GEOFF'S THESIS

by

Geoffrey Converse

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and Computational Sciences
in the Graduate College of
The University of Iowa

Date?

Thesis Committee: Professor Suely Oliveira, Thesis Supervisor
Member Two
Member Three
Member Four
Member Five

ACKNOWLEDGEMENTS

ABSTRACT

PUBLIC ABSTRACT

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
BACKGROUND - IRT PARAMETER ESTIMATION

In educational measurement, a common goal is to quantify the knowledge of students from the results of some assessment. In a classroom setting, grades are typically assigned based on the percentage of questions answered correctly by a student assignments. The letter grades assigned from these percentages can serve as a naive measure of student knowledge; "A" students have completely mastered the material, "B" students have a good grasp of material, "C" students are fairly average, and "D" and "F" students have significant gaps in their knowledge.

The practice of evaluating student ability purely from a raw percentage score is known as true score theory [9]. But there are clear issues with this approach. Not all questions on an exam or homework assignment is created equally: some questions are easier, and some more difficult. Consider a scenario where two students both answer 17 out of 20 questions correctly on a test for a raw score of 85%. But if Student A answered questions 1, 8, and 9 wrong while Student B answered 4, 17, and 20 incorrectly, it is not likely that that Student A and Student B possess the same level of knowledge. For example, questions 1, 8, and 9 could be much more difficult than questions 4, 17, and 20. Additionally, the two sets of problems could cover different types of material. True score theory does not account for either of these situations, and naively quantifies the knowledge of Student A and Student B as equal.

More sophisticated methods have been studied which attempt to more accurately quantify student learning. Cognitive Diagnostic Models (CDM) (TODO:

citation) aim to classify whether students possess mastery of a given skill or not. This discrete classification can be useful in determining whether or not a student meets a prerequisite, or deciding whether or not they are ready to move on to the next level of coursework. We focus instead on Item Response Theory, where student knowledge is assumed to be continuous.

1.1 Item Response Theory

Item Response Theory (IRT) is a field of quantitative psychology which uses statistical models to model student ability [6]. These models often give the probability of a question being answered correctly as a function of the student's ability. In IRT, it is assumed that each student, indexed by j, possesses some continuous latent ability θ_j. The term "latent ability" is synonymous with "knowledge"or "skill." Often, it is assumed that amongst the population of students, $\theta_j \sim \mathcal{N}(0, 1)$ [9].

In this work, we often consider the case where each student has multiple latent abilities. For example, in the context of an elementary math exam, we may wish to measure the four distinct skills "add", "subtract", "multiply", and "divide." This scenario is referred to as multidimensional item response theory, and we write the set of student j's K latent abilities as a vector $\Theta_j = (\theta_{1j}, \theta_{2j}, \ldots, \theta_{Kj})^\top$. It is then assumed that the latent abilities of students follow some multivariate Gaussian distirbution, $\mathcal{N}(0, \Sigma)$. For simplicity, the covariance matrix Σ is often taken to be the identity matrix, making each latent skill independent of one another.

Note that Θ_j is not directly observable in any way. Instead, a common goal is

to infer student's knowledge Θ_j from on their responses on some assessment containing n questions, referred to as items. A student's set of responses can be written as a binary n-dimensional vector $\vec{u}_j = (u_{1j}, u_{2j}, \ldots, u_{nj})^\top$, where

$$u_{ij} = \begin{cases} 1 & \text{if student } j \text{ answers item } i \text{ correctly} \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

IRT models aim to model the probability of a student answering a particular question correctly, so that the probability of student j answering item i correctly is given by some function of Θ_j:

$$P(u_{ij} = 1|\Theta_j) = f(\Theta_j; V_i) \tag{1.2}$$

where V_i is a set of parameters associated with item i. In general, $f : \mathbb{R}^K \to [0, 1]$ is some continuous function which is strictly increasing with respect to Θ_j.

In the following sections, we describe various candidates for the function f. Though each is presented in the context of single-dimensional IRT ($K = 1$), they can all be easily adapted to higher dimensions.

