Using Latent Variable Models to Observe Academic Pathways

Nate Gruver
Stanford University
ngruver@cs.stanford.edu

Ali Malik
Stanford University
malikali@cs.stanford.edu

Brahm Capoor
Stanford University
brahm@cs.stanford.edu

Chris Piech
Stanford University
piech@cs.stanford.edu

Mitchell L. Stevens
Stanford University
stevens4@stanford.edu

Andreas Paepcke
Stanford University
paepcke@cs.stanford.edu

ABSTRACT

Understanding large-scale patterns in student course enrollment is a problem of great interest to university administrators and educational researchers. Yet important decisions are often made without a good quantitative framework of the process underlying student choices. We propose a probabilistic approach to modelling course enrollment decisions, drawing inspiration from multilabel classification and mixture models. We use ten years of anonymized student transcripts from a large university to construct a Gaussian latent variable model that learns the joint distribution over course enrollments. The models allow for a diverse set of inference queries and robustness to data sparsity. We demonstrate the efficacy of this approach in comparison to others, including deep learning architectures, and demonstrate its ability to infer the underlying student interests that guide enrollment decisions.

1. INTRODUCTION

Education researchers increasingly recognize the need to understand the sequential accumulation of college coursework into academic pathways. In [2, 23], Bailey et al. call for change in how colleges organize course offerings to enable more efficient pathways. Rather than presenting a bewildering array of courses, cafeteria-style, they recommend "guided pathways" through academic offerings. Baker [3] builds on Bailey's work, suggesting "meta-majors" for simplifying choice without curtailment of options. Meta-majors entail combining coursework supporting multiple majors into larger, substantively coherent content domains. Baker proposes social-network analytic techniques to discover opportunities for building meta-majors. All of these authors argue that rather than limiting choice, such interventions can yield more tractable programs, faster degree completion, and lower cost for both students and schools.

Such reforms can be enabled by analysis of data corpora describing the academic sequences of prior student enrollments. For example, some courses may be de facto prerequisites for other courses, whether listed as "required" or not in formal catalogue entries. Similarly, "odd" delays in taking particular courses, or unexplained detours in course selection, can be symptoms of unintended scheduling conflicts.

In the service of such reforms, we offer a model of course enrollment capable of efficient inference over hundreds to thousands of classes. Our generative model captures the full joint distribution of course enrollments and can be used to sample potential pathways for any given student. The model's complexity allows us to determine an underlying "typography" of students, from implicit course-taking patterns to differing levels of novelty in their academic pathways relative to the overall population of paths.

2. BACKGROUND & MODELS

Predicting course enrollment decisions may be viewed as a problem of multi-label classification: the task of assigning a subset of labels to each data point in a collection. In context of academic course enrollments, each data point is a student and the labels are courses enrolled. The problem of modeling all possible enrollment choices scales exponentially with the number classes ($O(2^N)$), which motivates a statistical approach. Probabilistic graphical models (PGMs) and deep neural networks are perhaps the most prominent methods for stochastic models of high-dimensional data. As our motivation in this work is not simply high accuracy but also interpretability and inference, we focus on PGMs, which fare better on those aspects and are amenable to scaling adequate for our empirical setting.

2.1 Latent Variable Models

Latent variable models are a subclass of PGMs in which some variables are never observed in training data and are thus "latent." These models are more computationally demanding than fully observed models, but also are able to capture complex structure in data without supervision.

