Bachelors thesis

Thomas B. L. Christensen
Department of Computer Science
University of Copenhagen
thomas.christensen@di.ku.dk

Abstract

Case control studies require matching cases with controls to adjust for the possible difference in distribution of corvariates in the two groups. We investigate the possibility leveraging the variational autoencoder framework to map the observed covariates to a representation used for matching. We train our models on EHR from SARS-CoV-2 positive patients, preprocessed into a series of snapshots of the patients state each 24 hours. It found that a 190-dimensional vector of health information can be compressed to around 15-20 dimensions before losing large amounts of information. It was also found that decoding features used in the encoding, and features not used in the encoding, but highly correlated with those in the encoding, can be achieved at very low dimensional representations. We also observed that examining the placement of different groups of patients, showed that the learned representation space separates the groups in the representation space, affirming the notion that distance in the representation space corresponds to similarity of patients.

Synopsis

Epidemiology study designs

An often posed question, is that of cause and effect. Will exposure to some factor have an effect on some outcome of interest? This question is posed in a variety of contexts, also with varying ways of deriving an answer. In this project, the focus will be on epidemiological hypotheses. Based on data retrieved from a sample of subjects from a population, we hypothesize whether some exposures has an effect on some outcome of interest. The term exposure is used to denote presence of factors, besides actual exposure to disease, such as exposure to smoking, hospitalization etc. it can also denote inherent features of the subject, like gender, and it can also signify whether a subject has received some treatment. In some cases, use of the term 'treatment' instead may be more proper, but for the sake of consistency, the term exposure will be used throughout. In practice however, the exposures becomes any feature of the subject we record, which also opens up the question of whether we have sufficient amount of information about the sample to draw any proper statistical conclusion, since an important aspect of establishing causality, is that no intervening variable is unaccounted for[1].

Ideally, we would like to examine a patient both when exposed and unexposed, to asses its effect. In practice however, we only get to observe the effect of either being exposed or unexposed. As such, we instead look at the effects over larger populations to asses general effects instead of local ones. This means we have the choice of outcome to calculate, as the effect we examine can come in 2 variations. Since the group of people exposed to some factor, might systematically differ from the rest of the population in other regards as well, we can presume the expected effect of exposing the population to this factor and the expected effect of the exposed to not always coincide. This gives rise to two outcomes we can study. The expected effected of population-wide exposure, sometimes also called treating the whole population, is defined as the average treatment effect (ATE). The expected effect for the of the exposed is defined as the average treatment effect for the treated (ATT)[5].

Formally, if we assign an index i to each person in our subject in our sample. They will have a variable, X_i indicating whether the subject has been exposed to that factor, 1 indicating exposure and 0 indicating a lack of exposure. They would also have a pair of potential outcomes, $Y_i(0)$ and $Y_i(1)$, representing the outcome if the subject was part of the control group or the treated group respectively. Thus, for each patient, we can describe the observed outcome as $Y_i = Y_i(0)(1 - X_i) + Y_i(1)Z_i$ or $Y_i(X_i)$, and the treatment effect as $Y_i(1) - Y_i(0)$. Not here that we can only observe the effects, hence the treatment effect of one patient is never observed. We can evaluate the effect of treatment in 2 ways. We can calculate the average treatment effect for the treated (ATT) as $\mathbb{E}[Y_i(1) - Y_i(0)|X_i = 1]$, and the average treatment effect as $\mathbb{E}_i[Y_i(1) - Y_i(0)]$.

