, there are two tractable options of MA in RNA, representing different modeling assumptions and behaviors, each of which is useful for our two problems.

The LO / sub-LO System
Forget about concept-concept MA for now. We are soon going to have LOs with several sub-LOs (i.e. concepts). How are we going to infer proficiencies on both LOs (for goal completion) and concepts (for targeted recommendations) at the same time? Naively computing on just concepts seems bad. Consider an LO with 4 concepts. If a student gets one question correct on each of these sub-LOs, we probably want them to have destroyed the target LO. However, just averaging the masteries across these concepts will not lead to a high enough value. Attempts to define lower thresholds are hacky and ill-fated.

Disjunctive IRT (DIRT) provides an elegant solution. Items are aligned to both the concept, and the LO, and thus proficiency is inferred on both entities simultaneously. The general DIRT proficiency update equations given below provide the starting point for analyzing this system. In DIRT, we have to be careful about what things can in-principle be observed. It turns out that there are equivalence classes of predictive models, parametrized by scalar invariants, that flexibly enable us to "divvy up" the proficiency between the LO and concept in way that both (a) preserves exactly the behavior of observable quantities we have today for LOs with a single concept, and (b) yields promising results for jointly learning proficiencies on both LOs and concepts. There is a single new parameter that controls how the proficiency is divided between LOs and concepts.

Computing proficiencies is not enough. We need a way of computing mastery values. How do we extend the current definition of mastery on an LO to the case multiple sub-LOs? First, we

need to understand how to compute a score/RP variance in MA RNA. It turns out that score (co)-variances can be computed simply, in closed form, using just the RNA method. This is detailed in the appendix. Mastery is then defined as the score-at-percentile on a set of default items, one for each LO - concept pair in the system, and which are aligned to both the LO and concept. This definition matches the usual definition in the limit of one concept, but generalizes in a principled way.

Concept-concept multi-alignment

While DIRT proves very effective in the LO / sub-LO system, it is not adequate for modeling genuine concept-concept MA. Luckily, Conjunctive IRT (CIRT) seems ideal for this case. Proficiency is allocated in a nuanced way that accounts for what you knew and didn't know going in. Notably, on incorrect answers the decrement to proficiency accounts in a principled way for the uncertainty in which concept you failed on.

Putting it together

We propose using DIRT for the LO-concept system and CIRT for concept-concept MA items. Thus, in general we'll have a hybrid DIRT-CIRT MA. While at first this seems overly complex, these two types of MA operate nearly independently. Moreover, the math factorizes and remains tractable in closed form.

Towards Recommendations

The Core Assessment model in all flavors of MA RNA can be derived in closed form without the heuristics and bad approximations plaguing the current PROD implementation. An outstanding challenge is to specify the desired behavior of RX with respect to:

- Prioritizing which sub-LO to work on
- Choosing between MA items
- How to treat multi-item bundles

Core assessment provides a principled starting point for looking at these questions.

Disjunctive IRT (DIRT)

The first type of multi-alignment has a rich theory landscape, and potentially many uses, ranging from regular concept-concept multialignment (though we'll see DIRT is probably not a good option in this case), to modeling forgetting, jointly learning item difficulties and proficiencies, and jointly learning proficiencies across different levels of granularity, such sub-LOs and LOs.

We'll start by briefly defining the model mathematically, then discussing the possible uses in more detail. Then we'll dive into the main mathematical results.

Definition

Consider an item i aligned to concepts A, B, and C. It might be the case that the student doesn't strictly need mastery in all three concepts in order to answer i correctly. For example, a

Commented [1]: Oh good point. In the past, the only covariance I thought about was N items all assessing the same concept — leads to an important correlation in the score. But it gets much more complicated with multi-alignment and sub-LOs! If we don't get covariances right, things could go wrong in unexpected ways.

Commented [2]: Yeah see the discussion in the appendix. That old "too confident to quick" problem is more-or-less solved by the intuition from DIRT covariances.

Commented [3]: Worth including a short reference as to why here?

Commented [4]: I plan to add this to the main doc at some point. But in brief:
1) Explaining away (see the sub-section on that), wherein profs on singly aligned items can go down a bunch after corrects on other concepts
2) The "unintelligent" allocation of prof increments/decrements for correct/incorrect. Imagine an item aligned to A and B, with prof states (mean, variance) of A = (1.0, 0.5) and B =(-1.0, 0.5). If you get it wrong, both concepts would lose the same amount of prof, which is weird.

Commented [5]: Is there brief intuition to offer on how the two operate together — like, is it hierarchical, is it conjunctive on the outside, disjunctive on the inside?

Commented [6]: An item MA to subLOs c1 and c2 has likelihood
Phi(alpha_1*(w_1^T * theta_vec - beta_1)) *
Phi(alpha_1*(w_2^T * theta_vec - beta_2))
So DIRT inside each CIRT factor.

Commented [7]: And in (LO, c1, c2) space w_1 = (c, s, 0) and w_2 = (c, 0, s) where c = cos(phi), s=sin(phi) which represent the single DIRT rotation parameter dialing between LOs and concepts

deficiency in A can be compensated for by mastery in B and C. We'll call this scenario *compensatory*, or *disjunctive*, multi-alignment. For an item aligned to N concepts we write the likelihood of a correct response (r = 1) as:

$$P(r=1|\vec{\theta}) = \Phi\left(\sum_{c=1}^{N} w_c\theta_c\right) = \Phi\left(\vec{w}\cdot\vec{\theta}\right)$$

$$\text{where } \vec{\theta} := (\theta_1, \ldots, \theta_N)$$

Note that we choose a compact representation of DIRT where item parameters are absorbed into the weights, without loss of generality. To obtain the more standard representation we can simply add another variable representing the item difficulty (with zero variance, unless we want to learn difficulties jointly with proficiencies), with negative weight, and rescale the weights, like so:

$$\theta_{N+1} = \beta$$
$$w_{N+1} = -1$$
$$w_n \to \alpha w_n \;\; \forall n$$

Here and throughout this doc we denote item discriminations as α, item difficulties as β, weights as w, and the proficiency random-variables as θ. It should be clear from the above result that a deficiency in one concept can be made up for by another concept, assuming positive weights.

Uses of DIRT

DIRT multi-alignment may be useful in at least four different settings:

1. **Concept-concept multi-alignment**
   Even though this is the most straightforward application, we'll see below that DIRT is likely not a good fit here, and a different model (CIRT) seems to give behavior more in line with expectations.
2. **The joint estimation of proficiency across multiple levels of granularity**
   For example, we may be interested in learning proficiencies on both learning objectives (LOs), as well as constituent sub-LOs (i.e. concepts). More generally, we might want to estimate mastery on concepts, LOs, topics, chapters, or even entire domains.
3. **Temporal multi-alignment**
   Here an item interaction is influenced by several different proficiency parameters operating on different time scales, i.e. with different forgetting and learning rates. This may provide a more robust model of the temporal evolution of proficiencies that is both consistent with cognitive models, and provides a mechanism for prioritizing review.
4. **The joint learning of item difficulties with proficiencies**
   In the RNA framework both proficiencies and item difficulties can be estimated simultaneously.

While the interpretation of the math varies for each of these cases, the core math remains the same, and the main results are given below.

Note: Throughout this doc we sometimes use the term “concept” loosely here, since depending on the use case, we may associate a θ with an LO, topic, item difficulty, or more abstract entities such as the component of concept active at a characteristic time scale.

DIRT Proficiency Update Equations

Given a prior $\boldsymbol{\rho}$ on the proficiencies, the posterior becomes[1]

$$\rho(\vec{\theta}|r) = \frac{1}{P_r}\Phi\left((-1)^{r-1}\vec{w}\cdot\vec{\theta}\right)\rho(\vec{\theta})$$

We’ll be interested in three different moments of the posterior:

- The normalization P_r (i.e. The 0th moment of the unnormalized posterior), which is just the probability of getting correct (r = 1) or incorrect (r = 0).
- The posterior proficiency means (the first moments)
- The posterior proficiency variances (the second moments)

One important point is that we must solve the general case where $\rho(\boldsymbol{\theta})$ is a multivariate Normal with non-zero off-diagonal covariances. The reason is that even if we started with a independent Normal priors, the update equations induce off-diagonals terms. Therefore we assume the “prior” proficiency is:

$$\rho(\vec{\theta}) = \phi_{\vec{\mu},\boldsymbol{\Sigma}}(\vec{\theta})$$

The results below cover cases where we start with independent Normal priors, or graph-structured priors with propagation.

The mean RP (with derivation relegated to the appendix) is given by

$$\boxed{\begin{aligned} P_1 &= \Phi(\gamma) \\ \text{where } \gamma &= \frac{\mathbf{w}^T\boldsymbol{\mu}}{Q} \\ Q &= \sqrt{1+\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}} \end{aligned}}$$

Meanwhile, the posterior proficiency mean vector and variances are

[1] Recall that $\boldsymbol{\Phi}(-z) = 1 - \boldsymbol{\Phi}(z)$

$$
\begin{aligned}
\mu_c^{(r)} &:= \langle \theta_c | r \rangle = \mu_c + \frac{(\mathbf{\Sigma w})_c}{Q} R_r(\gamma) \\
\Sigma_{ab}^{(r)} &:= \langle (\theta_a - \mu_a^{(r)})(\theta_b - \mu_b^{(r)}) | r \rangle \\
&= \Sigma_{ab} - \frac{(\mathbf{\Sigma w})_a (\mathbf{\Sigma w})_b}{Q^2} R_r(\gamma) \left[\gamma + R_r(\gamma) \right] \\
\text{where } R_r(\gamma) &:= (r - \Phi(\gamma)) \frac{\phi(\gamma)}{\Phi(\gamma)\bar{\Phi}(\gamma)}
\end{aligned}
$$

These results are similar to the singly-aligned case for interacted concept c, with a few differences:

- Proficiency means of c in the update terms involving γ are replaced by projections of the mean vector along the direction of the weight:
$$\mu_c \to \mathbf{w}^T \boldsymbol{\mu}$$
- Similarly, proficiency variances of c in γ are replaced by an inner product with the weight vector:
$$\Sigma_{cc} \to \mathbf{w}^T \mathbf{\Sigma w}$$
- Finally, off diagonal elements involving the interacted concept c and another concept a in the numerator are changed:
$$\Sigma_{ac} \to (\mathbf{\Sigma w})_a$$

Observables, Latent Variables, and Symmetries of DIRT

The above equations have some interesting properties that inform practical model building with DIRT. Below we'll discuss what can be directly measured, what can be inferred, and symmetries/redundancies that exist in the model.

What can be predicted and observed?

We'll make a distinction here between three categories of "things", from more to less concrete and observable:

- **The Data**: The observed student interaction histories, including binary response data on questions (indexed by interaction time, student, concept, item, etc.), as well as instruction.
- **Observables**: These are things like a student's score on an exam or set of exams, performance of a population of students on an item, etc. In other words, things that can be predicted by our model, and can be measured, at least in principle. By this definition, some parameters referred to as "latent" might be observable, as we'll see below.

- **Unobservable latent parameters:** These are model parameters that cannot be measured, even in principle.

We'll be mainly concerned with the 2nd category, observables, since they are the main quantities predicted by our models. We list canonical examples in the table below. Note we've defined two commonly occurring scalar invariants of the mean vector and covariance matrix:

$$\text{Mean invariant } = M = \mathbf{w}^T \boldsymbol{\mu}$$

$$\text{Variance invariant } = V = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$$

Also note that in the table we've put in the usual item parameters for clarity.