1.1.1 Rasch Model

One of the first models was proposed by Georg Rasch in 1960. Rasch asserted that the probability of a student answering an item correctly is a function of the ratio ξ/δ, where $\xi > 0$ represents the student's knowledge, and $\delta > 0$ quantifies the difficulty of an item. Consider the $\frac{\xi}{\xi+\delta} = \frac{1}{1+\delta/\xi}$ and note that $\frac{\xi}{\xi+\delta} \to 1$ as $\xi \to \infty$. After the reparametarization $\xi = e^\theta$ and $\delta = e^b$, we arrive at the 1-Parameter Logistic

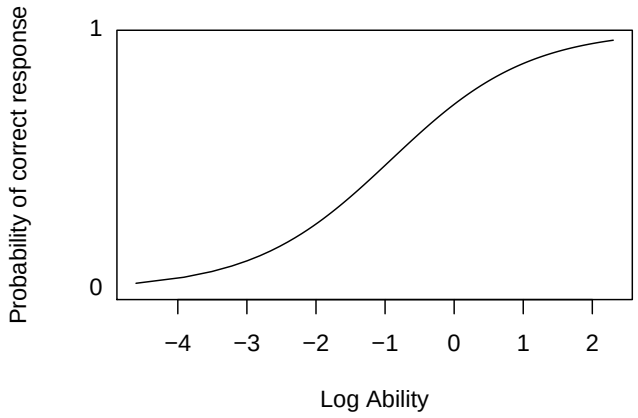

Figure 1.1: An item characteristic curve visualizes the relation between a student's ability and the probability of answering an item correctly.

Model, often referred to as the Rasch Model.

$$P(u_{ij} = 1|\theta_j; b_i) = \frac{1}{1 + e^{b_i - \theta_j}} \tag{1.3}$$

Note that $\theta \in \mathbb{R}$ and $b \in \mathbb{R}$ still represent student ability and item difficulty, respectively. We can interpret the difficulty parameter b as a threshold: when $\theta = b$, then the student has a 50% chance of answering the question correctly. A plot of Equation 1.3 for a fixed item (fixed b_i) is shown in Figure 1.1. The horizontal axis represents $\log \theta$, and the vertical axis represents $P(u_{ij} = 1|\theta, b_i)$. This type of graph is often referred to as an item characteristic curve (ICC).

1.1.2 Normal Ogive Model

A slightly more sophisticated method for measuring student performance is the normal ogive model. We introduce a discrimination parameter, a_i, which quantifies the capability of item i in distinguishing between students who have / have not mastered the knowledge concept θ [9]. In other words, a_i tells *how much* of skill θ is required to answer item i correctly.

The normal ogive model give the probability of student j answering item i correctly as

$$P(u_{ij} = 1|\theta_j; a_i, b_i) = \frac{1}{\sqrt{2\pi}} \int_{-a_i\theta_j + b_i}^{\infty} e^{\frac{-z^2}{2}} dz \tag{1.4}$$

Note the similarity between Equation 1.4 and the cumulative distribution function for a Gaussian distribution. The normal ogive model is popular among statisticians for this reason, but can be difficult to use for parameter estimation.

1.1.3 2-Parameter Logistic Model

The model which this work focuses on most is the 2-parameter logistic (2PL) model. Like the normal ogive model, the 2PL model uses both the discrimination and difficulty item parameters. The probability of student j answering item i correctly is given by

$$P(u_{ij} = 1|\theta_j; a_i, b_i) = \frac{1}{1 + e^{-a_i\theta_j + b_i}} \tag{1.5}$$

Equation 1.5 has the same form as that of the Rasch model in Equation 1.3, but adds in the discrimination parameter a_i. If this parameter is scaled by 1.7, then the ICC from the normal ogive model differs from that of the 2PL model by 0.01 [1].

In a sense, we can consider the 2PL model to be a very good approximation of the normal ogive model. Due to the simple form of Equation 1.5, using this model makes parameter estimation much easier.

1.1.4 Multidimensional Item Response Theory

The previously described statistical models can all be extended so that each student possesses K latent traits. In multidimensional item response theory (MIRT), models give the probability of a correct answer as a function of the student ability vector $\Theta = (\theta_1, \ldots, \theta_K)^\top$. The generalization of 1.5 is given by the multidimensional logistic 2-parameter (ML2P) model:

$$P(u_{ij} = 1 | \Theta_j; \vec{a_i}, b_i) = \frac{1}{1 + \exp\left(-\vec{a_i}^\top \Theta_j + b_i\right)} = \frac{1}{\exp\left(-\sum_{k=1}^{K} a_{ik}\theta_{kj} + b_i\right)} \tag{1.6}$$

Here, the discrimination parameters $\vec{a_i} \in \mathbb{R}^K$ are given as vector, where each entry $a_{ik} \in \vec{a_i}$ quantifies *how much* of skill k is required to answer item i correctly. The ML2P model is the main focus of this thesis.