2.1.1 Models of Conditional Independence

Among the simplest and most commonly used latent variable models is the naive Bayes model with hidden variable H taking discrete values h_i and observations X. In the enrollment setting, $X = [x^0, ..., x^N]$ and $x^i = [x^i_0, ..., x^i_M]$ with

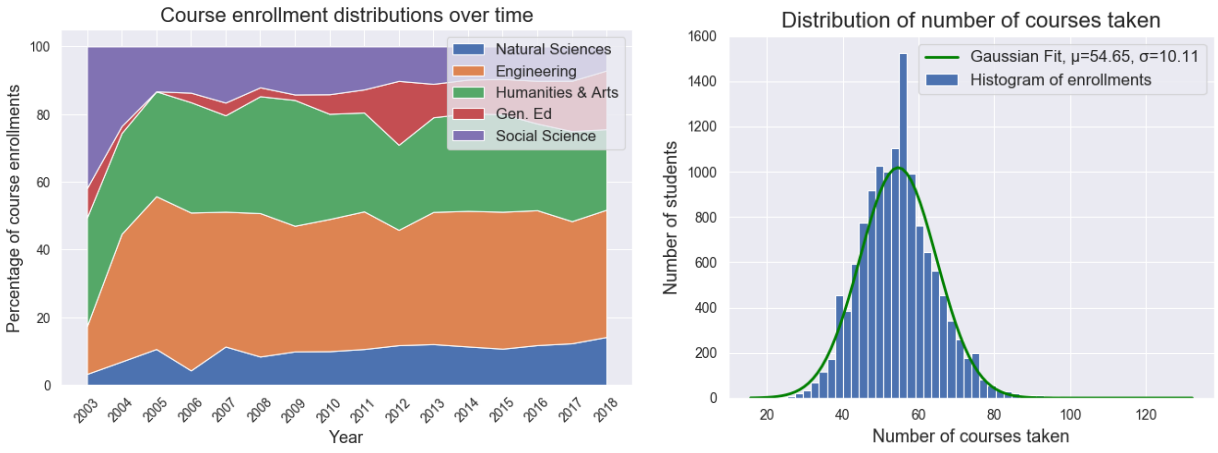

Figure 1: Top: A stacked area plot of enrolled courses per university sub-school per year. Bottom: A histogram of the number of courses taken by each individual student with a Gaussian fitting.

$x_j^i \in \{0, 1\}$ (0 denoting no enrollment and 1 an enrollment). The generative process of the model is described below:

$$h^i \sim \text{Multinomial}(\theta)$$
$$x_j^i \mid h^i \sim \text{Multinomial}(\phi_j)$$

Given the value of the hidden variable, each individual probability of enrolling in a course is independent. This is a poor inductive bias because enrollment decisions often influence one another, and the number of courses taken by a student in a given year is dependent on the courses taken. It is easier to capture these two facets of the data if we can model enrollments jointly, without strong independence assumptions.

2.1.2 *Gaussian Mixture Model*

Any joint probability distribution over all discrete combinations of $x^i \in \{0, 1\}^m$ requires $2^m - 1$ parameters and is thus intractable. One possible solution to is relaxation of the discrete problem to a real-valued vector space with $\bar{X}^i \in \mathbb{R}^m$ and

$$x_j^i = \begin{cases} 1 & \bar{x}_j^i > 0 \\ 0 & \text{else} \end{cases}$$

By training a model over $\bar{X}$, we can take advantage of real-valued distributions with much smaller parameter spaces.

The Gaussian Mixture Model (GMM) is an archetypal latent variable model for real-valued data [22]. We can describe a GMM by generative process below:

$$h^i \sim \text{Multinomial}(\theta)$$
$$\bar{x^i} \mid h^i \sim \mathcal{N}(\mu, \Sigma)$$

We can modify the GMM for the setting of multi-label classification by providing an unbiased estimator of the probability of each binary sample:

$$P(x = [1, 0, ..., 1]) = P(\bar{x}_0 > 0, \bar{x}_1 < 0, ..., \bar{x}_0 > 0)$$
$$\approx \frac{1}{K} \sum_{i=1}^{K} \Phi(\vec{0}; \mu(y^i), \Sigma(y^i)) \mathbb{1}[y^i > \vec{0}]$$

where y^i are samples drawn from a multivariate normal over a subset of the variables in x and $\mu(y^i)$, $\Sigma(y^i)$ are the parameters of a multivariate normal conditioned on the value of y^i. More detail on this estimator is provided in the online posting of this paper.

