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
INTRODUCTION

This is the first chapter of the thesis. Here is how to cite a reference [7]. You can use \Cref{label} Chapter 1 to reference a chapter, section, theorem, etc. Also, \sideremark{text} allows you to make remarks in the margin for editing purposes.

Here is a comment in the margin

CHAPTER 2
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 [7]. 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.

2.1 Item Response Theory

Item Response Theory (IRT) is a field of quantitative psychology which uses statistical models to model student ability [4]. 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, $\theta_j \sim \mathcal{N}(0, 1)$ [7].

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 $U_j = (u_{1j}, u_{2j}, \dots, u_{nj})^\top$, where

$$u_{ij} = \begin{cases} 1 & \text{if student } j \text{ answers item } i \text{ correctly} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

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:

$$P(u_{ij} = 1 | \Theta_j) = f(\Theta_j; V_i) \tag{2.2}$$

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.

2.1.1 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

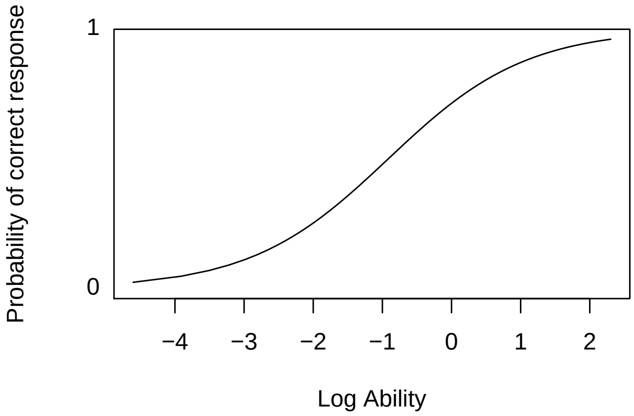

Figure 2.1: An item characteristic curve visualizes the relation between a student's ability and the probability of answering an item correctly.

Model, often referred to as the Rasch Model.

$$P(u_{ij} = 1|\theta_j; b_i) = \frac{1}{1 + e^{b_i - \theta_j}} \tag{2.3}$$

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 2.3 for a fixed item (fixed b_i) is shown in Figure 2.1. 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).

2.1.2 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 [7]. 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

$$P(u_{ij} = 1|\theta_j; a_i, b_i) = \frac{1}{\sqrt{2\pi}} \int_{-a_i\theta_j+b_i}^{\infty} e^{\frac{-z^2}{2}} dz \tag{2.4}$$

Note the similarity between Equation 2.4 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.

2.1.3 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

$$P(u_{ij} = 1|\theta_j; a_i, b_i) = \frac{1}{1 + e^{-a_i\theta_j+b_i}} \tag{2.5}$$

Equation 2.5 has the same form as that of the Rasch model in Equation 2.3, 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 [**?**].

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 2.5, using this model makes parameter estimation much easier.

2.1.4 Multidimensional Item Response Theory (MIRT)

The previously described statistical models can all be extended to multidimensional latent abilities $\Theta = (\theta_1, \ldots, \theta_K)^\top$. The generalization of 2.5 is given by the multidimensional logistic 2-parameter (ML2P) model:

$$Pju_{ij} = 1|\Theta_j; \vec{a_i}, b_i) = \frac{1}{1 + \exp\left(-\vec{a_i}^\top \Theta_j + b_i\right)} = \frac{1}{\exp\left(-\sum_{k=1}^{K} a_{ik}\theta_{kj} + b_i\right)} \tag{2.6}$$

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 (TODO: citation) $Q \in \mathbb{R}^{n \times K}$ so that

$$q_{ik} = \begin{cases} 1 & \text{if item } i \text{ requires skill } k \\ 0 & otherwise \end{cases}. \tag{2.7}$$

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 2.6), 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.

2.2 Parameter Estimation Methods

2.2.1 Maximum Likelihood Estimation

TODO: item parameter estimation

TODO: ability parameter estimation

2.2.2 Joint Maximum Likelihood Estimation

2.2.3 Marginal Maximum Likelihood Estimation

TODO: MMLE

TODO: EM

2.3 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 [6]. 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 [9].

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.

2.3.1 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

$$\mathcal{L}(x) = ||x - g(f(x))||_2^2 \tag{2.8}$$

or cross-entropy loss for binary data

$$\mathcal{L}(x) = \sum_{i=1}^{n} -x_i \log(g(f(x_i))) - (1 - x_i) \log(1 - g(f(x_i))). \tag{2.9}$$

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 [5]. AEs have clear applications in image compression straight out-of-the-box, and can be modified for more complicated problems. Denoising autoencoders [8] 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.

2.3.2 Variational Autoencoders

*relevant sources: [2] [3] [1], 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 3
METHODS - IRT PARAMETER ESTIMATION

3.1 ML2P-VAE Description

3.1.1 One-Parameter Logistic

3.1.2 2-Parameter Logistic

3.1.3 3-Parameter Logistic

tbd on this

3.1.4 Full Covariance Matrix Implementation

3.2 ML2Pvae Software Package for R

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

3.2.1 Package Functionality

CHAPTER 4
RESULTS - IRT PARAMETER ESTIMATION

4.1 Description of Data Sets

4.2 1-PL Results

4.3 2-PL Results

4.4 3-PL Results

(maybe)

CHAPTER 5
RELATED WORK

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

5.1 Sports Analytics Application

5.2 Health Sciences Application

CHAPTER 6
KNOWLEDGE TRACING BACKGROUND

6.1 Application Goal

6.2 Mathematical Setup

6.3 Literature Review

6.3.1 Bayesian Knowledge Tracing

6.3.2 Deep Knowledge Tracing

6.3.3 Dynamic Key-Value Memory Networks

CHAPTER 7
KNOWLEDGE TRACING - METHODS

7.1 Item-based Attention Networks

TODO:name this better

CHAPTER 8
KNOWLEDGE TRACING - RESULTS

8.1 Data Description

Describe each dataset used here.

8.2 Experiment Details

Hyper parameters here

8.3 Results

REFERENCES

[1] D.M. Bleia, A. Kucukelbirb, and J.D. McAuliffec. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

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

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

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

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

[6] 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.

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

[8] 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.

[9] 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.