Score/RP Observables and their DIRT Predictions		
Observable	**DIRT prediction**	**Invariants of knowledge state parameters ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$)**
Score median	$R_{50} = \Phi\left[\alpha\left(M - \beta\right)\right]$	M
Score mean	$\mu_R = \Phi(\gamma) = \Phi\left(\frac{\alpha\left(M-\beta\right)}{\sqrt{1+\alpha^2 V}}\right)$	M, V
Score variance[2] (See appendix for details)	$\text{var}\left(R\right) = \Phi(\gamma)\left(\Phi(\gamma^{(r=1)}) - \Phi(\gamma)\right)$ where $\gamma^{(r=1)} := \frac{\alpha\left(\mathbf{w}^T \boldsymbol{\mu}^{(r=1)} - \beta\right)}{\sqrt{1+\alpha^2 \mathbf{w}^T \boldsymbol{\Sigma}^{(r=1)} \mathbf{w}}}$ The notation r = 1 indicates that these quantities are the means and covariances that would arise after a hypothetical correct response on the item whose score variance we are computing.	M, V This can be seen by expanding out the new proficiency mean and covariance after r=1.
Score R_p at percentile p	$R_p = \Phi\left[\alpha\left(\Phi^{-1}_{M,V}(p) - \beta\right)\right]$	M, V

What is notable in the above table is that all observable predicted quantities can be written in terms of the two scalar invariants M and V. This motivates a theorem.

[2] This can be derived by starting with the definition of the score variance (involving a PDF times CDF[2] integral), and using the RNA approx on one of the (PDF x CDF) terms, leading to a PDF with new means and variances that would be obtained after a correct response (r = 1).

Theorems concerning invariants and observables

Theorem 1: *All observable score-related predictions on a set of items all with weight vector **w** can be written in terms of two scalar invariants M and V (defined above) of the current proficiency knowledge state KS = (**μ**, **Σ**). No other dependence on KS is possible.*

Proof: Any prediction related to the score distribution will be a function of an integral of some function f of $\mathbf{w}^T\boldsymbol{\theta}$ over the proficiency distribution:

$$I = \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta}) f(\mathbf{w}^T\boldsymbol{\theta})$$

To motivate this, note that **w** always appears with **θ** in the combination $\mathbf{w}^T\boldsymbol{\theta}$, and the full KS is specified by **θ** ~ N(**μ**, **Σ**). As a complicated example, note the defining equation for the score R_p at percentile p:

$$\int d^N\theta\ \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\ H\left(R_p - \Phi\left[\alpha\left(\mathbf{w}^T\boldsymbol{\theta} - \beta\right)\right]\right) = p$$

Here H(x) = 1 if x ≥ 0 else 0 denotes the Heaviside step function.

Let's now transform our variable of integration in `I` to a more convenient basis. Motivated by Cholesky decomposition:

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$$

We choose

$$\mathbf{x} = \mathbf{L}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})$$

This has Jacobian

$$|J| = |\mathbf{L}| = |\boldsymbol{\Sigma}|^{1/2}$$

The integral is now

$$I = \int d^N x\ \phi_{\mathbf{0},\mathbb{1}}(\mathbf{x}) f\left(\mathbf{w}^T\boldsymbol{\mu} + \mathbf{w}^T\mathbf{L}\mathbf{x}\right)$$

Next we perform another variable transformation, this time a pure N-dimensional rotation. Let $\mathbf{v} := \mathbf{L}^T\mathbf{w}$ and note that the function f only involves **x** via $\mathbf{v}^T\mathbf{x}$. Thus choose a rotation matrix **Λ**

$$\mathbf{y} := \boldsymbol{\Lambda}\mathbf{x}$$

$$\text{where } y_1 = \hat{\mathbf{v}}^T\mathbf{x}$$

We don't need to know the explicit form of **Λ**. Indeed there are many that satisfy our requirements since we only make demands on one element y_1. Note that the Jacobian of a pure rotation is 1, and further that we can now do all integrals y_2,..., y_N for free, leaving

$$\begin{aligned} I &= \int dy_1\ \phi(y_1) f\left(\mathbf{w}^T\boldsymbol{\mu} + \|\mathbf{v}\| y_1\right) \\ &= \int dy_1\ \phi(y_1) f\left(M + \sqrt{V} y_1\right) \end{aligned}$$

$$\text{since } \|\mathbf{v}\| = \sqrt{\mathbf{v}^T\mathbf{v}} = \sqrt{\mathbf{w}^T\mathbf{L}\mathbf{L}^T\mathbf{w}} = \sqrt{\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}} = \sqrt{V}$$

One final change of variables $z = M + \sqrt{V}y_1$ leads us to the result

$$I = \int dz \phi_{M,V}(z) f(z)$$

Thus, we have shown that any integral over the proficiency KS involving items aligned via **w** can be written purely in terms of M and V.

An alternative proof involves deriving the distribution of the score random variable R on an item aligned to concepts according to **w** with item parameters (α, β). The result can be derived using techniques similar to the above, resulting in

$$f_R(r) = \frac{\phi_{M,V}(\theta(r))}{\phi_{\beta,1/\alpha}(\theta(r))}$$

$$\text{where } \theta(r) := \Phi^{-1}_{\beta,1/\alpha}(r)$$

Thus the score distribution f_R involves only the invariants M and V. Next we'll generalize to the case where we make score-related predictions on a set of items with different types of multi-alignment.

Theorem 2: *All observable score-related predictions on a set of items with distinct weights vectors* {$\mathbf{w}_i$: i = 1,... N} *can be written in terms of the scalar invariants* M_i *and* V_{ij} *(defined below) of the proficiency knowledge state KS =* ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$)*. No other dependence on KS is possible.*

$$M_i = \mathbf{w}_i^T \boldsymbol{\mu}$$

$$V_{ij} = \mathbf{w}_i^T \boldsymbol{\Sigma} \mathbf{w}_j$$

$$\text{Note: } V_{ij} = V_{ji}$$

Proof: This has not been proven yet, so is more of a conjecture. But it holds in all the cases we care about. As an example, consider the score covariance terms between two items 1 and 2: cov(R_1, R_2). Note this occurs in the computation of the score variance for a multi-item exam the includes 1 and 2, for example. It's straightforward to show that in the RNA approx this covariance involves only invariants M_1, M_2, V_{11i}, V_{22}, and V_{12}. Similarly, the Core Assessment score can easily be written in terms of these invariants.

Theorem 3: *Consider a set of items with distinct weights vectors* {$\mathbf{w}_i$: i = 1,... N}. *All predictions of a DIRT model can be expressed solely in terms of the* ***prior*** *mean and covariance invariants* M_i *and* V_{ij}*. Thus, specifying these prior invariants fully specifies a predictive model.*

Proof: This follows easily from the recursive structure of the proficiency update equations in DIRT. Let's specialize to the case with as single weight vector **w** (i.e. a single type of multi-alignment) for simplicity. Taking suitable dot products of the update equations, we find:

$$\mathbf{w}^T \boldsymbol{\mu}^{(r)} = \mathbf{w}^T \boldsymbol{\mu} + \frac{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}}{Q} R_r(\gamma)$$

$$\mathbf{w}^T \boldsymbol{\Sigma}^{(r)} \mathbf{w} = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} + \frac{(\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w})^2}{Q^2} R_r(\gamma) \left[\gamma + R_r(\gamma)\right]$$

In other words the invariants evolve according to

$$M^{(r)} = M + \frac{V}{Q} R_r(\gamma)$$

$$V^{(r)} = V + \frac{V^2}{Q^2} R_r(\gamma) \left[\gamma + R_r(\gamma)\right]$$

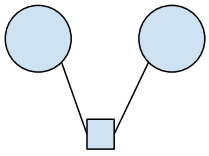
$$\text{where } Q = \sqrt{1 + V} \text{ and } \gamma = M/Q$$

So the new values after an interaction can be written solely in terms of the previous values. Thus, after any number of interactions, we can recursively write the current values of the invariants M, V in terms of their initial (prior) values. Combining this with Theorem 1 we obtain the desired result, that any prediction can be written only in terms of the prior invariants. The same argument holds with multiple types of MA, just with a little bit more complicated bookkeeping, and then using Theorem 2 at the end.

Symmetries and Equivalence Classes of Models

The above results on observables and invariants have important consequences for real-world model building. As we'll see now, they give rise to *equivalence classes* of predictive models defined by symmetries of the weight vectors.

Let's start with a concrete example. Consider the case where a student works solely on *singly-aligned* items on a concept C. After some interaction history, we can compute proficiencies and thus various physically observable predictions for other items in C. For example, the score median, the score mean, and scores at percentile might be of interest. Now imagine that we want to embed this concept in an LO L, and jointly learn proficiency on L and C. Thus we introduce DIRT MA with weights a and b for L and C, respectively, for each item i that was previously in C:



In this scenario, it would be strange if our actual predictions for this student changed, just because of this re-alignment. To emphasize, no interactions with other items of other alignments are considered here. This intuition is realized in the DIRT equations given above, in the form of symmetries that give rise to equivalence classes of models.

With weight vector $\mathbf{w}^T$ = (a, b), we see that any different choices of a and b that have equal values for the invariants *before* the interaction will also have equal invariants *after* the

interaction. In other words, for two different weight vectors (i.e. choices of a and b) $\mathbf{w_1}$ and $\mathbf{w_2}$ with corresponding invariants (M_1, V_1) and (M_2, V_2), respectively, we see that

$$(M_1 = M_2 \text{ and } V_1 = V_2) \implies (M_1^{(r)} = M_2^{(r)} \text{ and } V_1^{(r)} = V_2^{(r)})$$

Moreover, any such weight vectors that satisfy the equal-invariant condition initially (i.e. using the prior mean and covariance matrix), will have equal invariants after any number of interactions. This is really just a restating of Theorem 3 above: predictive models are completely defined by the prior invariants.

The weight vectors $\mathbf{w_1}$ and $\mathbf{w_2}$ are said to belong to the same *equivalence class* of predictive models. Consider the example above, with prior means of zero, and initial covariance matrix proportional to the identity: $\boldsymbol{\mu}_0^T = (0, 0)$ and $\boldsymbol{\Sigma}_0 = \text{diag}(V_0, V_0)$. Then we find the prior invariants:

$$\mathbf{w}^T \boldsymbol{\mu}_0 = 0$$

$$\mathbf{w}^T \boldsymbol{\Sigma}_0 \mathbf{w} = (a^2 + b^2) V_0$$

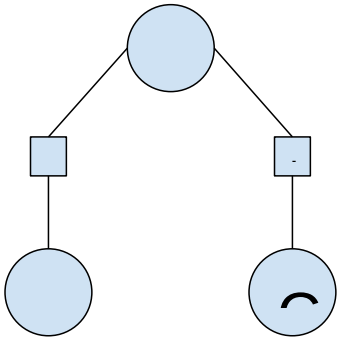
The above analysis means that we'd find exactly the same predictions for all choices of a and b that preserve the magnitude $a^2 + b^2$. In other words, any 2-D rotation of $\mathbf{w}$ provides physically indistinguishable models. Thus the following are equivalent models:

- Put all the weight on the concept, as is traditionally done: $\mathbf{w}^T = (0, 1)$
- Put all the weight on the LO (and none on the concept): $\mathbf{w}^T = (1, 0)$
- Divide the weight while preserving the magnitude of 1: $\mathbf{w}^T = (cos(\varphi), sin(\varphi))$

Similar result hold in higher dimensions, but for the LO / sub-LO system we'll be mainly interested in this 2-D rotational invariance.

Using DIRT Proficiencies for the LO / sub-LO System

We are now armed with the theory need to model the LO / sub-LO system. In this section 'concept' will refer only to a sub-LO (possibly the only one) of an LO. Below we show the generic setup we'll be interested in, with N concepts associated with the learning objective L. Each concept C_n has some set of items i_n that we align to only C_n and L. Thus, for now we are only considering what would normally be referred to as singly-aligned items (i.e. no concept-concept MA yet).



