Developing a Latent Variable Model for Critical Care Patients in the MIMIC-III Dataset using Variational Autoencoders

Scott Fleming (scottyf), Jesper Westell (jesperw), and Matt Millett (millett)
GitHub Repo: `https://github.com/millett/cs230_project`

Abstract

Electronic Health Records (EHR) are used extensively across the United States. This gives researchers access to large amounts of data and the ability to train deep neural networks for various tasks. We used Variational Autoencoders (VAEs) to encode physiological information from patient vitals, labs, and other items in their EHR. A traditional VAE more accurately encoded patient physiology into a few variables than traditional PCA, as reflected in our log-likelihood scores. However, the traditional VAE projects latent variables onto a single Gaussian distribution, where a mixture of Gaussians might be more appropriate and better capture physiological differences between patients. We therefore built a Gaussian Mixture VAE (GMVAE) to cluster patients by their input variables as part of the encoding process. The GMVAE grouped input variables more discretely across latent variables than the traditional VAE. Also, t-SNE plots of latent encodings of patients showed that the GMVAE can indeed better cluster patients by outcome variables, despite not being trained explicitly for the task. Transforming many inputs into a few latent variables using a GMVAE seemed to be an effective way to encode all the input information while still representing the underlying physiology (and potential prognoses) of patients.

1 Introduction & Motivation

Implementation and use of Electronic Health Record Systems (EHRs) in U.S. hospitals has exploded over the last decade, rising from less than 10% nationwide EHR adoption in all U.S. hospitals 2008 to more than 80% adoption by 2015 [7]. Adoption of EHR systems allows clinicians and researchers to more easily access vast amounts of data, and enables them to find patterns in patients that have not been found with traditional analysis methods (e.g. firsthand exposure by clinicians). We use the Variational Autoencoder, an unsupervised learning method, to discover latent features of Intensive Care Unit (ICU) patient data. The large amounts of data available on a single patient could overwhelm a clinician - dozens of vitals and lab test data points could hold useful information, but can be hard to interpret efficiently or even correctly [13]. A latent representation of encoded patient features could give a quick snapshot of a patient's physiological state and potential trajectory based on a few features, instead of dozens of values for demographics, vitals and labs.

2 Related Work

Many traditional studies applying machine learning to the healthcare setting use supervised learning techniques [2], where an expert labels large amounts of data for one specific task. A fundamental characteristic of these supervised models, however, is that they can only be trained on specified outcomes of interest, which are deemed "interesting" by the model's designer. Unsupervised learning, in which learned features are derived from the structure of input features alone and are not trained against an arbitrarily specified set of outcomes, is an attractive alternative in settings where it is impossible to enumerate all possible outcomes. The hospital setting, especially the ICU, fits into this paradigm [14].

Neural network-based unsupervised learning methods can be particularly effective for learning complex features of data when many observations are available. Large image databases have thus appropriately been targeted by researchers aiming to persuasively demonstrate the utility of novel unsupervised methods, including the Variational Autoencoder (VAE) [11] and Generative Adversarial Networks (GANs) [5]. With the increased quantity of medical data available, researchers have also begun exploring deep unsupervised learning methods in the healthcare setting. One example includes Deep Patient [15], which used stacked denoising autoencoders to extract clinically relevant features from a large EHR dataset. The authors turned 60,238 clinical descriptors into a dense vector of 500 features; using these learned patient representations as input to downstream predictive models improved accuracy for some outcomes. We analyzed a smaller set of 65 input features which are particularly relevant to the ICU setting and employed both a traditional VAE model as well as a novel Gaussian Mixture VAE (GMVAE) architecture which incorporates cluster discovery into the feature learning process.