At face value, it might seem odd to model enrollments as Gaussian-distributed. We choose this particular model both because it makes our real-valued relaxation tractable and because we think it is reasonable to assume enrollments within each cluster will be fairly unimodal and smooth, especially as sample sizes increase.

2.1.3 *Contextual Mixture Model*

Hidden Markov Models (HMMs) are a common extension of stationary mixture models to sequential data [21]. In these models, the single latent variable is replaced with a Markov chain of hidden states. This model is naturally recursive, a property that is extremely useful when modeling processes that are positive recurrent. However, as enrollments often exhibit a strict order and returning to previous states is unlikely, we prefer a model that is strictly time-dependent or, as we will call it here, "contextual." In general any Contextual Mixture Model (CMM) can be expressed using a Hidden Markov model, but enforcing this structure allows us to incorporate priors that significantly improve the chances of training a plausible model.

For a CMM with Gaussian emission probabilities, we have

$$h^0 \sim \text{Multinomial}(\theta)$$
$$h^{t+1} \mid h^t \sim \text{Multinomial}(\phi^t)$$
$$\bar{x}^t \mid h_i^t \sim \mathcal{N}(\mu_i^t, \Sigma_i^t)$$

Note that the parameters of the transition and emission distributions are different for each timestep. Figure 2 shows a diagram of our proposed model in plate notation.

The small, discrete latent space of our model offers highly interpretable representations compared with the continuous latent vector space of neural architectures (see Fig. 4) and inference is highly efficient as the model has low tree-width.

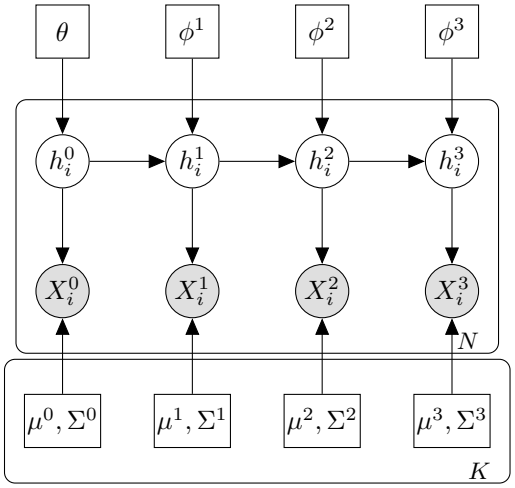

Figure 2: Graphical representation of contextual mixture model using plate notation

Our discrete probability estimator also allows modeling of all courses jointly, which is essential in this setting.

2.1.4 Parameter Learning

There are two primary methods of learning the parameters of the model we propose: expectation-maximization (EM) [8, 5] and gradient descent [6] on the log-likelihood objective. The approximate probability estimate of the model also creates the possibility for differing levels of precision in accordance with the amount of computation one is willing to invest. For a highly biased learning process, one can simply train the model on data shifted from $[0, 1]$ to $[-1, 1]$ using the exact probability estimate. For a less biased learning process, one can use the probability estimator described in Section 2.1.2 and unbiased estimator of the gradient with respect to the model parameters. For more details, see the online posting of the paper.

2.2 Baseline Models

In Section 4 we present a comparison of our model with three baseline models. The first of these is an naive Bayes model with the strongest model assumptions. The second is tree-augmented naive Bayes [9], which adds dependencies between variables to better model the joint density. Both models with trained with EM.

The last model we use for comparison is a Variational Autoencoder (VAE)–a deep generative model [15]. We use a simple VAE with two fully connected layers in the encoder and decoder, trained on binary cross-entropy loss.