A few notes about the chosen weights:

- We've chosen the same apportionment of weights for each LO/concept pair: the LO gets c := $cos(\varphi)$ and the concept gets s := $sin(\varphi)$. While we could consider different values for each concept, perhaps in a data driven way, this is an unnecessary complexity now.
- We've chosen to normalize the weight vector magnitudes to 1. This can be done WLOG by absorbing the weight magnitudes into the prior covariance matrix.

The table below lists desired model requirements for this system.

Modeling Requirement	**Why do we need this?**	**How does this come about?**
Single LO/concept pairs (i.e. only one sub-LO) must have the same predictions and behavior as in the current production model	We'd like to make changes that are backward compatible, unless we explicitly want to change behavior for all LOs, for example by reconsidering recentered item difficulties.	The 2-D rotational invariance of the weights in the equivalence class of models with prior variance invariant V = 2.0, the current value of priorVariance in production.
Meaningful mastery predictions on the LO as a whole	Goal completion and LO-level analytics.	Judicious choice of the weight (c, s) in the (LO, concept) direction, so as to balance the correlation between concepts with the independent identity of each concept.
Meaningful mastery predictions on each concept (sub-LO)	Targeted recommendations, and perhaps sub-LO level analytics at some point.	Ditto ^

Note the importance of equivalence classes of models that allow us to vary behavior for multiple sub-LOs while leaving regular concepts (single sub-LOs) untouched.

Explaining Away

A consequence of DIRT is "explaining away", wherein proficiencies can move in counter-intuitive directions due to new interactions. For example, let's say you get 1 correct on L-C_1 (an item of concept 1 and L). The proficiency mean μ_1 on concept 1 will of course go up (as will the proficiency mean of L). A subsequent correct on L-C_2 will then *lower* μ_1, while also raising μ_2 and μ_L. It is straightforward to show that μ_1 will go down by an amount proportional to $sin(\varphi)\ cos^2(\varphi)$.

This behavior results from negative off-diagonal covariance matrix elements generated between a concept and LO due to interactions.

There are two important things to realize from this behavior:

1. DIRT is probably *not* a good fit for concept-concept multi-alignment. Consider 3 concepts A, B, and C with items of all types of multi-alignment between them. Let's say you start by doing well on A-B items. Then if you get things correct on A-C items, you'll see your B proficiency (and predictions on singly aligned B items) go down. This is not good predictive behavior.
2. This absolutely does NOT matter for our use case of LOs / sub-LOs. By construction, there are no items only aligned to C_1 that expose the drop of μ_1. Predictions for L-C_1 items will go *up* after corrects on L-C_2, by virtue of the upward movement of μ_L outweighing the downward movement of μ_1. All of the observable predictions of the model make sense and behave intuitively.

How do we compute mastery?

In the world of regular LOs (with one concept), we have just one proficiency (mean, variance) to care about. We currently compute mastery as the score-at-percentile on a hypothetical "default item" (i.e. of default item parameters) using this proficiency state. Here we extend this definition to the sub-LO realm where we have a proficiency mean vector $\boldsymbol{\mu}$ with elements for the LO L and the N sub-LOs C_i and a covariance matrix $\boldsymbol{\Sigma}$ containing arbitrary covariances between {L, C_1,..., C_N}.

First, note that we must figure out how to handle off-diagonal covariance matrix elements in the score variance. The current implementation assumes independent Normality of the proficiency posteriors, and then uses a so-called Owen's T function approximation. In the appendix, we present a new method for computing score variances in DIRT, consistent with the RNA framework. This method is both much simpler, and more robust, in that we can easily incorporate the full covariances.

Keeping with the same spirit as the current definition of mastery, for LO / sub-LO systems with N sub-LOs we do the following:

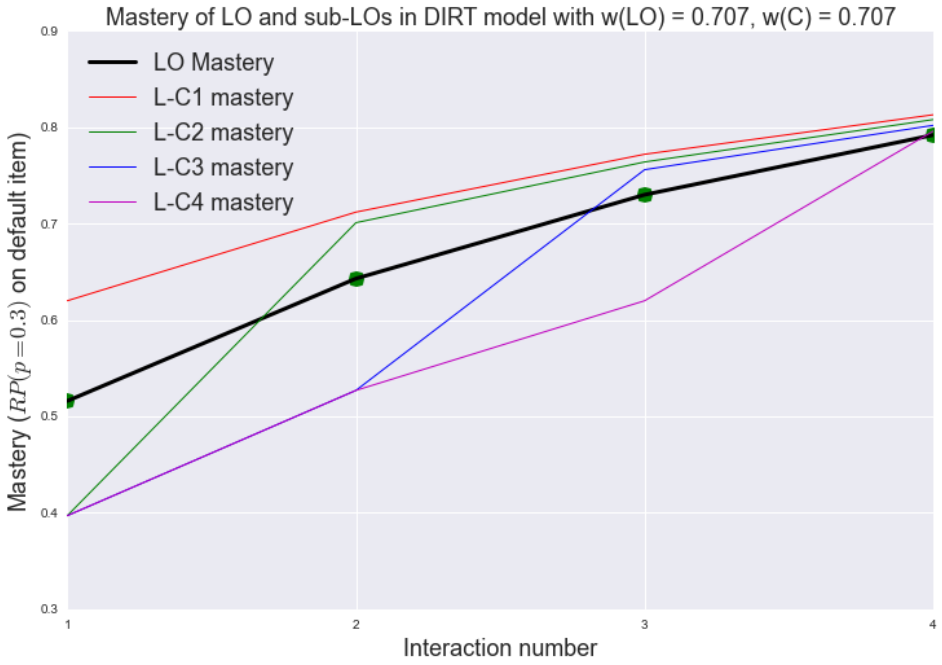
1. For each L-C_i pair with weight vector $\mathbf{w}_i$, we create a default item with discrimination 1, difficulty 0, and weight vector $\mathbf{w}_i$.
2. Compute the score mean as the sum of the N RP means of the default items, using the standard formula $\Phi(\gamma)$on each.
3. Compute the score variance on the set of N default items using the result for the RNA DIRT RP covariance given in the appendix.
4. Moment match the score mean and variance to the Beta distribution, and compute percentiles.

Note that we could (and probably should) reconsider the spirit of this algorithm, for example by reconsidering the recentering of item difficulties. However, we do no *need* to do this in order to

solve the problem of computing mastery for the LO/ sub-LO system. They are somewhat independent concerns.

Examples

Below is a plot of a 4 sub-LO system with 1 correct response on each concept (in order), with equal weight between the LO and concepts ($\sqrt{2}/2$). Connecting back to the discussion of explaining away, we note that the proficiency mean of C_1 goes to 0.65 after correct on L-C_1, then drops to 0.56, 0.49, and 0.43 after the next 3 corrects on other concepts.



Conjunctive IRT (CIRT)

The DIRT modeling framework assumes that assessed concepts can to some extent *compensate* for each other. What if this assumption is invalid? What if, in doing an item assessing concepts A, B, and C you definitely have to know how to do skills A, B, and C? This is where *conjunctive* IRT (CIRT) modeling comes in. We model the overall problem as being composed of subproblems assessing each concept, and that getting any of these sub-parts wrong will lead to an incorrect answer on the overall question. Since we don't have access to

the sub-part answers, we don't know exactly "which skill to blame" if you get it wrong. This uncertainty is handled in a principled way, as we'll see below.

For an item aligned to N concepts, the likelihoods are given by

$$P(r=1|\vec{\theta}) = \prod_{n=1}^{N} \Phi\left[\alpha_n\left(\theta_n - \beta_n\right)\right]$$

$$P(r=0|\vec{\theta}) = 1 - \prod_{n=1}^{N} \Phi\left[\alpha_n\left(\theta_n - \beta_n\right)\right]$$

Note that there are distinct item parameters associated with each assessed concept. In the DIRT case there were distinct weights, which play the role of distinct discriminations. But there are not distinct difficulties in the DIRT case. Note also that we explicitly include item parameters here in CIRT (in contrast to DIRT), since it seems that the only valid use case of CIRT is concept-concept multialignment.

The likelihood of an incorrect response automatically encodes all possible ways of getting the question wrong. In the N = 2 case, i.e. 2 sub-parts, you can get both parts wrong (WW), or just one part wrong (WR or RW). Letting f_i be the probability of getting sub-part i correct, we see that $1 - f_1f_2 = (1 - f_1)(1 - f_2) + (1 - f_1)f_2 + f_1(1 - f_2)$, and each term on the RHS corresponds to one of these three possible correctness patterns. This uncertainty on incorrect responses compared to correct responses will be born out in the results below. Generally speaking, the decrement to the proficiency mean and variance of one concept after an incorrect response will be less than in a similar singly-aligned case. This is related to the uncertainty as to which concept you failed on.

We'll give the mathematical results below when combining CIRT with DIRT, since the purely CIRT results can be easily obtained in the appropriate limit, and in any case we'll likely want to use this hybrid model.

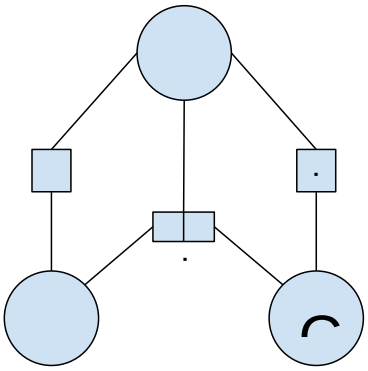
DIRT CIRT

Now we must face the task of combining our two types of multi-alignment:

- DIRT: needed to infer proficiencies on both LOs and concepts jointly
- CIRT: needed to handle genuinely multi-aligned questions

The diagram below represents this for a simple case where LO L has 2 sub-LO concepts C_1 and C_2 and there are three items:

- Items i_1 and i_2 are "singly-aligned", meaning they are DIRT multi-aligned to just one concept, plus the parent LO, with DIRT weights s and c, respectively.
- Item i_{12} is truly multi-aligned in the traditional sense, since it assesses both concepts, in addition to the LO. The item is drawn as a box with two distinct parts, representing the two distinct skills required to answer correctly, or perhaps even the two distinct steps in the solving process.



Thinking of the item i_{12} as being composed of two distinct sub-parts or steps each involving a different skill, it is clear what our likelihood should look like. It should be a product of DIRT likelihoods. In the general case where we have an item i assessing n distinct concepts, each of which is paired with exactly one LO, which are not necessarily the same for each concept (unlike drawn above), the likelihood is

$$P(r_i = 1|\boldsymbol{\theta}) = \prod_{c=1}^{n} \Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right]$$

Note that each weight vector $\mathbf{w}_c$ will have two non-zero elements, corresponding to the assessed concept c and the LO. The item parameters are specific to the item i and the concept c. Next we turn to the update equations of the model.