TODO: mention MDISC and how this scales

In MIRT, it is convenient to notate the relationship between skills and items with binary matrix. Define the Q-matrix [3] $Q \in \{0, 1\}^{n \times K}$ so that

$$q_{ik} = \begin{cases} 1 & \text{if item } i \text{ requires skill } k \\ 0 & \text{otherwise} \end{cases}. \tag{1.7}$$

In real applications, the Q-matrix is annotated by an expert in the field, as it is usually possible to discern the concepts need to answer an item correctly. In relation to the ML2P model (Equation 1.6), notice that if $q_{ik} = 0$, then $a_{ik} = 0$ as well. Though

experts can produce a Q-matrix for a given assessment, the matrix of discrimination parameters $(a_{ik})_{i,k}$ can not be discovered so easily.

1.2 IRT Parameter Estimation

1.2.1 Maximum Likelihood Estimation

TODO: item parameter estimation

TODO: ability parameter estimation

1.2.2 Joint Maximum Likelihood Estimation

1.2.3 Marginal Maximum Likelihood Estimation

TODO: MMLE

TODO: EM

1.3 Artificial Neural Networks

In recent years, artifical neural networks (ANN) have become an increasingly popular tool for machine learning problems. Though they have been around since the 1960's (TODO: citation), GPU technology has become more accessible and modern computers are more powerful, allowing anyone interested to train a basic neural network on their machine. ANN can be applied to a diverse set of problems, including regression, classification, computer vision, natural language processing, function approximation, data generation, and more (TODO: citations).

One of the biggest critiques of ANN is their black-box nature, meaning that the decision process that a trained model uses is typically not explainable by humans. As

opposed to simpler methods such as decision trees or linear regression, neural networks are not interpretable. This makes them less desirable in certain applications where researchers wish to know *why* a model predicts a particular data sample the way that it does. For example, if a financial institution is using data science methods to determine whether or not to approve someone's loan, the institution should be able to explain to the customer why they were denied. Most customers will not be satisfied with "the computer told us so," and there is a possibility that a black-box neural network could learn and use features such as race or gender in its prediction, which is illegal in the United States (TODO: definitely need citation or delete).

The push for explainable AI have led researchers down two paths. One group has tried to incorporate deep learning methods with existing interpretable methods, in hopes of increasing the performance of explainable methods without sacrificing its interpretability (TODO: citation). Another option is to use a sort of hybrid learning, where interpretable models defer to a black-box model if they are not confident in their prediction [8]. Others have started with deep models and cut back on complexity, making specific modifications which increase interpretability. For example, the loss function of a convolutional neural network can be adapted so that humans can understand the features extracted in the hidden layers [11].

The field of education is an application which often desires interpretable models. Researchers often need to be able to point out specific details of decisions made by AI. A student deserves an answer to *why* they failed a test, and a teacher should be given instructions on *how* to fix the student's misconceptions.

1.3.1 Autoencoders

An autoencoder (AE) is a neural network where the input and output layers are the same shape. The objective for a given data point is to minimize the difference between the output, called the reconstruction, and the input. Typically, the middle hidden layers of an AE are of smaller dimension than the input space. In this way, autoencoders are an unsupervised learning technique for (nonlinear) dimension reduction. Mathematically, we can define an autoencoder in two parts as follows.

For an input $x \in \mathbb{R}^n$, define the *encoder* as a function $f : \mathbb{R}^n \to \mathbb{R}^m$ mapping $x \mapsto z := f(x)$. Usually, $m < n$, and z lies in a hidden feature space. The encoder sends an observed data point to its representation in a learned feature space. Define the *decoder* as a function $g : \mathbb{R}^m \to \mathbb{R}^n$ mapping $z \mapsto \hat{x} := g(z)$. The decoder maps a hidden representation z to a reconstruction of the encoder input. Note that in our case, the functions f and g are both parameterized by neural networks, each of which can have any number of hidden layers. The end-to-end autoencoder is then the function composition $\mathcal{A}(x) := g(f(x)) : \mathbb{R}^n \to \mathbb{R}^n$. To train an AE, the loss function minimizes the difference between the input and output. This can be done in a number of ways, including the simple mean squared error loss

$$\mathcal{L}(x) = ||x - g(f(x))||_2^2 \tag{1.8}$$

or cross-entropy loss for binary data

$$\mathcal{L}(x) = \sum_{i=1}^{n} -x_i \log(g(f(x_i))) - (1 - x_i) \log(1 - g(f(x_i))). \tag{1.9}$$