It is important to note that while the VAE offers a good comparison point, the type of conditional inference (over sets of courses) that we describe for our Gaussian relaxation are not tractable in a standard VAE framework. In fact, VAE models can suffer from suboptimal inference in general when there is overfitting of the decoder network [7]. This issue is particularly concerning in this setting with relatively small sample size.

3. EXPERIMENTS

In this section, we describe the data used to train the model presented in Section 2 and how we evaluated them during training.

3.1 Data

We use eighteen years of course enrollment data from a large private university in the United States. The data comprise approximately $30,000$ student enrollment records with fields for course name and student major. We removed part-time and summer students from the dataset, limiting the analyses presented here to full-time academic-year enrollments only.

Figure 1 shows two basic visualizations of the data after preprocessing. There are at least two notable takeaways from these plots. First, the proportion of enrollments in each academic division within the university remains relatively stable through most of the time period represented in the dataset. We use this fact to aggregate over time without explicitly modeling changes in enrollment patterns. Second, the fact that the number of courses taken is approximately Gaussian-distributed shows that enrollment patterns are not intensely multi-modal; thus the assumptions of probabilistic model are plausible.

In what follows we replace full course names with abbreviated proxies to enable universal legibility. For example, CS1 corresponds to the introductory computer science class and "Alg" or "AI" correspond to algorithms or artificial intelligence classes respectively.

4. EVALUATION

4.1 Mean-Field Evaluation

Though we can compare many of the models under consideration with log-likelihood alone, some only offer an approximate lower bound (VAEs). Thus we provide another evaluation metric that can be used to compare any model that can generate sampled enrollments.

For this loss function, we compare the empirical enrollment distributions in samples from our model and the distributions of the hold-outs. Let p_j^t be the probability that class j is taken by any given student in the hold-out data, and p_j^s be the corresponding probability in the samples. We take as our error, $E(p^t, p^s)$ with

$$E(p^t, p^s) = \sum_j (p_j^t - p_j^s)^2$$

which approximates the distance between the two true multivariate distributions–the distribution of our model and the distribution of the data–if all the variables were independent (mean field approximation).

4.2 Sample Quality

In Figure 3 we compare the performance of our model on *hold-out data* relative to baseline models described in Section 2.2. We also include a direct comparison of the best performance for each model in Table 1.

In Figure 3 we can see that our proposed model outperforms the two baselines across the board. It also is evident that the VAE baseline suffers bad generalization as the complexity of

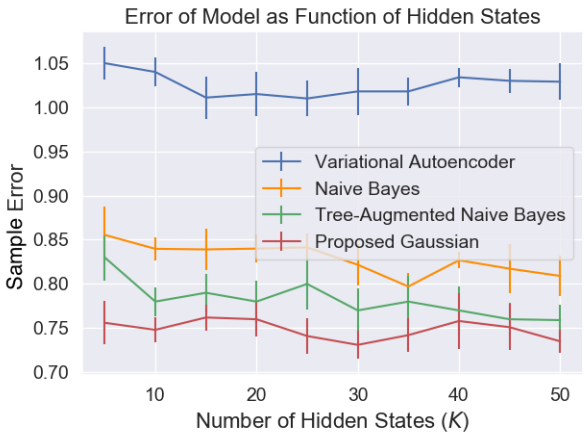

Figure 3: Plots of the error for the proposed model and other models. The parameter K is the dimension of the latent space.

Model	Sample Error	Inference Acc.
Deep Generative	1.01	N / A
Naive Bayes	0.78	60%
Tree-Augment Naive Bayes	0.76	66%
Our Gaussian	**0.72**	**86%**

Table 1: A direct comparison of the best performance from each model on hold-out data

the model increases. These models were trained adaptively, according to performance on the validation set, and thus are not simply underfitting due to increased training complexity.

In comparing the graphical models, increased complexity–both in the observation model and latent space–leads to lower error. As the error calculation itself makes an independence assumption, it is not surprising that the performance of all three graphical models is relatively close. The true dominance of the model proposed here is perhaps most evident in the inference task of Table 1, described in Section 5.3.