Proficiency Update Equations

As usual, the starting point is computing the normalization of the probability density, which is just the score mean (in the case of a correct response):

$$P(r_i = 1) = \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta}) \prod_{c=1}^{n} \Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right]$$

While the form of the integrals involved may seem intimidating, we'll see that in fact almost all of the work has already been done, and we just need to apply two sets of formulas: (a) the DIRT update equations, and (b) the Recursive Normal Approximation (RNA). Recall that the DIRT equations tell us how to compute integrals involving a multivariate Normal PDF times a CDF with arbitrary concept weights. And RNA tells us how to deal with multiple such CDF terms: we just apply moment-matching recursively at each step. In more detail, RNA applied to the first DIRT term is just

$$\phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\Phi\left[\alpha_{i1}\left(\mathbf{w}_1^T\boldsymbol{\theta} - \beta_{i1}\right)\right] \overset{\text{RNA}}{\approx} \Phi(\gamma_{i1})\phi_{\boldsymbol{\mu}^{(1)},\boldsymbol{\Sigma}^{(1)}}(\boldsymbol{\theta})$$

In the above, $(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)})$ is just the proficiency state after a hypothetical correct response on the first sub-part with item parameters $(\alpha_{i1}, \beta_{i1})$ and weight vector $\mathbf{w}_1$, as given in the DIRT update equations. The first term on the RHS is just the RP mean of that hypothetical stand-alone item,

and the gamma constant is usual combination of item parameters and the mean and variance invariants:

$$\gamma_{i1} = \frac{\alpha_{i1}\left(\mathbf{w}_1^T\boldsymbol{\mu} - \beta_{i1}\right)}{\sqrt{1+\alpha_{i1}^2\mathbf{w}_1^T\boldsymbol{\Sigma}\mathbf{w}_1}}$$

Applying the above procedure n times leads to the result for the RP mean of item i:

$$P_i := P(r_i = 1) = \prod_{c=1}^{n}\Phi\left(\gamma_{ic}^{(c-1)}\right)$$

$$\text{where } \gamma_{ic}^{(c-1)} := \frac{\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\mu}^{(c-1)} - \beta_{ic}\right)}{\sqrt{1+\alpha_{ic}^2\mathbf{w}_c^T\boldsymbol{\Sigma}^{(c-1)}\mathbf{w}_c}}$$

In the above, ($\boldsymbol{\mu}^{(c-1)}$, $\boldsymbol{\Sigma}^{(c-1)}$) is defined as the proficiency state after processing c - 1 hypothetical correct responses, in order. Thus for example ($\boldsymbol{\mu}^{(2)}$, $\boldsymbol{\Sigma}^{(2)}$) would be the state after the first two sub-parts were processed. While the notation is somewhat dense the idea is rather simple. The math of computing the RP mean of an item aligned to n concepts (via CIRT) is identical to that involved in computing the probability of getting correct responses on the hypothetical n sub-part questions, in sequence. Note that we chose an arbitrary ordering of the sub-parts. This is fine due to the robustness of the RNA. Different orderings will give very slightly different results.

Now let's turn to computing the proficiency mean and covariance updates. With the insights from above, these require very little new work to compute. The proficiency mean vector after a correct response on i is given by

$$\boldsymbol{\mu}^{(r_i=1)} = \frac{1}{P_i}\int d^N\theta\phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\prod_{c=1}^{n}\Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right]\boldsymbol{\theta}$$

Now we just perform the same iterative RNA procedure n times, which leads to the desired result. The new proficiency mean vector is just the value obtained by running the DIRT RNA update equations n times. There is nothing new to write down! However, the result for an incorrect response requires just a little work:

$$\boldsymbol{\mu}^{(r_i=0)} = \frac{1}{1-P_i}\int d^N\theta\phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\left(1-\prod_{c=1}^{n}\Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right]\right)\boldsymbol{\theta}$$

We can express the result in terms of the initial mean vector, and the result for a correct response, which we already know:

$$\boldsymbol{\mu}^{(r_i=0)} = \frac{1}{1-P_i}\left[\boldsymbol{\mu} - P_i\boldsymbol{\mu}^{(r_i=1)}\right]$$

Stepping back, we see that this is just saying that the expected value of the proficiency mean is unchanged by answering item i:

$$(1-P_i)\boldsymbol{\mu}^{(r_i=0)} + P_i\boldsymbol{\mu}^{(r_i=1)} = \boldsymbol{\mu}$$

More generally, all moments will have unchanged expected value due to the completeness of states {"correct", "incorrect"}.

Similar results hold for the covariance matrix update equation after a correct response on item i:

$$\Sigma_{ab}^{(r_i=1)} = \frac{1}{P_i} \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta}) \prod_{c=1}^{n} \Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right] \left(\boldsymbol{\theta} - \boldsymbol{\mu}^{(r_i=1)}\right)_a \left(\boldsymbol{\theta} - \boldsymbol{\mu}^{(r_i=1)}\right)_b$$

Applying the same logic as above, this is just the covariance matrix after n hypothetical correct responses, using the DIRT update equations recursively. Nothing new to write down. The final piece of the puzzle is the covariance matrix after an incorrect response:

$$\Sigma_{ab}^{(r_i=0)} = \frac{1}{1-P_i} \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta}) \left(1 - \prod_{c=1}^{n} \Phi\left[\alpha_{ic}\left(\mathbf{w}_c^T\boldsymbol{\theta} - \beta_{ic}\right)\right]\right) \left(\boldsymbol{\theta} - \boldsymbol{\mu}^{(r_i=0)}\right)_a \left(\boldsymbol{\theta} - \boldsymbol{\mu}^{(r_i=0)}\right)_b$$

After a little algebra, we can write this in terms of the already known covariance matrix:

$$\Sigma_{ab}^{(r_i=0)} = -\boldsymbol{\mu}_a^{(r_i=0)}\boldsymbol{\mu}_b^{(r_i=0)} + \frac{1}{1-P_i}\left[\left(\Sigma_{ab} + \boldsymbol{\mu}_a\boldsymbol{\mu}_b\right) - P_i\left(\Sigma_{ab}^{(r_i=1)} + \boldsymbol{\mu}_a^{(r_i=1)}\boldsymbol{\mu}_b^{(r_i=1)}\right)\right]$$

Analogous to the case of proficiency means, this just reflects that the expected value of the second moment remains unchanged given a response on i:

$$(1-P_i)\left(\Sigma_{ab}^{(r_i=0)} + \boldsymbol{\mu}_a^{(r_i=0)}\boldsymbol{\mu}_b^{(r_i=0)}\right) + P_i\left(\Sigma_{ab}^{(r_i=1)} + \boldsymbol{\mu}_a^{(r_i=1)}\boldsymbol{\mu}_b^{(r_i=1)}\right) = \Sigma_{ab} + \boldsymbol{\mu}_a\boldsymbol{\mu}_b$$

Core Assessment in RNA (CARNA)

The Core Assessment model (CA) is closely tied to the proficiency model, and serves an important role in the Recommender by prioritizing modules by the amount of information we would gain about the student knowledge state if the student were to interact with them. Before looking at CA in light of multi-alignment, let's review the current PROD implementation.

Current PROD model

The CA score for item i is defined as the expected value (over correctness r_i) of the reduction in variances across all concepts if the student interacted with i:

$$\begin{aligned} A(i) &= \left\langle \sum_{c=1}^{N_c} (\Sigma_{cc} - \Sigma_{cc}^{(r_i)}) \right\rangle_{r_i} \\ &= \mathrm{Tr}\left\langle \boldsymbol{\Sigma} - \boldsymbol{\Sigma}^{(r_i)} \right\rangle_{r_i} \end{aligned}$$

However, the current form of CA was put in place pre-RNA, when exact formulas for the covariance matrix were not available, and we could not efficiently estimate the would-be new value of the proficiency after an interaction. Thus, CA was implemented using various approximations and heuristics. The new RNA model (singly- and multi-aligned flavors) affords a more exact and correct definition of core assessment. In cases where **all** these conditions are met, the differences do not matter:

- Workflows chooses the concept to work without regard to the core assessment score

- Item discriminations are all constant (1)
- No multi-alignment
- No AB-only bundles (> 1 item)

In these stringent conditions, we'll always choose the item with difficulty closest to the proficiency mean on the concept in question. However, these conditions are not always met, even now. For example, several domains (C.A. Precalc + ?) have a lot of AB-only bundles. And with multi-alignment, we'll likely want to reconsider what it means to "work on a concept".

The current core assessment value for Workflows for a module m (possibly a bundle) assessing concept C is given by

$$A(m) = \frac{F(m)}{1 + F(m)V_C}$$

$$\text{where } F(m) := \text{ Fisher info } = \sum_{i \in m} \alpha_i^2 \frac{\phi^2(z_i)}{\Phi(z_i)(1 - \Phi(z_i))}$$

$$z_i = \alpha_i(\theta_C - \beta_i)$$

$$V_C := \text{proficiency variance of C}$$

Note the sum in the Fisher info is over any assessing items in the module m. This implies that, currently, purely assessing bundles (2 or more items) are upweighted wrt to single items in a concept. This was not done intentionally, as this form of the model was put out when we only had TB-AB bundles. We should intentionally decide how exactly we want to treat AB-bundles relative to single items, and make that happen with in concert with the CA-RNA options below.

Singly aligned RNA

Let's record the RNA core assessment value of a singly aligned item. Recall the definition of the core assessment score for item i, as the expected value of the sum of the variance reduction across all concepts:

$$A(i) = \text{Tr} \left\langle \mathbf{\Sigma} - \mathbf{\Sigma}^{(r_i)} \right\rangle_{r_i}$$

Since we have closed form equations for the covariance matrix updates, we can easily compute the CA score on item i assessing concept c as

$$A(i) = \frac{\alpha_i^2}{Q_i^2} \| \mathbf{\Sigma} \cdot \hat{e}_c \|^2 \frac{\phi^2(\gamma_i)}{\Phi(\gamma_i)\bar{\Phi}(\gamma_i)}$$

$$Q_i := \sqrt{1 + \alpha_i^2 \Sigma_{cc}}$$

$$\gamma_i := \frac{\alpha_i(\mu_c - \beta_i)}{Q_i}$$

This is a similar form to the PROD CA given above, where notably the same covariance-vector-squared factor is left out, simply because it represents inter-concept forces that are not relevant

for the current singly-aligned Workflows ranker. But in the CA score still used in the Classic ranker, there is exactly that same covariance-vector-squared factor.
The interpretation of the SA CARNA score is as follows:

- The covariance-vector-squared factor incorporates graph-connectivity effects from would-be assessed concept. Note this is the current *induced* graph connectivity, accounting for all interactions thus far. For example a concept with many interactions and a tight variance does not allow much proficiency to flow through edges connected to it.
- The remaining factors are a Bayesian version of the Fisher information.

The full CARNA score might be useful for review prioritization, for example.

Multi-item bundles

DIRT Core Assessment

We'll argue that a new definition of CA is needed for DIRT. Applying the existing definition of CA to the DIRT case would lead to a form proportional to $\mathbf{w}^T\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{w}$. This is just the extension of the covariance-vector-squared term in the singly aligned case. However, this scalar is *not* one of the physically meaningful invariants we discussed earlier. This raises alarm bells. As we have argued, in DIRT pure proficiency means or covariances are not meaningful. It is only their specific combinations with weight vectors that have meaning. Motivated by this, we posit the following form for DIRT CA, as the expected reduction in *variance invariants* due to an interaction. What variance invariants do we care about? Precisely the ones corresponding to multi-alignment patterns we are interested in. For the LO / sub-LO system, this is just the invariant for each LO-concept pair. For item i with weight vector $\mathbf{w}_i$ we define:

$$A(i) = \sum_{c \in L,C \text{ pairs}} \left\langle \mathbf{w}_c^T \left(\mathbf{\Sigma} - \mathbf{\Sigma}^{(r_i)} \right) \mathbf{w}_c \right\rangle_{r_i}$$

The sum is over all LO-concept pairs of interest. Working out the math, we find:

$$A(i) = \alpha_i^2 \sum_{c \in L,C \text{ pairs}} \frac{\left(\mathbf{w}_c^T \mathbf{\Sigma} \mathbf{w}_i \right)^2}{Q_i^2} \frac{\phi^2(\gamma_i)}{\Phi(\gamma_i)\bar{\Phi}(\gamma_i)}$$

$$Q_i := \sqrt{1 + \alpha_i^2 \mathbf{w}_i^T \mathbf{\Sigma} \mathbf{w}_i}$$

$$\gamma_i := \frac{\alpha_i \left(\mathbf{w}_i^T \boldsymbol{\mu} - \beta_i \right)}{Q_i}$$