3 Dataset and Features

We obtained data from the MIMIC-III Critical-Care Database, including de-identified EMR data for about 43,000 patients and 9,800 pediatric patients who stayed in the Beth Israel Deaconess Medical Center ICU between 2001 and 2012 [9]. The database contains de-identified record-level data including demographics, hourly vital measurements, lab test results, medications, mortality, ICD9 diagnoses, and discharge summaries for each patient. It was created by PhysioNet, an organization supported by the National Institute of General Medical Sciences (NIGMS) and the National Institute of Biomedical Imaging and Bioengineering (NIBIB), both of which are supported by an NIH grant. We had to formally

request access to the database; Although the data is de-identified, it still contains personal care information about patients. To gain access, we completed the CITI "Data or Specimens Only Research" course and registered with PhysioNet.

3.1 Data Preprocessing and Feature Engineering

As we were interested in deriving clinically useful features with the VAE, we decided to train our model on all labs and vitals recorded for each patient in the first 24 hours of their ICU stay. Care teams will often establish a baseline set of vitals and order a swathe of labs in the patient's first hours in the ICU. Clinicians must then interpret this deluge of information to make important decisions about the patient's treatment plan [4] and can easily make errors in determining the risk of each patient for an endless list of potential adverse outcomes [13]. A low-dimensional representation of these features thus provides a "snapshot" of each patient which is potentially indicative of underlying physiological processes.

To demonstrate the potential utility of our model, we also extracted outcomes of interest from the patient records, including flags for mortality in the hospital/ICU, congestive heart failure (CHF), atrial fibrillation, respiratory failure, and pneumonia. These outcome variables were extracted from the patient's entire ICU stay, not just the first 24 hours, and were identified with ICD-9 codes specified by [9].

As the dataset holds a record for every measurement and lab event (rather than a visit-level summary) and not all variables are recorded at each chart time, we used last-value-carried-forward imputation [10] to fill in entries for all missing lab values subsequent to a recorded value. Unreasonable outliers (e.g. ages over 300 years and heights of 16 ft.) were mean-replaced. We used K-Nearest Neighbors to impute any remaining missing values. All features were extracted from the MIMIC-III database using PostgreSQL 10.3 [16] and cleaned/imputed using the caret package in R [12].

4 Methods

We chose to build a variational autoencoder (VAE) to find latent characteristics of patients in the dataset. The VAE allows us to transform the many variables recorded in the ICU into a few key indicators for doctors to use. One example is that we could potentially capture "cardiovascular risk" into one variable, so clinicians have a better sense of what to watch out for when a new patient enters the ICU.

4.1 VAE with a Multivariate Gaussian Prior

We implemented the VAE model originally proposed by Kingma et al. (2013) in Keras [3]. The VAE encodes the input data vector, $\mathbf{x}$, into a latent representation, $\mathbf{z}$, before adding noise to the latent representation and decoding the noisy, latent representation back into the original data space. By utilizing a cost function that penalizes decoded representations which differ from the original data and by incorporating a penalty for completely arbitrary encodings, the VAE model encourages a 'meaningful' latent variable representation of the data [11]. We assume the generative model

$$p_\theta(x, z) = p_\theta(z)p_\theta(x \mid z) \tag{1}$$
$$z \sim \mathcal{N}(0, \mathrm{I}) \tag{2}$$
$$x \sim \mathcal{N}(\mu_x(z), \sigma_x^2(z)) \tag{3}$$

and inference model,

$$q_\phi(z \mid x) \tag{4}$$
$$z \sim \mathcal{N}(\mu_z(x), \sigma_z^2(x)) \tag{5}$$

In general, the model uses gradient ascent to maximize the following variational lower bound on the data log-likelihood:

$$\begin{aligned} \mathcal{L}(\theta, \phi; x) &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x, z) - \log q_\phi(z \mid x)] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(z)}{q_\phi(z \mid x)} + \log p_\theta(x \mid z)\right] \end{aligned} \tag{6}$$

How this is implemented using a neural network is described in appendix A.