4.3 Visualizing Hidden Variables

Beyond using the loss function defined in Section 4.1, we can also examine the hidden states of a trained model to validate the learning process. In particular, we can investigate whether the hidden space captures semantically meaningful categories. Figure 4 shows a visualization for our model trained on CS majors. The clusters in the grid correspond to required courses for three different concentration within the major[1], and the color shows the most likely latent state assigned by the model to each course. As we can see, courses within the same concentration are assigned strikingly similar latent states by the model, suggesting that the model captures a semantically meaningful notion of the different concentrations in its hidden state. Therefore, if there are unknown correlations in course enrollments—for example

[1]These requirements were taken from the department website: `https://exploredegrees.stanford.edu/schoolofengineering/computerscience`.

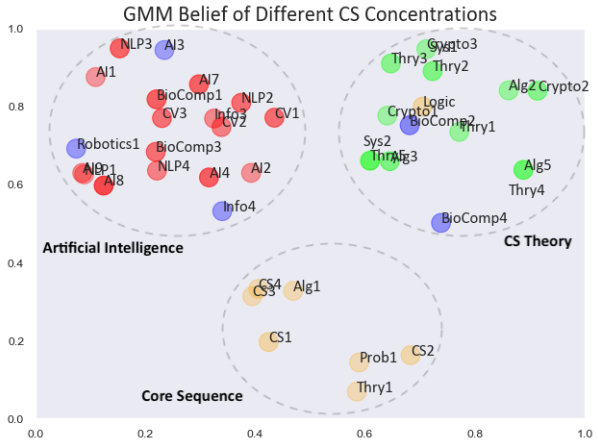

Figure 4: A visualization of the semantic meaning captured by the latent space of the model. Coloring corresponds to hidden state, and translucency indicates the confidence of the model.

many AI and biology courses taken together—this model could bring these patterns to the fore, allowing administrators an insight into possible ways to improve their degree concentrations.

5. APPLICATIONS

In this section we present results from two different experiments performed with our proposed model. These applications demonstrate only a fraction of the model's scope, but show its power to provide insights.

5.1 Quantifying Enrollment Likelihood

One of the useful applications made simple by our generative model is in quantifying enrollment likelihood. A model trained on student enrollments will approximate the distribution of the training data. Thus if we evaluate the likelihood of a new student's enrollments given the model, we can get a sense of how this student differs from the training examples. Taking this principle to its extreme, we can train a model for each student on every other student's enrollments, allowing us to model exactly how much each particular student varies from the typical.

By examining the classes taken by students who are evaluated as high versus low likelihood, we see that the model captures at least two meaningful axes of variance. Firstly, it recognizes that it is rare for students to take a very diverse set of courses spanning many academic subjects. This insight is demonstrated in Figure 6, which shows the average coursework for each type of student. The second insight that the model captures is the spectrum of ambition. More specifically, the model places very low probability on the small subgroup of students that take up to 30 computer science classes and places high probability on taking just the core requirements of the degree[2]. Atypical students take about 20 more courses than their counterparts on average.

[2]We can identify this trend by looking at the exact classes that are most commonly taken by these students e.g. the core requirements.