Note the following:

- A(i) is written entirely in terms of our mean and variance invariants (M's and V's)
- In the limit of single-alignment, this reduces to the SA CARNA result above

DIRT-CIRT Core Assessment

Let's start with the definition of Core Assessment with DIRT from above, noting that we can pull the weight vectors outside of the expectation over response correctness:

$$A(i) = \sum_{c \in L,C \text{ pairs}} \mathbf{w}_c^T \left\langle \boldsymbol{\Sigma} - \boldsymbol{\Sigma}^{(r_i)} \right\rangle_{r_i} \mathbf{w}_c$$

Writing out the expectation, we find

$$\begin{aligned} \left\langle \boldsymbol{\Sigma} - \boldsymbol{\Sigma}^{(r_i)} \right\rangle_{r_i} &:= \boldsymbol{\Sigma} - P_i \boldsymbol{\Sigma}^{(r_i=1)} - \bar{P}_i \boldsymbol{\Sigma}^{(r_i=0)} \\ &= P_i \boldsymbol{\mu}^{(r_i=1)} \boldsymbol{\mu}^{(r_i=1)T} + \bar{P}_i \boldsymbol{\mu}^{(r_i=0)} \boldsymbol{\mu}^{(r_i=0)T} - \boldsymbol{\mu}\boldsymbol{\mu}^T \end{aligned}$$

In the second line we used the constancy of the 2nd moment. Next we can use the constancy of the 1st moment to obtain (after some algebra):

$$\left\langle \boldsymbol{\Sigma} - \boldsymbol{\Sigma}^{(r_i)} \right\rangle_{r_i} = \frac{P_i}{\bar{P}_i} \left(\boldsymbol{\mu}^{(r_i=1)} - \boldsymbol{\mu} \right) \left(\boldsymbol{\mu}^{(r_i=1)} - \boldsymbol{\mu} \right)^T$$

Finally, applying the weight vectors to both sides, we find

$$A(i) = \frac{P_i}{\bar{P}_i} \sum_{c \in L,C \text{ pairs}} \left(M_c^{(r_i=1)} - M_c \right)^2$$

$$\text{where } M_c := \mathbf{w}_c^T \boldsymbol{\mu}$$

$$\text{and } M_c^{(r_i=1)} := \mathbf{w}_c^T \boldsymbol{\mu}^{(r_i=1)}$$

Thus, we have rewritten the CA score from the sum of the expected value of the change in *variance invariants*, to the sum of the change in *mean invariants* given a correct response. The reader can verify that this reduces to the DIRT result given above in the singly-aligned case, using the fundamental DIRT proficiency update equations.

Invariant Space Representation

Here we'll recast all of the previous results in terms of only the mean and variance invariants, discussed in greater detail previously. We start by recalling that any observables of interest (score-at-percentiles, Core Assessment, etc.) can be written entirely in terms of these invariants. This suggests that maybe we should just work with the invariants from the beginning, and not bother with mean and covariance matrices in raw proficiency space. This is both possible and is (usually) computationally more efficient.

DIRT

Consider the set of all possible DIRT weight vectors that we care about. For example in the LO/sub-LO case, there will be one weight vector for each concept, with a non-zero element only

for itself, and its LO. Thus, there are N_C weight vectors, while the raw DIRT space is $N_{LO}+N_C$ dimensional. Let's dot these weight vectors into the DIRT proficiency update equations, to obtain the update equations in *invariant space*:

$$M_a^{(r_i)} := \mathbf{w}_a^T \boldsymbol{\mu}^{(r)} = M_a + \frac{V_{ai}}{Q_i} R_{r_i}(\gamma_i)$$

$$V_{ab}^{(r_i)} := \mathbf{w}_a^T \boldsymbol{\Sigma}^{(r_i)} \mathbf{w}_b = V_{ab} - \frac{V_{ai} V_{bi}}{Q_i^2} R_{r_i}(\gamma_i) \left[\gamma_i + R_{r_i}(\gamma_i)\right]$$

The indices (a, b) run from 1 to N_C, and we use 'i' to indicate the index of the interacted concept (we are considering just singly-aligned DIRT for now). Writing this in terms of matrices instead of components:

$$\mathbf{M}^{(r_i)} = \mathbf{M} + \frac{\mathbf{V}_i}{Q_i} R_{r_i}(\gamma_i)$$

$$\mathbf{V}^{(r_i)} = \mathbf{V} - \frac{\mathbf{V}_i \mathbf{V}_i^T}{Q_i^2} R_{r_i}(\gamma_i) \left[\gamma_i + R_{r_i}(\gamma_i)\right]$$

$$\text{where } \mathbf{V}_i := \mathbf{V} \cdot \hat{e}_i$$

Thus we see that for the LO/sub-LO system both the space and time complexity is $O(N_C^2)$, instead of $O((N_{LO}+N_C)^2)$ in raw proficiency space. When there is only one sub-LO per LO, this is a factor of 4 more efficient.

DIRT CIRT

The algorithm for multi-aligned item interaction updates is nearly identical in invariant space and raw proficiency space. First, we find the would-be mean and variance invariant matrices (M and V) assuming a correct response, as before by recursively applying the above DIRT results n-times for an n-aligned item. If the response was actually incorrect, we then use the constancy of 1st and 2nd moments, now in invariant space, to relate correct and incorrect updates:

$$M_a = P_i M_a^{(r_i=1)} + \bar{P}_i M_a^{(r_i=0)}$$

$$V_{ab} + M_a M_b = P_i \left(V_{ab}^{(r_i=1)} + M_a^{(r_i=1)} M_b^{(r_i=1)}\right) + \bar{P}_i \left(V_{ab}^{(r_i=0)} + M_a^{(r_i=0)} M_b^{(r_i=0)}\right)$$

The above formula relates the updates to M and V due to incorrect (r = 0) responses and correct (r = 1) responses. However, note that 3 outer products are required to get $V^{(r=0)}$ from $V^{(r=1)}$. We can actually reduce that to just one outer product, by using the first equation in the second. In matrix notation we find

$$\mathbf{V}^{(r_i=0)} = \frac{1}{\bar{P}_i} \left[\mathbf{V} - P_i \mathbf{V}^{(r_i=1)} - \frac{P_i}{\bar{P}_i} \left(\mathbf{M}^{(r_i=1)} - \mathbf{M}\right) \left(\mathbf{M}^{(r_i=1)} - \mathbf{M}\right)^T\right]$$

Benefits of Invariant Space Representation

The invariant space representation (ISR) reduces[3] the dimensionality of the space we need to work in, and is thus analogous to the "kernel trick". The full raw proficiency representation contains redundant information that can be compressed. This reduction has several consequences:

- **Efficient Proficiency Model**
  Clearly, the update equations for an item interaction will be more efficient: $O(N_C^2)$ instead of $O((N_{LO}+N_C)^2)$. Note that graph-propagation requires us to work in the N_C dimensional space, since for example an item interaction on concept i induces changes to the invariants of concepts (a, b) of a different LO via the terms V_{ai} and V_{bi} above.
- **Efficient Observables/Endpoints**
  Computing observables will also be more efficient. Note that both masteries (i.e. score-at-percentiles) and Core Assessment (CA) require running the DIRT update equations. For mastery, we do this to compute the score variance in the RNA framework, whereas for DIRT-CIRT CA we need the mean (invariant) after a correct response. In fact, since these observables only involve concepts within an LO, we can work in a much smaller space while computing these quantities, namely just considering the concepts of one LO. For masteries, this is true because we only compute LO mastery, not something like overall topic mastery, which *would* require inter-LO cross terms. For CA, we only care about the assessing the already-chosen LO.
- **Efficient Knowledge State Storage**
  Given the last point, we only need to keep (*after* processing all interactions) the variance invariants V_{ab} where (a, b) are concepts of the same LO. This typically means we'll need to store (some small integer) x N_c invariants, instead of the full N_c x N_c matrix **V**. Note that we *do* need the full **V** while processing interactions, due to propagation. But the proficiency object containing endpoints/observables does not need most of these elements (unless we need to compute observables with inter-LO correlation).

Recency and Instructional Effects

We previously derived constraints on the form that recency and instructional Gaussian Process effects should take, based on avoiding spurious accumulations of proficiency. The idea is simple:

- Imagine we start at some initial proficiency state ($\boldsymbol{\mu}_c$, $\boldsymbol{\Sigma}_{cc}$) on concept c, and then answer a series of questions, getting some correct and some incorrect.
- Now consider if combined effects of this data and the application of recency leads us exactly back to the initial state ($\boldsymbol{\mu}_c$, $\boldsymbol{\Sigma}_{cc}$) on c, with no net change in the proficiency mean or variance.

Commented [8]: +illya@knewton.com +andrew.jones@knewton.com
In case you are looking for some riveting late night reading on the technical details of recency in ISR, you are in luck. This ended up being longer than I thought, but is complete.

[3] Only in cases where the number of weight vectors is less than the dimensionality of the raw proficiency space, which need not always be the case. For example, consider jointly learning proficiencies, item difficulties, and interaction-indexed concept learning parameters. In that case the invariant space representation can be much larger than the raw proficiency space.

- We would then expect that any *propagated* (or more generally correlations from other covariance terms) effects to other concepts should also vanish.
- Insisting that this is the case uniquely constrains the form of the recency update to the covariance matrix, making recency *propagate* in exactly the same way as item interaction updates.
- The resulting Gaussian Process is no longer strictly a Wiener process, but instead a hybrid temporal-conceptual multivariate Gaussian process. This means that the *index set* of the GP is (item interaction indexed) time, and concept space.
- Analogous results holds for mean increments applied at each item interaction, as well as for additive instructional effects.

We'll show here that the same basic results hold in the ISR of DIRT-CIRT RNA, and also consider more general forms of recency where the strength of the update is state-dependent. We'll also consider different possible orderings in applying the item interaction update and recency update.

Notation and Setup

Let's consider the usual set-up where we have some latent multivariate knowledge state (**μ**, **Σ**) living in a space consisting of LOs and concepts and of dimension $N_{LO} + N_C$. We can express the update equations and any observable quantities purely in terms of the invariant space representation (ISR) knowledge state (**M**, **V**) living in a space of dimension N_C. Going forward, let's be explicit about the two effects of an interaction on item i: r_i denotes the purely IRT effects, and t denotes the temporal (recency) effects. Let's write the IRT update equations (without recency) for a singly-aligned DIRT interaction on item i assessing concept c with item parameters (α_i, $\boldsymbol{\beta}_i$) in compact notation:

$$\mathbf{M}^{(r_i)} = \mathbf{M} + y_i \mathbf{x}$$

$$\mathbf{V}^{(r_i)} = \mathbf{V} - z_i \mathbf{x}\mathbf{x}^T$$

$$\text{where } \mathbf{V}_c := \mathbf{V} \cdot \hat{e}_c$$

$$\mathbf{x} := \frac{\mathbf{V}_c}{V_{cc}}$$

$$y_i := \frac{\alpha_i V_{cc}}{Q_i} R_{r_i}(\gamma_i)$$

$$z_i := \left(\frac{\alpha_i V_{cc}}{Q_i}\right)^2 R_{r_i}(\gamma_i)\left[\gamma_i + R_{r_i}(\gamma_i)\right]$$

$$Q_i = \sqrt{1 + \alpha_i^2 V_{cc}}$$

$$\gamma_i = \frac{\alpha_i \left(M_c - \beta_i\right)}{Q_i}$$

Note that the scalars y_i and z_i depends only on the item parameters of item i and the knowledge state s of the interacted concept, s := (M_c, V_{cc}). We will make this *state-dependence* explicit going forward, and also drop the 'i' index from now on: y(s), z(s). Also note that since $\mathbf{x}_c = 1$, these y(s), z(s) quantities represent the increment/decrement to the proficiency mean/variance invariants of the interacted concept c.