4.2 VAE with a Gaussian Mixture Prior

In the traditional VAE, where the latent variable prior takes on a standard multivariate normal distribution, there will likely be no clusters in the learned representation of the latent variables, as the loss function will "push" the latent distribution toward the non-cluster prior by penalizing latent representations that are far from the prior, as measured by KL-divergence. However, we hypothesize that there may be underlying distinctions between groups of patients in the

dataset. For that reason we have experimented using a Gaussian Mixture Prior for the latent variables [17]. For a model with K distinct Gaussians, we assume the generative model,

$$p_\theta(x, y, z) = p_\theta(y)p_\theta(z \mid y)p_\theta(x \mid z) \tag{7}$$

$$y \sim Cat\left(\frac{1}{K}\right) \tag{8}$$

$$z \sim \mathcal{N}(\mu_z(y), \sigma_z^2(y)) \tag{9}$$

$$x \sim \mathcal{N}(\mu_x(z), \sigma_x^2(z)) \tag{10}$$

and inference model,

$$q_\phi(y, z \mid x) = q_\phi(y \mid x)q_\phi(z \mid x, y) \tag{11}$$

$$y \sim \text{Multinomial}(\theta(x)) \tag{12}$$

$$z \sim \mathcal{N}(\mu_z(x, y), \sigma_z^2(x, y)) \tag{13}$$

The variational lower bound is defined as

$$\begin{aligned}\mathcal{L}(\theta, \phi; x) &= \mathbb{E}_{q_\phi(y,z|x)}[\log p_\theta(x, y, z) - \log q_\phi(y, z \mid x)] \\ &= \mathbb{E}_{q_\phi(y,z|x)}\left[\log \frac{p_\theta(y)}{q_\phi(y \mid x)} + \log \frac{p_\theta(z \mid y)}{q_\phi(z \mid x, y)} + \log p_\theta(x \mid z)\right]\end{aligned} \tag{14}$$

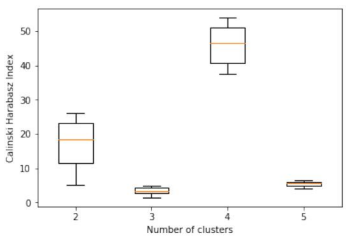
The neural network representation of this is described in appendix B.

4.3 Choosing Hyperparameters

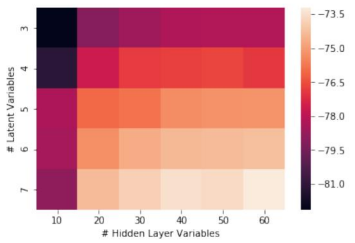
To decide on the number of units per hidden layer in our VAE and how many latent variables to use, we ran a grid search to examine performance. We ultimately ended up choosing to use 5 hidden units in the latent variable layer and 65 hidden units in the intermediate layers, because, as the number of hidden units in the intermediate layer increased, model performance increased (see Figure 1(b)). As there were only 65 features in the training data, we decided to cap the number of hidden units in the intermediate layer at 65. Subsequent experiments showed that even dramatic increases in the number of hidden units in the intermediate layer (e.g. >200) did not significantly improve model performance. We also decided to use just 5 latent variables for ease of interpretability (the whole point of the autoencoding exercise was to compress the information into a clinically useful summary, after all). We used these same hyperparameters in our GMVAE as a means of comparison.

In training the VAE model, we occasionally ran into overflow issues with the loss. In order to address this issue, we incorporated batch normalization [8], initialized the ReLU layers with He initialization [6], and initialized the $\sigma_z(x)$ vectors to have unit variance. These changes effectively stabilized the training process. We used an Adam optimizer in our model training, with learning rate $\alpha = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

To find the number of latent clusters to use in the GMVAE model, we calculated at the Calinski-Harabasz [1] index for each number of clusters on our dataset. For each data point, we sampled its latent representation 10 times and looked at the score for each sample. The Calinski-Harabasz (CH) score measures the ratio of average between-clusters dispersion to the average within-cluster dispersion across all clusters. Thus between-cluster dispersion will be greater than within-cluster dispersion when clusters are dense and well-separated (i.e. a "good" clustering) and this will correspond to a higher CH score. We therefore chose to use the number of clusters corresponding to the highest CH score (see Figure 1(a)).