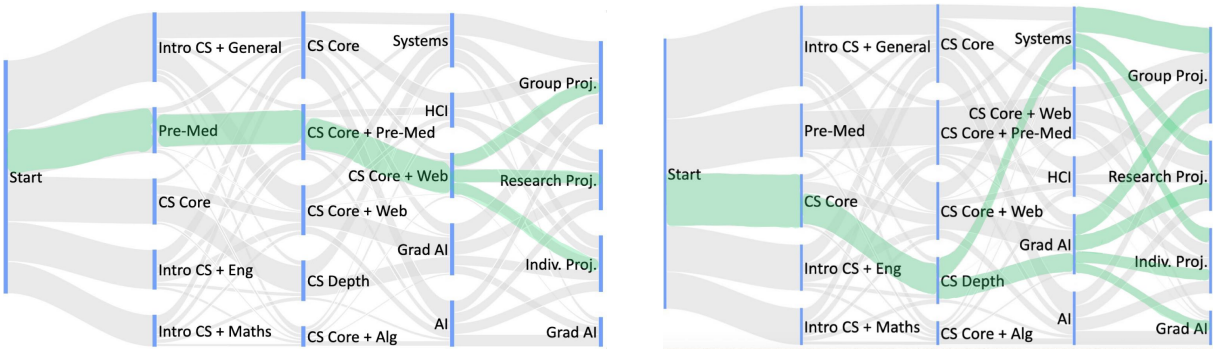

Figure 5: Two shadings of the same Sankey diagram constructed from the CMM trained on CS undergraduate enrollments. Top: A common path taken by students engaging in pre-med requirements is highlighted in blue. Bottom: A common path for students committed to in-depth study of computer science is highlighted.

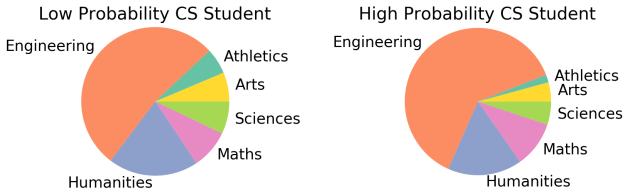

Figure 6: Pie charts representing the difference between enrollment patterns captured by the model. Sections correspond to the average number of courses taken in a academic subject.

5.2 Understanding Pathways

Another capability of the model presented here is the ability to analyze sequences of enrollments, from inferring likely paths between X and Y, to uncovering unspoken student strategies. We can display this visually using Sankey diagrams in which the width of the line between adjacent segments is proportional to the transition probability between the corresponding hidden states in the model. Figure 5 shows this style of Sankey for CS students. In this diagram, we can note the two types of paths highlighted in the diagram. The first of these captures students who were actively taking the pre-medical requirements freshman and sophomore year. These same students were subsequently much more likely to take depth courses later and were more likely to focus on web development of information systems in their depth courses. We can contrast these students with the students that are highly committed to the CS major and its core classes starting freshman year. These students are much more likely to enroll in depth classes by their sophomore year and are predisposed towards the systems and AI concentrations within the major.

5.3 Inferring Intermediate Classes

Another unique capability of our model is inferring the likelihood of intermediate classes. Given the classes taken freshman year and goal classes for senior year, the model can place a likelihood on intermediate classes. One possible use case for this ability is inference of soft or tacit prerequisites for courses.

To test this aspect of the model, we predicted whether students would take each of 5 common classes in their sophomore year given freshman and senior year enrollments. We were able to recover the correct enrollment with around 86% probability. From this result it is clear that the model can learn a sensible joint distribution over multi-year enrollments. We can compare this performance with that of the baseline models in Table 1, noting a substantial gain.

An even more interesting use case of this inference ability, however, is not simply prediction of common courses, but the potential for improving course selection tools. Only a model that captures the temporal dependencies across all courses is capable of offering helpful insights for goal-directed course selection.

6. PRIOR WORK

Much of the prior work on enrollment modeling in the university setting is dedicated purely to predictive models of future course enrollment [13, 18, 24] and academic performance [16, 11]. These models are largely incapable of producing the kinds of insights shown here. Preliminary work has also seen application of clustering algorithms to enrollments in form of simple latent variable models like Latent Dirichlet Allocation (LDA) [17] and recurrent neural networks [20].

Much of the state-of-the-art research in student decision modeling is now found in the study of massive open online courses (MOOCs). Gardner and Brooks [10] provide a thorough overview of modern models for the problem setting. Of note, Balakrishnan and Coetzee use a Hidden Markov Model (HMM) to predict attrition in MOOCs [4]. Similarly, Al-Shabandar et al. use Gaussian Mixture Models (GMMs) to cluster MOOC students at each timestep, and thus identify clusters of students that are likely to withdraw from the courses [1]. Both of these models resemble ours though their task is prediction of simple binary outcomes.