The data collection process comes in different shapes and sizes, both in regards to the sample size, and the way of which that data was collected. Since the end goal is to draw some conclusion supported by statistical tests, the type of statistical analysis will vary with the method used to collect the data. An example of this, is the randomized control trial (RCT). Here we divide our sample into a treatment group and a control group, where the treatment group is the subject to exposure, while the control group gets placebo exposure. For example, if we were to test the effect of a new treatment on some set of test subjects, we would divide them into a treatment group and a control group at random. We would then observe the difference in outcome in these two groups, usually over a period of time. The inherent randomness in the exposure assignment results in some desirable properties, as exposure is not uncorrelated with any other factor. An unbiased estimate of the ATE can be calculated directly from the data, since $\mathbb{E}_i[Y_i(1) - Y_i(0)] = \mathbb{E}_i[Y_i(1)] - \mathbb{E}_i[Y_i(0)]$ from which an unbiased estimate can be calculated by simple means of the data. Inherent in this type of study, is that the exposure we wish to examine with regards to some outcome, is determined before the data collection. This makes the study expensive as data collection can often be time consuming and costly, especially if we wish to study multiple exposure factors. This study setup is experimental in the sense that we actively expose some subjects, which can cause problems. If the outcome we wish to study is very rare, numerous trails would have to be performed. Some factors are also considered highly unethical to actively expose to subjects, e.g. serious or even terminal illnesses. This method of study is thus limited by the type of exposure, since it might not be feasible or ethical to study the relation of exposures and outcomes in this manner.

Since the central problem with the RCT is the exposure, one could study the outcome of some group already subject to the exposure of interest, and follow them for a period of time, to observe the outcome. The risk of outcomes would then be compared to a control group of the unexposed people from the population. This type of study is called a prospective cohort study, or follow-up cohort study, where cohort refers to a population. This type of study closely relates to the RCT, but instead of the experimental approach, we instead use an observational one, hence this type of study is called an observational study. Because of the observational nature, it makes no difference if the outcome we are monitoring has happened or not, given we can still measure it accordingly. This is the method used in the retroactive cohort study, which studies the relation of exposure and outcome that has already happened in some cohort of subjects. Since the assignment of exposure is no longer random, like the RCT, we lose the ability to make an unbiased estimate of the ATE. This is also why we instead talk about risks instead of average treatment effect, more specifically, we are most interested in the relative risk $\frac{P(O=1|X=1)}{P(O=1|X=0)}$ where O is a variable indicating whether the outcome occurs. Note here that the outcome variable has been restricted to be binary, indicating that we need to formulate the outcomes of interest as categorical, whereas the RCT also allowed for continuous outcomes. This also means that we can represent the data in a contingency table

	Exposed	**Unexposed**
Outcome	EO	UO
No outcome	EN	UN

From this, an unbiased estimate of the relative risk can be calculated by $\frac{EO/(EN+EO)}{UO/(UN+UO)}$.

The focus of the retrospective cohort study, is to study a certain exposure w.r.t. how it affects some outcome. If we instead wish to study the risk of an outcome w.r.t. some exposures, we enter the domain of the case-control study. Like the retrospective cohort study, the case-control study happens after both the exposure and the health outcome has happened, and is likewise also an observational study method. The difference comes in what data is available to study. In the cohort study, we have the data for the complete cohort, which can be expensive to obtain in practice. In the case control

study we work with the people who has the outcome of interest, called the cases, and some people who do not, called the controls, which are sampled from the same population as the cases. This method of study grants some advantages. Among others we shall shortly see, it is cheaper to conduct as we do not require data on the whole cohort, but just the sampled cases and controls. There are also some drawbacks for the case control study compared to the retrospective cohort study. Firstly, the controls and cases need to be representative of the same groups in the general population. The drawback comes directly from the advantage of the case control, namely that we do not have data on the whole cohort, so we have to sample some representative subjects from the cohort. The problem of finding the representative outcome groups, as well as adjusting for differences, has a long history, as will be discussed later. Another drawback of using the case control study, also stems from the fact that we no longer have the exposure rate for the complete cohort, which inhibits the calculation of relative risk as it requires that we know the exposure rate for both outcome groups.

Various ways of deriving conclusion from case control studies, but matching the cases with controls ensures the best similarity of exposures, thus giving better results[2]. There is two general methods for calculating the similarities used to do matching, covariate matching and propensity matching[20]. In covariate matching we define 'closeness' based on observed baseline covariates $\mathbf{B}_i$, thus similarity on multiple covariates can be ensured. In propensity matching we calculate a propensity score, which is the probability of exposure given observed baseline covariates, $P(O_i = 1|\mathbf{B}_i)$, which under certain assumptions, can be used as a balancing score to mimic properties of a RCT[2]. In both cases we match cases with subjects from the control group with similar scores.