(a) Calinski-Harabasz box plot showing goodness of each number of cluster.

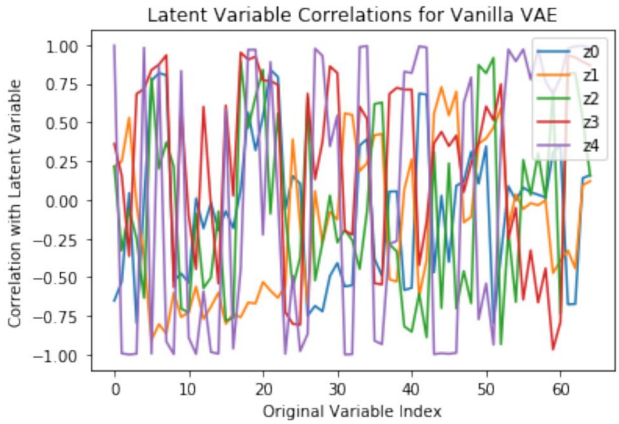


(b) Grid search for our VAE hyperparameters. Colors represent negative log-likelihoods.

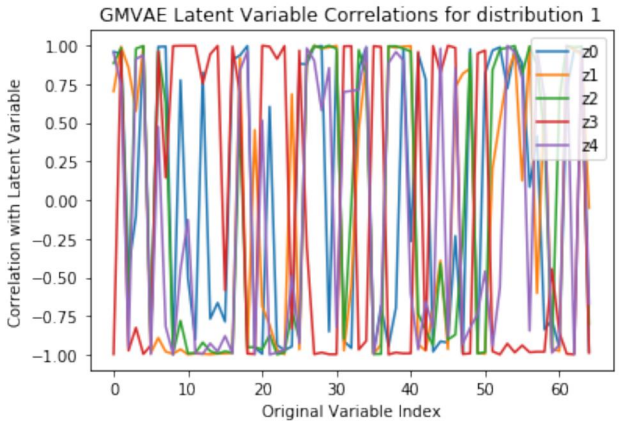
Figure 1: Hyperparameter search results

5 Results & Discussion

First, to test whether our GMVAE architecture could adequately infer cluster assignment and representation of data generated from a latent cluster model, we generated a dataset of 2 Gaussian clusters in 2 dimensions and visually assessed cluster tendency at the latent variable layer. The model managed to capture the variables among the two latent clusters, and correctly categorize them. The model also succeeded in reconstructing the data in an acceptable manner. As a



(a) Conventional VAE correlations for each latent variable.



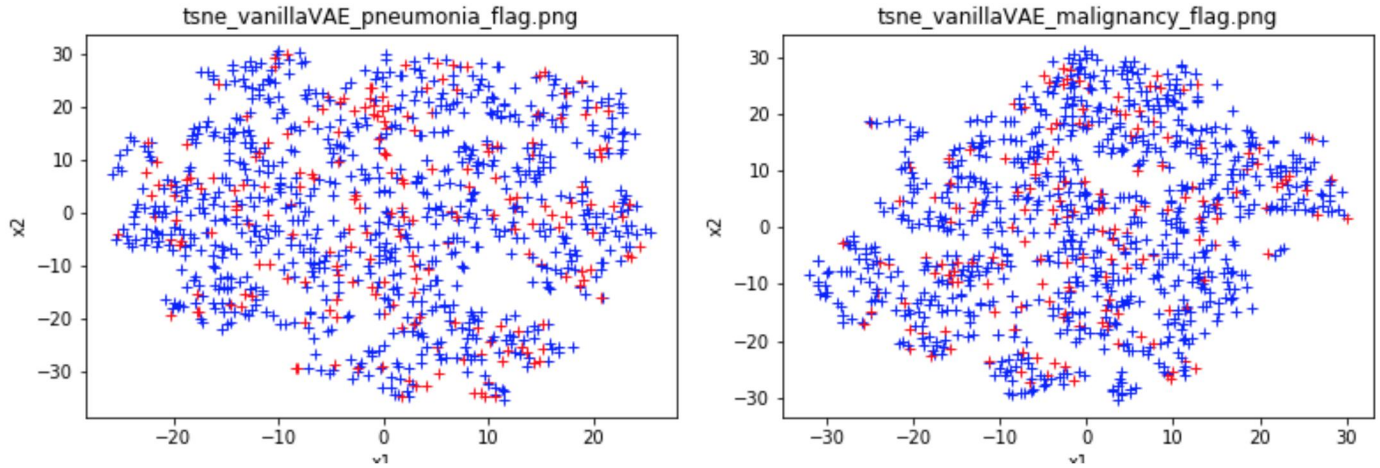
(b) Correlations from one of the GMVAE distributions.