Autoencoders with only a single hidden layer can be compared with nonlinear

principal components analysis (PCA), and using linear activation functions allows for recovery of PCA loading vectors [7]. AEs have clear applications in image compression straight out-of-the-box, and can be modified for more complicated problems. Denoising autoencoders [10] are capable of processing noisy images and cleaning them up. To do this, they corrupt input data by deleting pixels at random and reconstructing the original image. Autoencoders can also be modified for data generation applications using a variational autoencoder.

1.3.2 Variational Autoencoders

*relevant sources: [4] [5] [2], infoVAE, ELBO, "towards deeper understanding of VAE"

TODO: describe probabilistic derivation of VAE (ie Kingma and Welling). Also talk about how Zhao et al (InfoVAE) show that if decoder is Gaussian, then maximizing ELBO makes the latent distribution bad - bu I've shown this isn't the case in our model, where the decoder is Bernoulli.

CHAPTER 2
METHODS - IRT PARAMETER ESTIMATION

The primary contribution of this thesis is the development of a method for IRT parameter estimation which uses a modified VAE. The method, titled "ML2P-VAE", is interesting from multiple perspectives. In the application area, it is an unconventional approach can produce estimates as accurate as those of traditional parameter estimation techniques. Further, ML2P-VAE scales much better than traditional methods as the number of latent abilities becomes large. In the field of machine learning, ML2P-VAE is an unsupervised learning technique which yields explainable results. A variational autoencoder is used in an unorthodox way, and the trainable parameters and a hidden neural layer are able to be understood in a real-world context.

2.1 ML2P-VAE Method Description

Assume we are given an assessment with n items which tests K latent skills, and that N students take this exam. Data is given as an $N \times n$ binary matrix, as in Equation 1.1. No information about student latent ability parameters $\Theta \in \mathbb{R}^K$ or item parameters a_{ik} and b_i is provided. However, there is access to an expert-annotated binary Q-matrix detailing the item-skill associations, as in Equation 1.7.

A number of specifications are required in order to use a VAE as an IRT parameter estimation method. First, set up the neural network so that the input and output layers each have n nodes, each representing one item. The inputs are be the binary response vectors $\vec{u}_j$ and the outputs are approximations of the probabil-

ity of student j answering each item correctly. Next, the dimension of the hidden distribution (the output of the encoder) must be equal to K, the number of latent skills.

The modifications to a typical VAE architecture are focused on the decoder. No hidden layers are used in the decoder. Instead, a non-dense set of weights connects the decoder input to the decoder output. The non-zero weights here are determined by the Q-matrix; recall that the input to the decoder has K nodes, the decoder output hase n nodes, and $Q \in \{0,1\}^{n \times K}$. Further, require the use of the sigmoidal activation function

$$\sigma(z_i) = \frac{1}{1 + e^{z_i}} \tag{2.1}$$

in the output layer. Here, $z_i = \sum_{k=1^K} w_{ik}\alpha_k + \beta_i$, where w_{ik} is the weight between the k-th and i-th nodes in the decoder input and decoder output layer, α_k is the activation of the k-th node in the decoder input layer, and β_i is the additive bias in the output layer. Note the similarity between Equation 2.1 and Equation 1.6. The constraint on the weights along with the sigmoidal activation function allows for interpretation of the decoder as an ML2P model.

Specifically, the decoder weights w_{ik} can be interpreted as estimates to the discrimination parameters a_{ik}, the output bias β_i can be interpreted as estimates to the difficulty parameters b_i, and the activations α_k produced by the encoder (given response input $\vec{u}_j$) can be interpreted as estimates to the student ability parameter θ_{kj}.

may want to mention that the value outputed by encoder is then sampled from for dceoder input

Further modifications can improve the performance of ML2P-VAE. In IRT,

discrimination parameters are assumed to be non-negative, because an increase in a skill should never decrease the probability of answering an item correctly. With this assumption in mind, requiring all decoder weights $w_{ik} \geq 0$ avoids a potential identification problem.