The main drawback of the propensity score method is that it is based on the value of parameterized distribution, which means the matching is done from an aggregate of the observed covariates. In covariate matching, this is circumvented by directly using the covariates for comparison, but this poses practical challenges as it may be hard to find samples that are close enough to a case in regards to all the observed covariates.

Scope of this project

In this project, we will try to learn a representation of the covariates using the variational autoencoder architecture. The purpose of the representations will be purely exemplary, to examine the usefulness through general metrics and general tasks, and thus are not suited by design to any specific case control study matching. By doing covariate matching on the learned representation, we aim to create a matching method that takes all observed covariates into account, as well as gives a more robust notion of similarity through distance.

The evaluation of the learned representation aim to examine the fundamental properties of encoded health data, namely, how much information is preserved, and what the structure of the learned representation space is? As seen in figures 1 and 2 we examine these by methods like learning models for reconstructing observed covariates or highly correlated values as well as examining the representation space visually.

Related work

Learning representations of electronic health records has seen approaches using deep learning[19]. Deep learning models are well suited to this task as they are inherent feature learners[3]. The problem has been approached in a varied number of ways. The deep learning method applied is typically tied to the type of data the EHR are represented in. Deep learning architectures like the recurrent neural network and its variants like LSTM and GRU have been applied to sequence-based patient representation from sequences of EHRs. Fully connected deep neural networks, convolutional networks, as well as the word2vec[7] algorithm has been applied to learn single vector representations from feature vectors, feature images, and full EHR sequences respectively.

The problem has most recently been approached using the transformer model [23], training a transformer model on a large corpus to learn general contextualized embeddings of EHR[18], which can be fine-tuned for specific tasks with good results[13]. Our representations will be learned using a variational autoencoder[10] on pre-processed snapshots of a patients state, consisting of the latest recorded values created each 24 hour cycle.

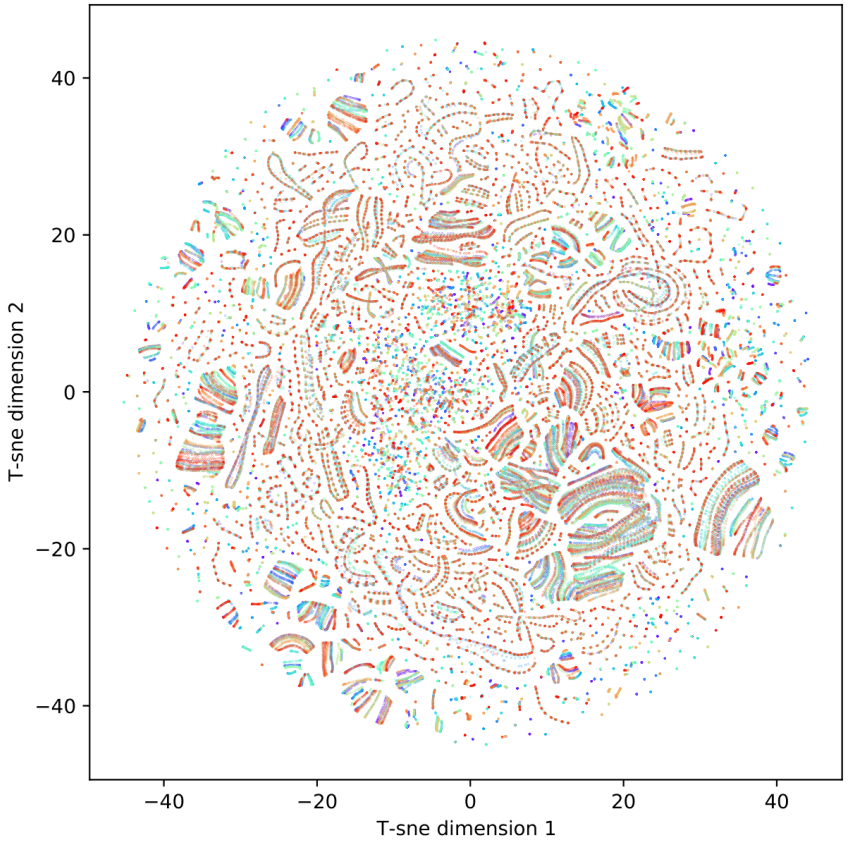


Figure 1: 223185 snapshots of EHR from 8367 patients encoded by a VAE with $dim(z) = 20$, visualized using t-SNE, colored by patient ID.