Figure 2: Correlations between input variables from ICU patient records and latent variables for the conventional VAE and the GMVAE. The GMVAE correlations look similar for all GMM distributions in our model, so we only show one here. But it is clear the relationships between input variables and latent variables are more discretely separated in the GMVAE.

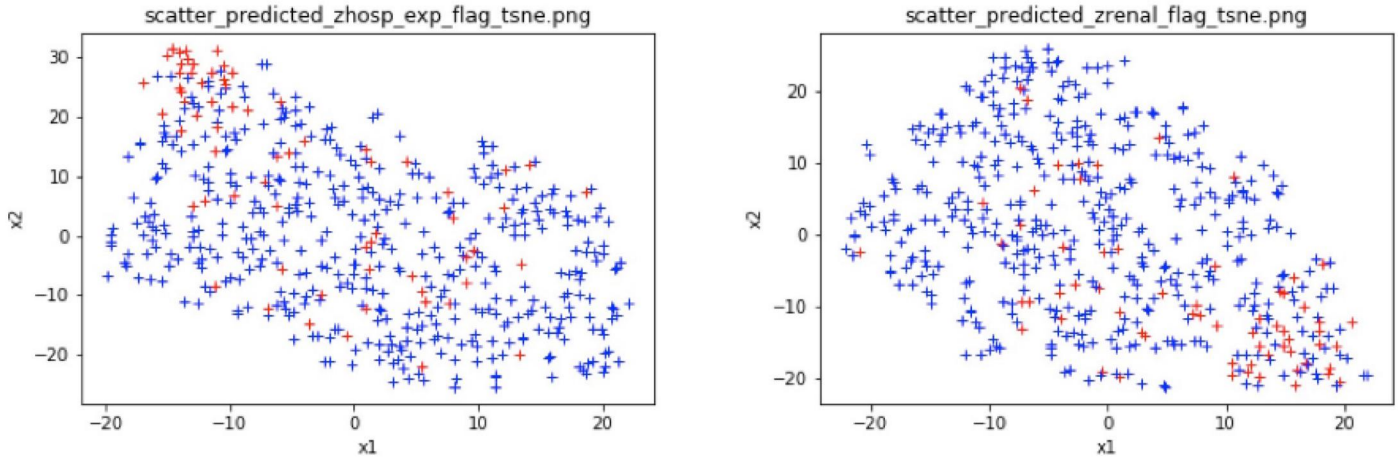
benchmark comparison for our VAE model's performance, we compared validation log-likelihood under our VAE model (e.g. "How likely is the validation data given our VAE model parameters?") with the validation log-likelihood under Probabilistic PCA and Factor Analysis, each constrained to 5 latent variables. We found that our VAE model was able to achieve a higher log-likelihood (-74.78346) compared to both PCA (-81.0489) and Factor Analysis (-77.53086) (see Figure 5 in the Appendix). This, together with our test of the GMVAE model, seemed to confirm that our models were accurately capturing an information-rich latent variable representation of the MIMIC data.