Work in course recommender systems is also inspiring. Khorasani et al. create a recommender based on a Markov model [14]. Jiang et al. use a neural-network system [12], and add the choice of using grade considerations to create custom course recommendations (also see [19]). This second model yields extremely compelling results, but is not capable of the

broad range of inference queries possible with our model.

7. CONCLUSION & FUTURE WORK

We have presented a new probabilistic model that is capable of capturing joint relationships between course enrollments, while also allowing for powerful inference queries. There is, however, at least one important drawback to our approach: the strictly Markovian character of the model. Although this assumption allows us to easily learn model parameters, in practice the enrollments observed at one timestep will impact those sampled at the next timestep. Because of this inductive bias our approach is effective with less training data than, for example, a recurrent neural net, and is therefore more easily deployed for institutions smaller in size than our case university.

We emphasize the potential for future work that links data of the sort investigated here with other rich information, such as demographic information describing students, and earned grades. Models incorporating such information could meaningfully identify differences between course trajectories of particular kinds of students, providing insights into how academic policies and programs might be tuned to benefit specific constituencies.

8. REFERENCES

[1] R. Al-Shabandar, A. Hussain, R. Keight, A. Laws, and T. Baker. The application of gaussian mixture models for the identification of at-risk learners in massive open online courses. *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2018.

[2] T. R. Bailey, S. S. Jaggars, and D. Jenkins. *Redesigning America's community colleges*. Harvard University Press, 2015.

[3] R. Baker. Understanding college students' major choices using social network analysis. *Research in Higher Education*, 59(2):198–225, Mar 2018.

[4] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2013.

[5] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[7] C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. *arXiv preprint arXiv:1801.03558*, 2018.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[9] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[10] J. Gardner and C. Brooks. Student success prediction in moocs. *User Modeling and User-Adapted Interaction*, 28:127–203, 2018.

[11] M. Hlosta, Z. Zdrahal, and J. Zendulka. Ouroboros: early identification of at-risk students without models based on legacy data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 6–15. ACM, 2017.

[12] W. Jiang, Z. A. Pardos, and Q. Wei. Goal-based course recommendation. *CoRR*, abs/1812.10078, 2018.

[13] A. A. Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani. Prediction of student course selection in online higher education institutes using neural network. *Computers & Education*, 65:1–11, 2013.

[14] E. S. Khorasani, Z. Zhenge, and J. Champaign. A markov chain collaborative filtering model for course enrollment recommendations. *2016 IEEE International Conference on Big Data (Big Data)*, pages 3484–3490, 2016.

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Z. Kovacic. Early prediction of student success: Mining students' enrolment data. 2010.

[17] B. Motz, T. A. Busey, M. E. Rickert, and D. Landy. Finding topics in enrollment data. In *EDM*, 2018.

[18] A. Nandeshwar and S. Chaudhari. Enrollment prediction models using data mining. *Retrieved January*, 10:2010, 2009.

[19] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild. *arXiv preprint arXiv:1803.09535*, 2018.

[20] Z. A. Pardos and A. J. H. Nam. A map of knowledge. *CoRR*, abs/1811.07974, 2018.

[21] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[22] D. Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832, 2015.

[23] J. Scott-Clayton. The shapeless river: Does a lack of structure inhibit students' progress at community colleges? ccrc working paper no. 25. assessment of evidence series. *Community College Research Center, Columbia University*, 2011.

[24] Q. Song and B. S. Chissom. New models for forecasting enrollments: Fuzzy time series and neural network approaches. 1993.