Introduction

A common problem of epidemiological case control studies is that of matching the cases with the controls[5]. The matching problem has two general solutions, one is matching on a calculated propensity score, and the other is matching on the observed covariates[20]. We explore learning a representation meant for matching, such that the matching is done w.r.t. every observed covariate. Learning representations of EHR data has seen work[19], with the latest work being in general contextualized embeddings using transformers [18, 13]. We use the variational autoencoder model to learn a latent representation of a snapshot dataset of EHR from SARS-CoV-2 positive patients and evaluate the amount of information preserved from the features used to encode, as well as highly correlated features not used directly in the representation. We also perform qualitative analyses of the structure of the learned representations to asses whether distance-based matching is feasible.

Method

Data source and modality

We will use electronic health records (EHR) from citizens in the danish capital and Zealand regions, whose populations totals approximately 2.5 million people, filtered down to people whom have tested positive for SARS-CoV-2 between March 3rd, 2020 and May 26th, 2021. For these patients, their health data including age, gender, smoking status, BMI, and diagnoses were pulled, as well as temporal data describing the time of tests, hospital and ICU admissions, usage of ventilator, lab tests performed, medication administered, and death where present. The data is structured as a sequence of EHR, sorted by time, from the time of a positive SARS-CoV-2 test until either death occurs, there is no record of hospitalization in 30 days, or the dataset was extracted.

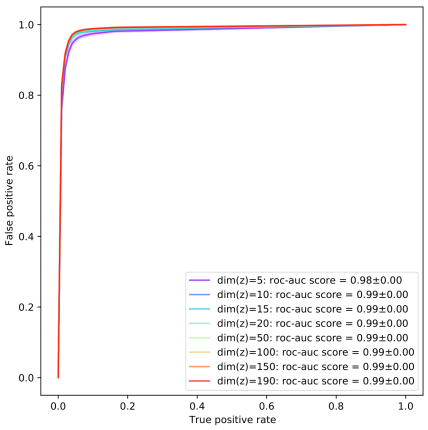


Figure 2: ROC-curves of random forest classifiers trained to predict whether a patient will be hospitalized 5 days in the future, each trained on encodings of various dimensionalities.

The dataset was constructed as snapshots for each 24 hour interval for each patient. The snapshots contain the latest updated information for each feature, as well as information about that patients status (has SARS-CoV-2, hospitalized, in intensive care unit (ICU), in ventilator, or dead) n days in the future, where $n = 1..15$. The features containing information about the patients future, will be referred to as target values, because of the inclination to use the data in the snapshot to predict the state of the patient n days in the future.

The final dataset consists of 223185 snapshots of 8367 unique patients. Each snapshot contains 190 values of health record information and 75 target values. Missing values are filled with 0, except for BMI, which is imputed by nearest neighbor on age and sex with $k = 100$.

Information theory

Throughout this report certain concepts in information theory will be used to motivate conclusions and hypotheses, these concepts will briefly be introduced in this section.

Firstly, we introduce the stochastic variable X, which has a distribution $p(x), x \in \mathcal{X}$ where $\mathcal{X}$ is the sample space of X. If we wish to communicate some outcomes of X through some channel, also known as the entropy of p, we can calculate the average number of information units needed to express the outcome as

$$H[X] = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \quad \text{[4, equation 1.93]}$$

where the base of the logarithm will determine the base of the logarithm. If we choose base 2 the information unit becomes bits, while if we choose the natural logarithm, we get 'nats' as the unit of information. Note here that we assume X to be discrete, when working with continuous distributions, we would integrate instead of summing, which would also allow for the calculated entropy to be negative, which doesn't agree with the interpretation of entropy. The properties of information theoretical concepts thus differ from discrete and continuous probability spaces.