Next, in order to interpret the learned latent variable representations, we mapped the input to the latent variable space and, manipulating one latent variable at a time, examined the correlation between manual changes in the latent variable and changes in the generated data. This gave a proxy for the "meaning" of each latent variable. In cases where increasing latent variable z_i consistently led to increases in original variable x_i, we interpreted this to mean that z_i was closely associated with x_i. Both the GMVAE and conventional VAE were able to capture physiologically meaningful associations between latent variables and the input variables for patients in the MIMIC-III database (see Figure 2a). However, the GMVAE was able to capture those correlations more cleanly across all its distributions (e.g. each latent variable seemed to be highly associated with a smaller number of raw variables; see figure 2b). Thus, while both the VAE encoding and the GMVAE encoding seem to be able to capture underlying physiology, the clustering of the GMVAE gives a more clear relation to underlying physiology. The top associations with each latent variable from both models did seem to group in a medically sensible fashion. For example, one latent variable produced by the VAE correlated strongly with heartrate (0.83), lactate (0.86), and anion gap lab values (0.84), suggesting potential kidney failure due to problems with circulation; interestingly, the same latent variable was also highly associated with renal failure (0.83), congestive heart failure (0.81), and atrial fibrilation (0.85). See Appendix E for more correlation values. Apart from stronger correlations, the better encoding of physiology for the GMVAE can also be seen in the difference in outcome variables. For certain patient outcomes, such as if or when a patient dies after admission or has kidney failure, the GMVAE produces noticeable clustering behavior using t-SNE [19] for the latent variables, meaning that a doctor reading them might more easily interpret what a particular value means in terms of outcomes (see Figure . The ability to produce this clustering without including outcome data in the training suggests that the GMVAE effectively captures the patient's underlying physiology and potential prognosis in only 5 latent variables, taken from an original 65.

To develop further understanding of the nature of the GMVAE, we explored how well it would work in capturing latent features in the MNIST dataset, which can be read about in appendix C. The GMVAE worked better than expected, and we managed to get very interesting results, showing how each digit can be assigned to one cluster in the latent space. These results were much clearer than the ones received when working in MIMIC-III. We believe that the different types of digits may be more distinctly separated among images in the dataset, compared to the latent clusters of patients.



(a) Conventional VAE. Outcome colors represent the patient having pneumonia or a malignant tumor.



(b) GMVAE. Outcome colors represent in-hospital mortality and renal comorbidity.

Figure 3: t-SNE plots of latent outputs for patients from the conventional VAE (a) and the GMVAE (b). Red is positive, blue is negative. We compressed 65 inputs into 5 latent variables.

6 Conclusion and Future Work

The research presented in this paper shows promise of the use of Variational Autoencoders in this field, which needs to be explored further. We were able to meaningfully capture patient physiology from 65 variables and compress it into 5 latent variables. These latent variables represent clear physiological signs (e.g. lab tests and vitals), but also group across different outcome variables (like in-hospital patient mortality). Despite this, we need to stabilize the GMVAE as it isn't always able to cluster patient input variables. We also would like to find a larger dataset, such as UK Biobank [18], which has data for 500,000 patients. We could also explore other patient input variables or metrics, or compare our outcome clustering to the predictions of a multitask neural network. This is the beginning of a new era where doctors can get quick snapshots of their patients' underlying physiology without being overwhelmed with too many different variables and numbers. Ultimately, more research in this area will lead to more efficient doctors and a better healthcare system.

7 Contributions

Scott Fleming extracted/processed the MIMIC data, developed the VAE model, and performed the baseline analysis comparing VAE with PCA and Factor Analysis. Matt Millett analyzed relationships between latent variables and input and outcome variables in both the VAE and GMVAE model and created most of the figures for this paper. Jesper Westell developed the GMVAE model, as well as evaluation metrics used in experiments.