explain what this means $-\theta a_{ik} = \theta(-a_{ik})$

2.1.1 Full Covariance Matrix Implementation

2.1.2 Variants - 1PL and 3PL

2.1.3 On Convergence

Can't really guarantee convergence globally with SGD because neural nets are super nonconvex

2.1.3.1 Proving local minimum at true solution

We define the true and predicted probability of student j answering item i correctly with P_{ij} and $\hat{P}_{ij}$, respectively. The former comes from the ML2P model, and the latter is the output of a neural network. We assume the more simple case, where $\Theta \sim \mathcal{N}(0, I)$, rather than $\mathcal{N}(\mu, \Sigma)$.

$$\begin{aligned} P_{ij} &= \frac{1}{1 + \exp\left(-\sum_{k=1}^{K} a_{ik}\theta_{jk} + b_i\right)} \\ \hat{P}_{ij} &= \frac{1}{1 + \exp\left(-\sum_{k=1}^{K} \hat{a}_{ik}(\hat{\theta}_{jk} + \varepsilon_k\hat{\sigma}_k) + \hat{b}_i\right)} \end{aligned} \tag{2.2}$$

Note that P_{ij} is unknown, and we instead have a response sequence $\vec{u}_j = (u_{1j}, \ldots, u_{nj})^\top$ with $u_{ij} = \text{Bern}(P_{ij})$. This also means that $\mathbb{E}[u_{ij}] = P_{ij}$. The variables $\hat{a}_{ik}$, $\hat{b}_i$, $\hat{\theta}_{ik}$, and are parameter estimates from the VAE. The first two are parameters in the neural network, and the ability estimates are taken from feeding responses to the encoder, i.e., $\hat{\Theta}_j = \text{Encoder}(\vec{u}_j)$. The noise $\varepsilon = (\varepsilon_k)_{1 \leq k \leq K} \sim \mathcal{N}(0, I)$ is introduced by the sampling operation in the VAE.

The loss function for a VAE is given by

$$\begin{aligned}\mathcal{L}(\vec{u}_j) &= -\sum_{i=1}^{n}\Big[u_{ij}\log(\hat{P}_{ij}) + (1-u_{ij})\log(1-\hat{P}_{ij})\Big] + \mathbb{E}_{q_\alpha(\hat{\theta}|\vec{u}_j)}\log\left(\frac{q_\alpha(\hat{\theta}|\vec{u}_j)}{p(\theta)}\right) \\ &= \mathcal{L}_{\text{REC}} + \mathcal{L}_{\text{KL}}\end{aligned} \tag{2.3}$$

We break up the VAE loss into two terms, the reconstruction loss $\mathcal{L}_{\text{REC}}$ and the KL-divergence loss $\mathcal{L}_{\text{KL}}$. in the latter, the distribution $q_\alpha(\hat{\theta}|\vec{u}_j)$ is the output of the encoder, and $p(\theta)$ is the assumed prior distribution of Θ, which we set to be $\mathcal{N}(0, I)$.

We write $P_{ij} = \mathbb{E}(u_{ij})$, and similarly define a new "expected" loss function:

$$\begin{aligned}\mathcal{L}_{\mathbb{E}}(P_j) &= \mathbb{E}_{u_j}[\mathcal{L}(u_{:j})] \\ &= -\sum_{i=1}^{n}(P_{ij}\log(\hat{P}_{ij}) + (1-P_{ij})\log(1-\hat{P}_{ij}) + \mathbb{E}_{q_\alpha(\hat{\theta}|u_j)}\log\left(\frac{q_\alpha(\hat{\theta}|u_j)}{p(\theta)}\right) \\ &= \mathcal{L}_{\mathbb{E}[\text{REC}]} + \mathcal{L}_{\mathbb{E}[\text{KL}]}\end{aligned} \tag{2.4}$$

Notice that calculation of this "expected loss" requires the unknown $P_{:j}$. But when we have large amounts of data, we can think of P_{ij} as the average value of the response u_{ij}, so using this unknown value here is justified.

Define $z_i = a_{i:} \cdot \theta_{j:} - b_i$ and $\hat{z}_i = \hat{a}_{i:} \cdot (\hat{\theta}_{j:} + \varepsilon \cdot \hat{\sigma}) - \hat{b}_i$. Note that z_i is fixed, dependent on the data, and does not depend on any parameters of the neural network. $\hat{z}_{:}$ is the input to the final layer of the decoder, and the VAE output is $\hat{P}_{ij} = \sigma(\hat{z}_i)$, where $\sigma(\cdot)$ is the sigmoidal activation function. We compute derivatives of the expected loss function, looking individually at the reconstruction and KL terms.