Entropy of a distribution also extends to multivariate distributions, where the entropy of a two dimensional distribution $p(x, y)$ can be calculated as the amount of information needed to encode x, and the amount of information needed to encode y given x

$$H[X, Y] = H[X] + H[Y|X] \text{ [4, equation 1.112]}$$

This introduces the term $H[y|x]$ which is called the conditional entropy of y given x and is calculated as

$$H[Y|X] = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log p(y|x) \text{ [4, equation 1.111]}$$

Where $\mathcal{Y}$ is the sample space of Y.

Now we introduce relative entropy, which is defined as the additional amount of information needed to transmit values of X if we use an approximate distribution $q(x)$ instead of $p(x)$. The relative entropy is also called the Kullback-Liebler divergence or KL divergence and is calculated as

$$KL(q||p) = \sum_{x \in \mathcal{X}} p(x) \log \left(\frac{q(x)}{p(x)} \right) \text{ [4, equation 1.113]}$$

Some properties of the KL divergence are

$$KL(p||q) \geq 0$$
$$KL(p||p) = 0$$

Which makes it a good tool to asses the closeness between two distributions, but it is only a pseudo-metric as it is not symmetric.

$$KL(p||q) \neq KL(q||p) \text{ if } q(x) \neq p(x)$$

Given two stochastic variables X and Y, we can use relative entropy to construct a measure of how much information one gives about the other. This is called mutual information, $I(X, Y)$, and is calculated as

$$I(X;Y) = KL\left(p(x,y)||p(x)p(y)\right) \text{ [4, equation 1.120]}$$

This expression can be rewritten to

$$I(X;Y) = H[X] - H[X|Y] = H[Y] - H[Y|X]$$

Another common setting in information theory is that of information compression, for which Tishby et al.[21] introduced a framework for calculating the number of information units needed to represent the data, given we know the maximum loss of information in the compression. If we compress X to some encoding $\widetilde{X}$, we can examine the information bottleneck of X and some variable Y, which is the amount of information X provides about Y when X is compressed through the 'bottleneck' $\widetilde{X}$. Based on the notion that more compression will lead to a larger loss of information, the optimal compression assignment, $p(\widetilde{x}|x)$ can be found by minimizing the functional

$$\mathcal{L}[p(\widetilde{x}|x)] = I(\widetilde{X};X) - \beta I(\widetilde{X};Y) \text{ [21, equation 15]}$$

where β is a constraint of how much relevant information is to be preserved in the compression. Note that when $\beta = 0$, the compression will not preserve any information that X provides about Y, while when $\beta \to \infty$ no information will be lost. The explicit formula for $p(\widetilde{x}|x)$ is

$$p(\widetilde{x}|x) = p(\widetilde{x}) \frac{\exp[-\beta KL(p(y|x)||p(y|\widetilde{x})]}{\sum_{\widetilde{x}} \exp\left[-\beta KL(p(y|x)||p(y|\widetilde{x})\right]} \text{ [21, equation 28]}$$

The penalty for loss of information here is thus the dissimilarity between the probability of Y conditioned X and $\hat{X}$ respectively, weighted by β.

Calculating the quantities when the variables are continuous are intractable, in addition to the aforementioned possibility of negative entropies, discretization is needed to numerically solve the integrals, making calculations very sensitive to hyperparameters. While multiple methods of estimating mutual information has been proposed[8, 25, 12], no reliable implementation of such methods were found for this project.

Variational autoencoder

The variational autoencoder (VAE) was first proposed by Kingma et al.[10] and is a unsupervised deep learning architecture that aims to learn a compressed representation z for our samples x where $dim(z) \leq dim(x)$, where the distribution of the latent space $p_\theta(x)$, is made to approximate some pre-defined prior $p(z)$

VAEs consists of an encoder and a decoder, both of whom are neural networks. The encoder takes a sample from our dataset and encodes a distribution of its representation in the latent space Z, a