Let's now write the pure recency update as

$$\mathbf{M}^{(t)} = \mathbf{M}$$

$$\mathbf{V}^{(t)} = \mathbf{V} + \eta(s)\mathbf{x}\mathbf{x}^T$$

We've jumped the gun a bit by assuming the form of the recency covariance update as $\mathbf{x}\mathbf{x}^T$, but we'll justify that later. Also note we've included state-dependence in the parameter η(s) controlling the strength of the recency increment, to maintain generality[4].

We chose the above parametrization because of its compactness, and because the **x** vector has nice properties. In addition to $\mathbf{x}_c = 1$, note that by using $\mathbf{V}_c = V_{cc}\mathbf{x}$ we can show that **x** is invariant under both IRT and temporal updates:

$$\mathbf{x}^{(r)} := \frac{\mathbf{V}_c^{(r)}}{V_{cc}^{(r)}} = \frac{\mathbf{V}_c - z(s)\mathbf{x}}{V_{cc} - z(s)} = \mathbf{x}$$

$$\mathbf{x}^{(t)} := \frac{\mathbf{V}_c^{(t)}}{V_{cc}^{(t)}} = \frac{\mathbf{V}_c + \eta(s)\mathbf{x}}{V_{cc} + \eta(s)} = \mathbf{x}$$

Three Orderings of Recency

Now that we've gone to all this notational trouble, let's make use of it. We typically think of recency and IRT effects as happening "at the same time". But what does this mean? We can imagine three different interpretations: *Simultaneous Recency (SR)*, *Recency First (RF)*, and *Recency Last (RL)*.

Simultaneous Recency (SR)

This defines the combined IRT and temporal updates *simultaneously* as:

$$\mathbf{M}^{SR} = \mathbf{M} + y(s)\mathbf{x}$$

$$\mathbf{V}^{SR} = \mathbf{V} + \left(\eta(s) - z(s)\right)\mathbf{x}\mathbf{x}^T$$

This is what is currently done in our production proficiency model. We'll use this as a baseline against which to compare the other two options.

Recency First (RF)

Here we apply the temporal effects first, i.e. infinitesimally before the IRT update, then use the updated state as the inputs to the IRT update:

[4] We've only used constant (i.e. state-independent) η thus far in production proficiency models.

$$\mathbf{M}^{(RF)} = \mathbf{M}^{(t)} + y(s^{(t)})\mathbf{x}^{(t)}$$
$$\mathbf{V}^{(RF)} = \mathbf{V}^{(t)} - z(s^{(t)})\mathbf{x}^{(t)}\mathbf{x}^{(t)T}$$

This can be simplified using the invariance of **x** and the formulas above:

$$\begin{aligned}\mathbf{M}^{(RF)} &= \mathbf{M} + y(s^{(t)})\mathbf{x} \\ &= \mathbf{M}^{(SR)} + \left(y(s^{(t)}) - y(s)\right)\mathbf{x} \\ \mathbf{V}^{(RF)} &= \mathbf{V} + \left(\eta(s) - z(s^{(t)})\right)\mathbf{x}\mathbf{x}^T \\ &= \mathbf{V}^{(SR)} - \left(z(s^{(t)}) - z(s)\right)\mathbf{x}\mathbf{x}^T\end{aligned}$$

It can be shown that for positive η(s) we have $|y(s^{(t)})| > |y(s)|$ and $z(s^{(t)}) > z(s)$. Thus, compared to SR, RF leads to greater upward/downward movement in the mean invariants upon correct/incorrect, and greater downward movement to the variance invariants (regardless of correctness). This makes sense given that we are increasing the variance *before* applying the IRT effects. These properties make RF a compelling alternative to SR, which sometimes leads to undesirable behavior where the variance goes *up* after an item interaction. RF will mitigate this behavior.

Recency Last (RL)

Here we apply the IRT effects first (infinitesimally earlier), then use that state as the inputs to the temporal update:

$$\begin{aligned}\mathbf{M}^{(RL)} &= \mathbf{M}^{(r)} = \mathbf{M} + y(s)\mathbf{x} \\ &= \mathbf{M}^{(SR)} \\ \mathbf{V}^{(RL)} &= \mathbf{V}^{(r)} + \eta(s^{(r)})\mathbf{x}^{(r)}\mathbf{x}^{(r)T} \\ &= \mathbf{V} + \left(\eta(s^{(r)}) - z(s)\right)\mathbf{x}\mathbf{x}^T \\ &= \mathbf{V}^{(SR)} + \left(\eta(s^{(r)}) - \eta(s)\right)\mathbf{x}\mathbf{x}^T\end{aligned}$$

Note that RL is the same as SR *if* we choose a constant (i.e. state-independent) η parameter, as is currently done in production. However, choosing state-dependent η(s) can lead to either higher or lower variances compared to SR, dependent on the functional form chosen.

Recency Constraints in ISR

Here we'll justify the choice made above that recency effect "propagate" just like item interactions: $\mathbf{V}^{(t)} = \mathbf{V} + \eta(s)\mathbf{x}\mathbf{x}^T$. This follows the previous analysis, but in the new notation. Consider N item interactions on concept c, with mean updates:

$$\mathbf{M}^{(n)} = \mathbf{M}^{(n-1)} + y^{(n)}\mathbf{x}^{(n-1)}$$

$$\text{where } y^{(n)} := y(\alpha_n, \beta_n, s^{(n-1)})$$

$$\mathbf{x}^{(n-1)} = \frac{\mathbf{V}_c^{(n-1)}}{V_{cc}^{(n-1)}}$$

We've indicated that $y^{(n)}$ depends on the n^{th} item parameters and the state $s^{(n-1)}$ after the n-1st interaction, including both IRT and recency effects. While we've chosen the same notation ('y') above that only includes the IRT effects, we'll find that our results hold even if we include recency effects into the y's as well, and they will hold for all three orderings of recency discussed above. Note that

$$M_c^{(N)} = M_c^{(0)} + \sum_{n=1}^{N} y^{(n)}$$

Thus, if M_c remains unchanged after N interaction, we must have the sum of $y^{(n)}$ is zero. Next notice that for some other concept p we have

$$M_p^{(N)} = M_p^{(0)} + \sum_{n=1}^{N} y^{(n)} x_p^{(n-1)}$$

Thus, if we want M_p to be unchanged as well (i.e. no spurious proficiencies), we need the sum of $y^{(n)}x_p^{(n-1)}$ terms to be zero. Neglecting the possibility of a conspiracy between covariance terms and future item parameters, we conclude that we must have $x_p^{(n-1)}$ constant for all n. This means we must have for all n and all concepts p:

$$\frac{V_{pc}^{(n)}}{V_{cc}^{(n)}} = \frac{V_{pc}^{(n-1)}}{V_{cc}^{(n-1)}}$$

This follows automatically for pure IRT updates (i.e. without recency), but implies non-trivial constraints with recency. Let's write the recency update in the most general form in terms of any symmetric real matrix A:

$$\mathbf{V}^{(t)} = \mathbf{V} + \eta(s)\mathbf{A}$$

We'll also choose A_{cc} = 1 WLOG, since we could just absorb this state-dependent constant into the parameter η(s). For concreteness, we'll work in the *simultaneous recency* (SR) ordering, though as discussed below the same conclusions will hold regardless of ordering. The iterative update for the variance invariant matrix is (note both IRT and recency effects are implied in the superscripts n and n-1 now):

$$\mathbf{V}^{(n)} = \mathbf{V}^{(n-1)} + \eta(s^{(n-1)})\mathbf{A}^{(n-1)} - z^{(n)}\mathbf{x}^{(n-1)}\mathbf{x}^{(n-1)T}$$

Note as for $y^{(n)}$ above, $z^{(n)}$ depends on the n^{th} item parameters and n-1st state. Let's solve for $z^{(n)}$ from the (c,c) component of this equation, and substitute into the (p,c) component, leading to:

$$V_{pc}^{(n)} = V_{pc}^{(n-1)}\frac{V_{cc}^{(n)}}{V_{cc}^{(n-1)}} + \eta(s^{(n-1)})\left(A_{pc}^{(n-1)} - \frac{V_{pc}^{(n-1)}}{V_{cc}^{(n-1)}}\right)$$

First we note that in the absence of recency ($\eta = 0$), we have the desired constancy of covariance ratios, as claimed above. However, with recency we find the constraint:

$$A_{pc}^{(n-1)} = \frac{V_{pc}^{(n-1)}}{V_{cc}^{(n-1)}}$$

The remaining off-diagonal matrix elements of **A** are found by a similar (and more tedious) analysis demanding we avoid spurious accumulations in the *covariance matrix*. In short, if we have $V^{(N)}_{cc} = V^{(0)}_{cc}$, we also want $V^{(N)}_{pc} = V^{(0)}_{pc}$. Going through the math, we find that this requires

$$A_{pq}^{(n-1)} = \frac{V_{pc}^{(n-1)} V_{qc}^{(n-1)}}{\left(V_{cc}^{(n-1)}\right)^2}$$

In conclusion, we find the avoiding spurious proficiency mean and covariance biases leads us uniquely to the form recency must take. It must "propagate" exactly like item interactions, i.e. in our compact notation $\mathbf{A} = \mathbf{x}\mathbf{x}^T$. By being careful with indices, it is straightforward to show that this form holds for all three orderings of recency discussed (SR, RF, and RL).

Graph Propagation and the Prior V Matrix

We previously argued that there exists an equivalence class of predictive models parameterized by an angle cos(ϕ) that governs the alignment items have to their LO and their concept. The models in each class give the same predictions (thus "equivalence") when concepts and LOs are in 1-1 correspondence. This allows us to generalize the model to handle multiple sub-LOs and multi-alignment, without changing the behavior for regular non-multi-anything concept/LOs. This is all true and good, but not the full story. There is an additional subtlety related to graph-structured propagation that can break exact equivalence depending on how we handle it.

First let's note that the ISR update equations are exactly of the same form as the regular mvRNA update equations. So if at any point we have $\mathbf{M} = \boldsymbol{\mu}_{mvrna}$ and $\mathbf{V} = \boldsymbol{\Sigma}_{mvrna}$, then subsequent evolution will maintain equality. Note that the invariant matrices are defined via $M_a := \mathbf{w}_a^T \boldsymbol{\mu}$ and $V_{ab} := \mathbf{w}_a^T \boldsymbol{\Sigma} \mathbf{w}_b$ in terms of the full $N_{LO} + N_C$ dimensional ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$) matrices and weight vectors $\mathbf{w}_a$ defined for each concept 'a'., Therefore, we only need to verify that the prior values for ISR M, V matrices are the same as the mvRNA matrices ($\boldsymbol{\mu}_{mvrna}$, $\boldsymbol{\Sigma}_{mvrna}$) in order to prove exact equivalence in the non-MA limit.

Consider the full DIRT LO + concept space prior covariance matrix $\boldsymbol{\Sigma}$, and let's assume it is composed of block-diagonal LO (L) and concept (C) covariance matrices:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_L & 0 \\ 0 & \boldsymbol{\Sigma}_C \end{pmatrix}$$

Given the parametrization of the weight vectors **w** in terms of the angle cos(ϕ), and defining the LO index of concept 'a' as L(a), we have

$$V_{ab} = \cos^2(\phi)\,(\mathbf{\Sigma}_L)_{L(a),L(b)} + \sin^2(\phi)\,(\mathbf{\Sigma}_C)_{ab}$$

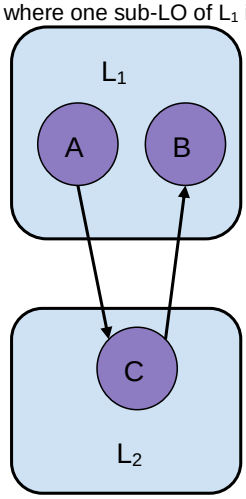
Let's stop and note that we conventionally incorporate graph-structured propagation in the prior concept inverse covariance matrix (i.e. information matrix) via:

$$\mathbf{\Sigma}_C^{-1} := v_0\,(1-\mathbf{W})\,(1-\mathbf{W})^T$$

Here v_0 is the prior variance scale and **W** is the upper-diagonal weight matrix (unrelated to the weight *vectors*) defined by the prereqs.