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i} &= \frac{-1}{1+e^{-z_i}} \cdot \frac{1}{1+e^{\hat{z}_i}} - \frac{1}{1+e^{z_i}} \cdot \frac{-1}{1+e^{-\hat{z}_i}} \\ &= \frac{-1}{(1+e^{-z_i})(1+e^{\hat{z}_i})} + \frac{1}{(1+e^{z_i})(1+e^{-\hat{z}_i})} \\ &= \frac{-1}{(1+e^{-a_{ik}\theta_{jk}+b_i})(1+e^{\hat{a}_{ik}(\hat{\theta}_{kj}+\varepsilon_k\hat{\sigma}_k)-\hat{b}_i})} \\ &\quad + \frac{1}{(1+e^{a_{ik}\theta_{jk}-b_i})(1+e^{-\hat{a}_{ik}(\hat{\theta}_{jk}+\varepsilon_k\hat{\sigma}_k)+\hat{b}_i})}\end{aligned} \tag{2.5}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{a}_{ik}} &= \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}\frac{\partial \hat{z}_i}{\partial \hat{a}_{ik}} = \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}(\hat{\theta}_{jk}+\varepsilon_k\hat{\sigma}_k) \\ \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{b}_i} &= \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}\frac{\partial \hat{z}_i}{\partial \hat{b}_i} = \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}(-1) \\ \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{\theta}_{ik}} &= \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}\frac{\partial \hat{z}_i}{\partial \hat{\theta}_{ik}} = \frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}(\hat{a}_{ik})\end{aligned} \tag{2.6}$$

Rather than setting these to zero and solving, we show that the most intuitive solution, $\hat{a}_{ik} = a_{ik}$, $\hat{b}_i = b_i$, and $\hat{\theta}_{jk} = \theta_{jk}$, is in fact a minimum of the expected loss function. But first, we must take another expectation over the random variable $\varepsilon \sim \mathcal{N}(0, I)$. Obviously, we have that $\mathbb{E}[\varepsilon_k] = 0$; this makes our calculations very simple. Notice that we have

$$\begin{aligned}&\mathbb{E}_\varepsilon\left[\frac{\partial \mathcal{L}_{\mathbb{E}[\text{REC}]}}{\partial \hat{z}_i}\right]\Big|_{\hat{a}_{ik}=a_{ik},\hat{b}_i=b_i,\hat{\theta}_{jk}=\theta_{jk}} \\ &= \frac{-1}{(1+e^{-a_{ik}\theta_{jk}+b_i})(1+e^{a_{ik}(\theta_{kj}+0\hat{\sigma}_k)-b_i})} + \frac{1}{(1+e^{a_{ik}\theta_{jk}-b_i})(1+e^{-a_{ik}(\theta_{jk}+0\hat{\sigma}_k)+b_i})} \\ &= 0\end{aligned} \tag{2.7}$$

Therefore we clearly have

$$\begin{aligned}
&\mathbb{E}_{\varepsilon}\left[\frac{\partial \mathcal{L}_{\mathbb{E}[\mathrm{REC}]}}{\partial \hat{a}_{ik}}\right]\Big|_{\hat{a}_{ik}=a_{ik},\hat{b}_i=b_i,\hat{\theta}_{jk}=\theta_{jk}} \\
=&\mathbb{E}_{\varepsilon}\left[\frac{\partial \mathcal{L}_{\mathbb{E}[\mathrm{REC}]}}{\partial \hat{b}_{i}}\right]\Big|_{\hat{a}_{ik}=a_{ik},\hat{b}_i=b_i,\hat{\theta}_{jk}=\theta_{jk}} \\
=&\mathbb{E}_{\varepsilon}\left[\frac{\partial \mathcal{L}_{\mathbb{E}[\mathrm{REC}]}}{\partial \hat{\theta}_{jk}}\right]\Big|_{\hat{a}_{ik}=a_{ik},\hat{b}_i=b_i,\hat{\theta}_{jk}=\theta_{jk}} \\
=&0 \quad \forall i,j,k
\end{aligned} \tag{2.8}$$

This proves that the true parameters give a local minimum for the expected reconstruction error in the VAE loss. And because the expected cross-entropy loss function $\mathcal{L}_{\mathbb{E}[\mathrm{REC}]}$ is non-negative, the reconstruction error at the true IRT paramters is a global minimum.

any chance this is unique?