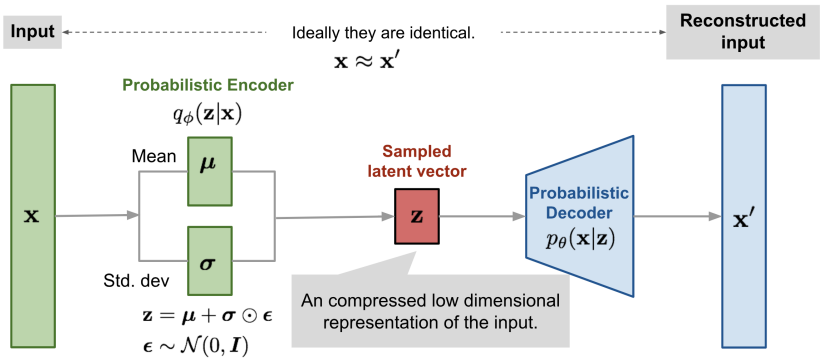


Figure 3: Variational autoencoder with Gaussian prior, illustration by Lilian Wang[24]

sample is then drawn from that distribution and used as input in the decoder, which then reproduces the original sample. This raises 2 distributions, the distribution of samples in the latent space $q(z|x)$, and the distribution over the reconstructions $p(x|z)$. A restriction is put on $q_\theta(z|x)$ to approximate the prior $p(z)$ by adding $KL(q_\theta(z|x)||p(z))$, also called the regularization loss, to the reconstruction loss. This loss function corresponds to the evidence lower bound, which won't be explored in this project, but can be explored in [11].

The natural choice of prior is a k-dimensional random variable of independent Gaussians, that is, $q_\theta(z|x) = \mathcal{N}(z; \mu, \text{diag}(\Sigma))$ where μ and Σ are the resulting vectors of the encoder neural network and diag transforms a vector into a diagonal matrix via $\text{diag}(x) = I \odot (x\mathbf{1}^T)$. An illustration of this setup can be seen in figure 3. Assuming this prior, as well as assuming $p(x|z) \sim \mathcal{N}(D(z), I)$ where $D(z)$ is the predicted reconstruction of the decoder, the loss simplifies to

$$\mathcal{L}_{VAE} = \overbrace{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}^{\text{Reconstruction loss}} + \overbrace{\frac{1}{2}\left[\log\left(\det(\Sigma)\right) - k + \text{tr}\left(\Sigma\right) + \mu^T\mu\right]}^{\text{Regularization loss}}$$

The full derivation of the loss function can be found in appendix A.

As both the encoder and decoder are neural networks, the sampling of z should allow for the back-propagation to flow though to the encoder. This is accomplished by the reparameterization trick by instead sampling $\epsilon \sim \mathcal{N}(0, I)$, and calculating $z = \mu + \Sigma \odot \epsilon$, still allowing the gradient to flow through the calculations of μ and Σ.

Quantitative evaluation of encodings

The evaluations of the encodings produced should reflect the task for which these encodings are to be used. The purpose of our encodings are to examine the amount of information preserved, and their possible usefulness as general purpose encoding of EHR. The evaluation of our encodings will thus be both through the reconstruction loss, as a direct indicator of the amount of information preserved, as well as the ability of machine learning models to recreate features from the encodings, to asses whether parts of the decoding process can be learned.

The first part of the evaluations will start by examining the trade off between having a lower dimensional encoding and preserving more information. The loss of information will be examined using the reconstruction loss of the models, sorted by the size of their respective latent space. Since we can only lose information by decreasing the latent-space of the models, we expect the reconstruction loss to increase as the size of the encoding space decreases. This will be supplemented by a theoretical upper bound on the reconstruction loss for a fully trained model, corresponding to a model with a 0-dimensional latent space, which guesses the global mean value of each feature.

The loss of information will be examined on the basis of the lens of the information bottleneck method, but with two major differences. Firstly, our encodings are produced by a trained network,

with the number of bits used to represented the data predefined through the predefined dimensionality of the latent space. This means that instead of choosing how much information should at least be preserved through β and then finding the minimum number of bits needed, we instead do the reverse and set some amount of bits and then maximize the amount of relevant information stored in representation using that amount of bits. The other difference is that we use surrogate measures instead of mutual information, as it is not practical to calculate. These surrogate measures will include the reconstruction loss of the variational autoencoder as a direct replacement of mutual information, both the total and feature-wise, and the ability to reproduce specific parts of the original input from the encodings by learning simple machine learning models. The preserved correlation of the data and some features un-used in the encoding will be examined though the methodology.