Now, in the case where LO's and concepts are 1-1, we can decide that the LO prior is exactly the same as the concept prior, and we'd find the dependence on cos(ϕ) disappears and we have identical prior covariance values: $\mathbf{V} = \mathbf{\Sigma}_C$. Then we can be assured that, for example, $M_a := \cos(\phi)\,\mu_{L(a)} + \sin(\phi)\,\mu_a$ in ISR MARNA evolves identically to μ_a in regular mvRNA. Moreover, all physical predictions of the two theories will be the same.

However, we have not addressed how to generalize to the case where LOs and concepts are *not* 1-1. Drawing prereq edges between LOs becomes more challenging. Consider the case below where one sub-LO of L_1 is prereq and another is postreq to the sub-LO of L_2.



We could come up with some rules for handling all these scenarios, such that in the limiting case of 1-1 LO-concept correspondence we retain equivalence (i.e. the LO prior covariance is the same as the concept prior covariance). However, this extra modeling step seems unnecessary. We can also choose a simpler prior where the LO covariance is proportional to the identity matrix, thus leading to:

$$V_{ab} = \cos^2(\phi) v_0 \delta_{L(a),L(b)} + \sin^2(\phi)\,(\mathbf{\Sigma}_C)_{ab}$$

This model will be equivalent to mvRNA for island concepts, but different in general, due to propagation only occurring through the $\sin^2(\phi)$ weighted concept-covariance. Thus, in general, propagation will be lighter than in the regular mvRNA case, given the same weight-penalty parameters.

Algorithm

Let's collect the algorithm for how to work purely in invariant space:

1. Choose prior proficiency mean vector $\boldsymbol{\mu} = \mathbf{0}$ and therefore mean invariant vector $\mathbf{M} = \mathbf{0}$
2. Compute the concept-concept prior covariance matrix $\boldsymbol{\Sigma}_{con}$, including any graph-structured propagation terms.
3. Compute the prior variance invariant matrix $\mathbf{V}$, where $V_{ab} = \mathbf{w}_a^T \boldsymbol{\Sigma} \mathbf{w}_b$, and $\boldsymbol{\Sigma}$ represents the full DIRT LO + concept space covariance matrix, which is a block diagonal combination of LO covariance matrix $\boldsymbol{\Sigma}_{LO}$ and the concept covariance matrix $\boldsymbol{\Sigma}_{con}$. However, $\boldsymbol{\Sigma}_{LO}$ is just the prior variance parameter v_0 times the identity matrix, since we choose to include graph structured propagation via concepts. Note that each weight vector has two non-zero values: cos(ϕ) at the LO index, and sin(ϕ) at the concept index. Thus the (a, b) element of the variance invariant matrix is just
$$V_{ab} = \sin^2(\phi)\,(\Sigma_{con})_{ab} + \cos^2(\phi) v_0 \delta_{LO(a),LO(b)}$$
Note the second term is non-zero only if the concepts at indices a, b share the same LO.
4. Process interactions to update the invariants according to the DIRT or DIRT-CIRT algorithms given above, keeping all N_C^2 terms in the matrix $\mathbf{V}$.
5. Once all interactions have been processed, keep a mapping from each LO L to its own variance invariant matric $\mathbf{V}_L$, obtained by trivially integrating out all concepts not belonging to the LO. For LO's with n sub-LO concepts, this will be an n x n (symmetric) matrix. Of course, we'll also keep the full $\mathbf{M}$ vector containing the N_C mean invariants.
6. Compute masteries or CA on demand, using the efficient representation of the knowledge state described in #5, and the update equations for invariants as needed.

Proof of ISR via Integral Variable Transformation [In progress]

Earlier we derived ISR equations by noticing that (a) all observables can be written in terms of mean and variance invariants, and then (b) dotting weight vectors into the latent space representation (LSR) proficiency mean and covariance update equations, to obtain update equations in invariant space. There is an alternative, perhaps more fundamental derivation, which we'll outline here.

Now we'll modify our starting point slightly and demand that any observable

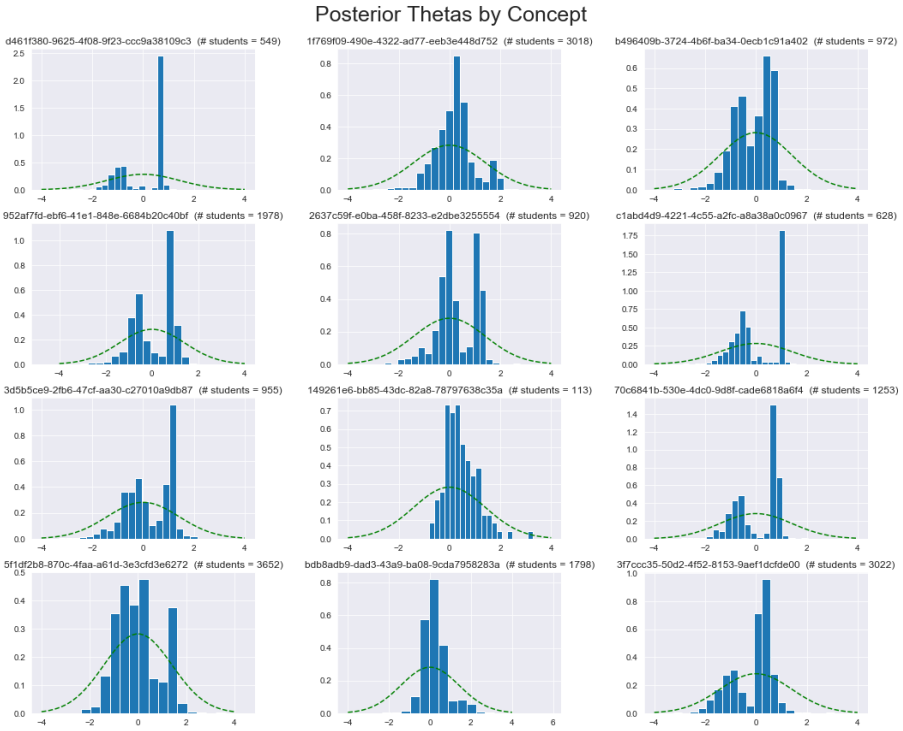
Gaussian Mixture Priors

It may be the case that our assumption of Gaussian proficiency priors is not robust. Perhaps there are often two classes of students, those who are prepared and those who are under-prepared and tend to struggle. One piece of suggestive evidence comes from looking at posterior proficiency distributions. Consider the posterior proficiency histograms shown below, where we see fairly strong bimodal tendencies in several concepts.

This can be modelled using a Gaussian mixture prior:

$$\rho(\boldsymbol{\theta}) := \sum_{i=1}^{n} a_i \phi_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}(\boldsymbol{\theta})$$

$$\text{where } \sum_{i=1}^{n} a_i = 1$$



The prior mean and covariance can be straightforwardly found to be

$$\boldsymbol{\mu} = \sum_{i=1}^{n} a_i \boldsymbol{\mu}_i$$

$$\boldsymbol{\Sigma} = \sum_{i=1}^{n} a_i \left[\boldsymbol{\Sigma}_i + (\boldsymbol{\mu} - \boldsymbol{\mu}_i)(\boldsymbol{\mu} - \boldsymbol{\mu}_i)^T \right]$$

Next we compute the probability of response r (0 or 1) on an item, and find it is a linear combination of the probabilities of each component:

$$P_r = \sum_{i=1}^{n} a_i \Phi_r(\gamma_i)$$

$$\text{where } \Phi_r(\gamma_i) = \Phi(\gamma_i) \text{ for } r = 1 \text{ else } 1 - \Phi(\gamma_i)$$

$$\text{and } \gamma_i := \frac{\mathbf{w}^T \boldsymbol{\mu}_i}{\sqrt{1 + \mathbf{w}^T \boldsymbol{\Sigma}_i \mathbf{w}}} := \frac{\mathbf{w}^T \boldsymbol{\mu}_i}{Q_i}$$

After an item interaction r, we find the form of the equations relating overall means and covariances to components' means and covariances remains unchanged:

$$\boldsymbol{\mu}^{(r)} = \sum_{i=1}^{n} a_i^{(r)} \boldsymbol{\mu}_i{}^{(r)}$$

$$\boldsymbol{\Sigma}^{(r)} = \sum_{i=1}^{n} a_i^{(r)} \left[\boldsymbol{\Sigma}_i^{(r)} + (\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu}_i^{(r)})(\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu}_i^{(r)})^T \right]$$

In the above, the individual component means and covariances evolve according to the usual RNA (DIRT in the general case) equations (i.e. exactly as if they were the only prior):

$$\boldsymbol{\mu}_i^{(r)} = \boldsymbol{\mu}_i + \frac{\boldsymbol{\Sigma}_i \mathbf{w}}{Q_i} R_r(\gamma_i)$$

$$\boldsymbol{\Sigma}_i^{(r)} = \boldsymbol{\Sigma}_i - \frac{(\boldsymbol{\Sigma}_i \mathbf{w})(\boldsymbol{\Sigma}_i \mathbf{w})^T}{Q_i^2} R_r(\gamma_i) \left[\gamma_i + R_r(\gamma_i) \right]$$

Finally, perhaps the most interesting part is the evolution of the a_i weights:

$$a_i^{(r)} = \frac{a_i \Phi_r(\gamma_i)}{P_r}$$

Thus, each components' weight adjusts depending on how well it predicted the response r.

Appendix 1: DIRT Math

Here's we'll treat the general case of DIRT MA, with likelihood

$$P(r = 1|\vec{\theta}) = \Phi\left(\sum_{c=1}^{N} w_c \theta_c \right) = \Phi\left(\vec{w} \cdot \vec{\theta} \right)$$

$$\text{where } \vec{\theta} := (\theta_1, \ldots, \theta_N)$$

The prior is assumed to be Gaussian:

$$\rho(\vec{\theta}) = \phi_{\vec{\mu}, \boldsymbol{\Sigma}}(\vec{\theta})$$

This leads to the posterior

$$\rho(\vec{\theta}|r) = \frac{1}{P_r} \Phi\left((-1)^{r-1} \vec{w} \cdot \vec{\theta} \right) \rho(\vec{\theta})$$

We'll want to derive the 0th moment of the unnormalized distribution (P_r), the 1st moment (proficiency means), and the 2nd moment (proficiency covariances). Before proceeding with the derivations, we'll collect some integrals that will be useful.