We now consider the Kullback-Leibler divergence term in the expected loss function. Again assuming independent latent traits, we have

$$\mathcal{L}_{KL} = \mathbb{E}_{q(\theta|u)}[\log\left(\frac{q(\hat{\theta}|u)}{p(\theta)}\right) = KL(q(\hat{\theta}|u)||p(\theta)) = -\frac{1}{2}\sum_{k=1}^{K}(1+\log(\hat{\sigma}_k^2)-\hat{\theta}_k^2-\hat{\sigma}_k^2) \tag{2.9}$$

It is clear that this regularization term is minimized (and equal to zero) when $\hat{\theta}_{jk} = 0$ and $\hat{\sigma}_{jk} = 1$. But what happens when we plug in the "true" student ability values as before? We have

$$\mathcal{L}_{KL}\Big|_{\hat{\theta}=\theta,\hat{\sigma}=\sigma} = KL(p(\theta|u)||p(\theta)) \tag{2.10}$$

Notice that this is the KL divergence between the **true posterior** $p(\theta|u)$ and the **true prior** $p(\theta)$. This is interpreted as the average difference of number of bits required to encode samples of $p(\theta|u)$ using a code optimized for $p(\theta)$, rather than one optimized for $p(\theta|u)$.

We should be okay with accepting this loss, since the true posterior is not actually known, and we are just using the prior as a reference.

2.1.3.2 Requirements on sparsity of Q-matrix

need to look into this idea at some point

TODO: try to show that this is a global minimum for the full VAE loss function. Also take derivatives of the KL loss w.r.t $\hat{\theta}_{jk}$. The Q-matrix may help with an identifiability issue (existence of other local minimums) in solving the system $(a_{ik}\theta_{jk} + b_i)_{jk} = z_i$. The Q-matrix *may* make the solution unique.

2.2 ML2Pvae Software Package for R

Plug that I made this and it is publicly available - hopefully on CRAN.

2.2.1 Package Functionality

CHAPTER 3
RESULTS - IRT PARAMETER ESTIMATION

3.1 Description of Data Sets

3.2 1-PL Results

3.3 2-PL Results

3.4 3-PL Results

(maybe)

CHAPTER 4
RELATED WORK

How this type of technology can be used in other fields.

4.1 Sports Analytics Application

4.2 Health Sciences Application

CHAPTER 5
KNOWLEDGE TRACING BACKGROUND

5.1 Application Goal

5.2 Mathematical Setup

5.3 Literature Review

5.3.1 Bayesian Knowledge Tracing

5.3.2 Deep Knowledge Tracing

5.3.3 Dynamic Key-Value Memory Networks

CHAPTER 6
KNOWLEDGE TRACING - METHODS

6.1 Item-based Attention Networks

TODO:name this better

CHAPTER 7
KNOWLEDGE TRACING - RESULTS

7.1 Data Description

Describe each dataset used here.

7.2 Experiment Details

Hyper parameters here

7.3 Results

REFERENCES

[1] F. Baker and S. Kim. *Item Response Theory Parameter Estimation Techniques*. Taylor & Francis Group, 2nd edition, 2004.

[2] D.M. Blei, A. Kucukelbir, and J.D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] M.A. da Silva, R. Liu, A.C. Huggins-Manley, and J.L. Bazan. Incorporating the q-matrix into multidimensional item response theory models. *Educational and Psychological Measurement*, 2018.

[4] Carl Doersch. Tutorial on variational autoencoders, 2016.

[5] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[6] F. Lord and M. R Novick. *Statistical theories of mental test scores*. IAP, 1968.

[7] Elad Plaut. From principal subspaces to principal components with linear autoencoders. *arXiv:1804.10253v2*, 2018.

[8] Hassan Rafique, Tong Wang, Quihang Lin, and Arshia Sighani. Transparency promotion with model-agnostic linear competitors. In *Proceedings of the International Conference on Machine Learning*, 2020.

[9] David Thissen and Howard Wainer. *Test Scoring*. Lawrence Erlbaum Associates Publishers, 2001.

[10] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008.

[11] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.