[25] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

APPENDIX

A. MATHEMATICAL DETAILS

Estimator: For our proposed Gaussian model we have

$$\begin{aligned} P(x = [1, 0, ..., 1]) &= P(\bar{x}_0 > 0, \bar{x}_1 < 0, ..., \bar{x}_0 > 0) \\ &= P(\bar{x}_G > \vec{0}, \bar{x}_L < \vec{0}) \end{aligned}$$

where G and L denote the indices where $\bar{x}$ is greater than or less than zero respectively, and inequalities denote element-wise inequalities.

$$\begin{aligned} P(\bar{x}_G > \vec{0}, \bar{x}_L < \vec{0} &= \int_0^\infty P(\bar{x}_L < \vec{0} | \bar{x}_G = y) P(\bar{x}_G = y) dy \\ &\approx \frac{1}{K} \sum_{i=1}^{K} \Phi(\vec{0}; \mu(y^i), \Sigma(y^i)) \mathbb{1}[y^i > \vec{0}] \\ & \qquad y^i \sim \mathcal{N}(\mu_G, \Sigma_G) \end{aligned}$$

where $\mu(y^i)$, $\Sigma(y^i)$ are parameters of conditional normal distribution

For most practical inference queries, the cardinality of G is small enough that this estimator is tractable as is. For learning, however, the Gaussian CDF and indicator variable can become poor reward signals, and thus we use a product of independent Gaussian CDFs and $\frac{1}{M} \sum_j \mathbb{1}[y_j^i > 0]$ instead. To learn the parameters of Gaussian distribution we begin with the simple -1/1 approximation and standard EM updates and then refine with this estimator, optimizing the parameters via policy gradient optimization [25].

To learn the parameters of the full CMM, we use EM with the following parameter updates.

E Step: We calculate

$$\begin{aligned} Q(h^t \mid X_i) &= \frac{\alpha_k(t)\beta_k(t)}{\sum_k \alpha_k(t)\beta_k(t)} \\ Q(h^t, h^{t+1} \mid X_i) &= \frac{\alpha_k(t)\phi_{kk'}^t \mathcal{N}(X_{t+1}; \mu_{k'}, \Sigma_{k'})\beta_{k'}(t)}{\sum_k \sum_{k'} \alpha_k(t)\phi_{kk'}^t \mathcal{N}(X_{t+1}; \mu_{k'}, \Sigma_{k'})\beta_{k'}(t)} \end{aligned}$$

with

$$\begin{aligned} \alpha_k(0) &= \theta_k \mathcal{N}(X^0; \mu_k^0, \Sigma_k^0) \\ \alpha_k(t+1) &= \mathcal{N}(X^{t+1}; \mu_k^{t+1}, \Sigma_k^{t+1}) \sum_{k'} \alpha_{k'}(t)\phi_{kk'}^t \end{aligned}$$

and

$$\begin{aligned} \beta_k(T-1) &= 1 \\ \beta_k(t) &= \sum_{k'} \beta_{k'}(t+1) \mathcal{N}(X^{t+1}; \mu_{k'}^{t+1}, \Sigma_{k'}^{t+1}) \phi_{kk'}^t \end{aligned}$$

M Step: If we use the -1/1 approximation we can find parameter estimates maximizing the evidence lower bound as

$$\begin{aligned} \theta &= \frac{1}{N} \sum_i Q(h^0 \mid X_i) \\ \phi^t &= \frac{\sum_i Q(h^t, h^{t+1} \mid X_i)}{\sum_i Q(h^t \mid X_i)} \\ \mu_k^t &= \frac{\sum_i Q(h_k^t \mid X_i) X_i^t}{\sum_i Q(h_k^t \mid X_i)} \\ \Sigma_k^t &= \frac{\sum_i Q(h_k^t \mid X_i)(\mu_k^t - X_i^t)(\mu_k^t - X_i^t)^T}{\sum_i Q(h_k^t \mid X_i)} \end{aligned}$$

otherwise we update parameters using the stochastic probability estimator and policy gradient optimization.