Useful Integrals

First let's recall the scalar integrals used in singly-aligned RNA:

$$I_n := \int_{-\infty}^{\infty} dx x^n \phi(x) \Phi\left(A + Bx\right)$$

The results for n = 0, 1, and 2 are

$$I_0 = \Phi\left[\frac{A}{\sqrt{1+B^2}}\right] := \Phi(\gamma)$$

$$I_1 = \frac{B}{\sqrt{1+B^2}} \phi\left[\frac{A}{\sqrt{1+B^2}}\right] := \frac{B}{A}\gamma\phi(\gamma)$$

$$\text{where } \gamma := \frac{A}{\sqrt{1+B^2}}$$

$$I_2 = \Phi(\gamma) - \frac{B^2}{A^2}\gamma^3\phi(\gamma)$$

We'll need generalizations of these to higher dimensions. First consider

$$I_0^{(N)} := \int d^N x \phi(\vec{x}) \Phi\left(A + \vec{B} \cdot \vec{x}\right)$$

Note that $\boldsymbol{\phi}$ represents the N-dimensional standard Normal PDF, i.e. the product of N scalar Gaussians. This integral (and the ones below) can be derived in at least two ways:

1. Straightforwardly apply the formula for I_0 N-times, and build up the result iteratively.
2. Choose a change of variables that is a linear rotation of **x** such that one of the new variables is just the unit vector $\hat{B} \cdot x$. The Jacobian for a pure length-preserving rotation is just 1, and we now can do N - 1 integrals "for free", leaving only the univariate PDF x CDF integral.

Either way, the result is

$$I_0^{(N)} = \Phi\left(\frac{A}{\sqrt{1+\|\vec{B}\|^2}}\right) := \Phi\left(\frac{A}{Q}\right) := \Phi(\gamma)$$

Note we introduced some compact notation Q and γ which will be used below.

Next, we'll need the generalization of I_1 in order to compute proficiency means:

$$\vec{I}_1^{(N)} := \int d^N x \vec{x} \phi(\vec{x}) \Phi\left(A + \vec{B} \cdot \vec{x}\right)$$

The derivation follows similarly to that of $I_0^{(N)}$. We'll sketch out the iterative approach (#1 above). With the x-vector in the integrand, we should be careful about the order of integration. Consider the N^{th} component, and note we can easily do the first N-1 integrals in exactly the same way as for $I_0^{(N)}$. Then we perform the final integral over x_N using the scalar integral I_1 above. The result for other components are analogous, by symmetry.. The final result is thus

$$\vec{I}_1^{(N)} = \frac{\vec{B}}{Q}\phi(\gamma)$$

Finally, we'll need the analog of I_2 in order to compute the covariance matrix:

$$I_{ab}^{(N)} := \int d^N x x_a x_b \phi(\vec{x})\Phi\left(A + \vec{B}\cdot\vec{x}\right)$$

This is best done in two parts. First consider the a = b case, and choose a = b = N WLOG. This can be derived very similarly to $I_1^{(N)}$, except in the last integral we'll need to use I_2. The a ≠ b case is a little harder. Let a = N and b = N-1 for concreteness. Then perform the first N - 2 integrals. The (N-1)st integral follows easily using I_1 above. However, then we are left with an integral over x_N involving x_N times two different Normal PDFs. Here we perform integration-by-parts to get the final result. Combining with the a = b case, we find

$$I_{ab}^{(N)} := I(\gamma)\delta_{ab} - \frac{AB_aB_b}{Q^3}\phi(\gamma)$$

RP Mean

The integral for the mean RP is given by

$$P_1 = \int d^N\theta\Phi\left(\mathbf{w}^T\boldsymbol{\theta}\right)\rho(\boldsymbol{\theta})$$

$$\text{where } \rho(\boldsymbol{\theta}) := \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})$$

$$= \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|}\exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})\right)$$

This seems like a challenging integral to perform, but as we'll see a judicious change of variables will save the day. Before moving on, we note a sub-optimal brute force way of doing this integral. We *could* simply use the formula for the conditional of a MVN, which would allow us to perform each integral, one-by-one. This is tractable for N = 2 and N = 3 (though quite tedious in this case), but gives no suggestion about how to prove the result for all N. The change-of-variables approach below is much simpler.

Recall that the our symmetric, positive definite covariance matrix has a "square root", namely its Cholesky decomposition:

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$$

This suggests a variable change that converts the correlated N-dimensional Gaussian into an uncorrelated product of N Gaussians:

$$\mathbf{x} = \mathbf{L}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})$$

The Jacobian of this transformation is simply

$$|J| = |\mathbf{L}| = |\boldsymbol{\Sigma}|^{1/2}$$

Thus we obtain

$$P_1 = \int d^N x \Phi\left[\mathbf{w}^T(\boldsymbol{\mu} + \mathbf{L}\mathbf{x})\right] \phi(\mathbf{x})$$

This is exactly of the form of $\mathtt{I}_0^{(\mathrm{N})}$ given above, allowing us to write down the answer given in the main text:

$$P_1 = \Phi(\gamma)$$
$$\text{where } \gamma = \frac{\mathbf{w}^T\boldsymbol{\mu}}{Q}$$
$$Q = \sqrt{1 + \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}}$$

Proficiency Means

With the insight gained doing the P_1 integral, computing the proficiency means is pretty straightforward:

$$\begin{aligned}\boldsymbol{\mu}^{(r)} &:= \langle\boldsymbol{\theta}|r\rangle := \frac{1}{P_r}\int d^N\theta\Phi\left[(-1)^{r-1}\mathbf{w}^T\boldsymbol{\theta}\right]\rho(\boldsymbol{\theta})\boldsymbol{\theta} \\ &= \frac{1}{P_r}\int d^N x\Phi\left[(-1)^{r-1}\mathbf{w}^T(\boldsymbol{\mu}+\mathbf{L}\mathbf{x})\right]\phi(\mathbf{x})(\boldsymbol{\mu}+\mathbf{L}\mathbf{x}) \\ &= \boldsymbol{\mu} + \frac{\mathbf{L}}{P_r}\int d^N x\Phi\left[(-1)^{r-1}\mathbf{w}^T(\boldsymbol{\mu}+\mathbf{L}\mathbf{x})\right]\phi(\mathbf{x})\mathbf{x}\end{aligned}$$

The integral on the last line is just $\mathtt{I}_1^{(\mathrm{N})}$ derived above. Thus we obtain the result

$$\boldsymbol{\mu}^{(r)} = \boldsymbol{\mu} + \frac{\boldsymbol{\Sigma}\mathbf{w}}{Q}R_r(\gamma)$$

Proficiency Variances

With the previous two sections under our belts, we have all of the tricks needed to perform the covariance integrals:

$$\begin{aligned}\boldsymbol{\Sigma}^{(r)} &:= \langle(\boldsymbol{\theta}-\boldsymbol{\mu}^{(r)})(\boldsymbol{\theta}-\boldsymbol{\mu}^{(r)})^T|r\rangle \\ &:= \frac{1}{P_r}\int d^N\theta\Phi\left[(-1)^{r-1}\mathbf{w}^T\boldsymbol{\theta}\right]\rho(\boldsymbol{\theta})(\boldsymbol{\theta}-\boldsymbol{\mu}^{(r)})(\boldsymbol{\theta}-\boldsymbol{\mu}^{(r)})^T \\ &= \frac{1}{P_r}\int d^N x\Phi\left[(-1)^{r-1}\mathbf{w}^T(\boldsymbol{\mu}+\mathbf{L}\mathbf{x})\right]\phi(\mathbf{x})(\mathbf{L}\mathbf{x}-\boldsymbol{\delta})(\mathbf{L}\mathbf{x}-\boldsymbol{\delta})^T\end{aligned}$$
$$\text{where } \boldsymbol{\delta} := \boldsymbol{\mu}^{(r)} - \boldsymbol{\mu}$$

This is then just a combination of the three new integrals discussed above: $\mathtt{I}_0^{(\mathrm{N})}$, $\mathtt{I}_1^{(\mathrm{N})}$, and $\mathtt{I}_{ab}^{(\mathrm{N})}$. Performing the somewhat lengthy algebra yields:

$$\boldsymbol{\Sigma}^{(r)} = \boldsymbol{\Sigma} + \frac{(\boldsymbol{\Sigma}\mathbf{w})(\boldsymbol{\Sigma}\mathbf{w})^T}{Q^2}R_r(\gamma)\left[\gamma + R_r(\gamma)\right]$$

This vectorized form can also be written in terms of components:

$$\Sigma_{ab}^{(r)} = \Sigma_{ab} + \frac{(\boldsymbol{\Sigma}\mathbf{w})_a(\boldsymbol{\Sigma}\mathbf{w})_b}{Q^2} R_r(\gamma)\left[\gamma + R_r(\gamma)\right]$$

Appendix 2: Score (Co)-Variances

Let R_i be the response correctness random variable for some item i with weight vector $\mathbf{w}_i$. By construction, in DIRT this is just $R_i = \Phi(\mathbf{w}_i^T\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the proficiency vector random variable. The score covariance between two items is defined by

$$\begin{aligned}\operatorname{cov}(R_1, R_2) &= \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\left(\Phi(\mathbf{w}_1^T\boldsymbol{\theta}) - \mu_{R_1}\right)\left(\Phi(\mathbf{w}_2^T\boldsymbol{\theta}) - \mu_{R_2}\right) \\ &= -\mu_{R_1}\mu_{R_2} + \int d^N\theta \phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\Phi(\mathbf{w}_1^T\boldsymbol{\theta})\Phi(\mathbf{w}_2^T\boldsymbol{\theta}) \\ \text{where } \mu_{R_i} &:= \mathbb{E}\left[R_i\right] = \Phi(\gamma_i) = \Phi\left(\frac{\mathbf{w}_i^T\boldsymbol{\mu}}{\sqrt{1 + \mathbf{w}_i^T\boldsymbol{\Sigma}\mathbf{w}_i}}\right)\end{aligned}$$

The N-dimensional PDF x CDF x CDF is challenging. Previously for score variances, we've encountered this type of integral for N = 1 and used the Owen's T function approximation. This works well enough in that case, though it's rather complicated and difficult to intuit about. Including the effects of the N-dimensional covariance matrix is probably feasible[5], but not implemented and somewhat complicated.

Happily, there is a very simple RNA approach to this score covariance, that requires no new numerical approximations like Owen's T, and automatically uses the full N-dimensional proficiency distribution, not an artificial independent Normal approximation. Recall the heart of the RNA framework is moment matching a Gaussian PDF times CDF with Gaussian PDF. With the normalization factor included:

$$\phi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\theta})\Phi(\mathbf{w}^T\boldsymbol{\theta}) \overset{\text{RNA}}{\approx} \Phi(\gamma)\phi_{\boldsymbol{\mu}^{(r=1)},\boldsymbol{\Sigma}^{(r=1)}}(\boldsymbol{\theta})$$

Quantities with (r = 1) superscript are just the values after applying a correct response, given in the main text here. This approximation performs surprisingly well over the range of parameters and use cases we have. This allows us to compute the score covariance in closed form with no other approximations:

$$\operatorname{cov}(R_1, R_2) = \mu_{R_1}\left(\mu_{R_2}^{(r_1=1)} - \mu_{R_2}\right)$$

Thus the covariance is computed in terms of 3 quantities: the two score means now, and then the score mean of item 2 after a would-be correct on item 1. Note that we could just as well swap items 1 and 2 here. Indeed, the result is numerically the same for all practical purposes, usually equal to 3 or 4 decimal places.

[5] Unpublished note, Chaitu E (2015)

The above results straightforwardly yield the variance of score S that is the sum (here normalized) of individual items, as would be of interest in predicting the score on an exam, for example:

$$\mathrm{var}(S) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{cov}(R_i, R_j)$$

$$\text{where } S := \frac{1}{N} \sum_{i=1}^{N} R_i$$

Finally, we note that the above form for the score covariances points towards a resolution to the outstanding issue wherein the old score variance endpoints get “too confident too quick” for multi-concept scores, because of the artificial independence assumption. The resolution involves acknowledging the importance of correlations between different items in the graph, which in this case mean implementing suitable shared “god concepts” at different granularity levels. The LO in our case serves this purpose. One could imagine jointly learning proficiency on the topic or even chapter level as well. The apportionment of weights between these entities is precisely the knob that dials between no correlation (except via propagation) and strong correlation, and could be fit to data (or at least reasonable levels).