Because of the stochastic nature of the VAE as well as the stochasticity of random initialization, we expect the reconstruction loss to have potentially large variance. To evaluate an architectures ability to preserve information, multiple models will be trained independently for which we consider the reconstruction loss of the last 50 training epochs, and then average out the losses over the epochs and then the training runs. This will minimize the variance from both the stochastic layer in the VAE, as well as the stochasticity of the random initialization of the models.

The evaluation the ability to learn piece-wise reconstruction of the original samples from encoded data, will fall into 2 parts, one for the binary features and one for the temporal features of the original samples. For the binary features, we will evaluate a logistic regression model using receiving operating characteristic (ROC), where we will look at both the ROC-curve and the area under curve (ROC-AUC). For the continuous features, we use a simple Lin regression model and evaluate it using mean squared error (MSE) and the coefficient of determination $\left(R^2\right)$. The evaluation will be done on held-out data because performance on the training data will be biased, since the targets have directly influenced the encodings we use to predict them.

Lastly, to asses how well correlation between encoding and non-encoding features are preserved, the ability to guess the values of the target variables given the encodings will be examined. This evaluation will be done in the same manner as the binary encoding features, since all of the target variables are also binary.

We use sklearn [17], a machine learning library for python, for the implementations of the logistic regression model, linear regression model, and random forest model, and we use the XGBoost python library[6] for an implementation of the xgboost algorithm.

Qualitative examination of encodings

To asses the structure of the preserved information, qualitative examinations of the encoding spaces will be performed. Data that are similar are expected to be close in the embedded space per the properties of VAE, thus we can expect this in the encoded dataset. Thus we can visualize the data colored by value of features in the original samples, e.g. their gender, whether they have some measured illness or condition etc. For the purpose of visualization, dimensionality reduction methods will be used to provide 2-dimensional images of the data in the encoded space. The preferred method will be t-distributed stochastic neighbor embeddings (T-SNE) [22] because it preserves local clusters, but it will be contrasted to principal component analysis (PCA) [16] for illustrative purposes. The python library sklearn will also be used for implementation of PCA and T-SNE. Here we expect a well-behaved encoded space to separate the data w.r.t. a feature, more clearly the more information about that feature is preserved in the encoding. Thus we expect the separation of samples categorized by a feature to be clearer the higher dimensionality of the latent space.

Because the dataset consists of multiple snapshots of the same patient, we are left with multiple groups of very similar data. The placement of samples from individual patients will also be examined in the projected space produced by T-SNE, as well as the how the patients snapshots move in the encoded space over time depending on their course of disease.

Experimental setup

Our experiments will consist of evaluating 8 different variational autoencoder architectures, each trained with 10 random initializations. All of the models will contain encoders and decoders with 3 fully connected linear layers, utilizing the non-linear function ReLU(x) = max(x,0) as proposed in

[14]. For reconstruction loss, we utilize mean squared error (MSE), and weight updates will be done using the Adam optimizer[9] with a learning rate of $\gamma = 0.001$, beta values of $\beta_1 = 0.9, \beta_2 = 0.999$, and an epsilon value of $\epsilon = 1\text{e}{-8}$. The β_1, β_2, and ϵ parameters was taken as the standard parameters of the PyTorch[15] implementation of Adam, the learning rate γ was tested and showed good results, and was thus selected.
We use open-source PyTorch framework[15] for implementing and training the models. The full model descriptions can be found in appendix B

Results

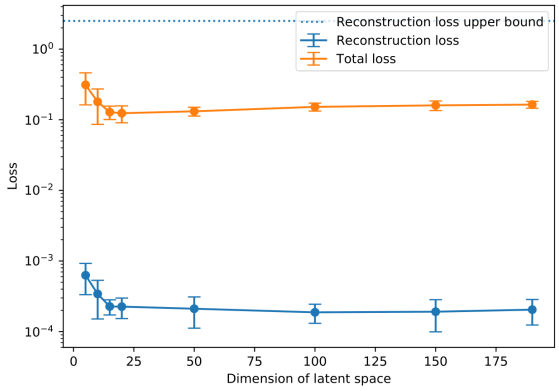
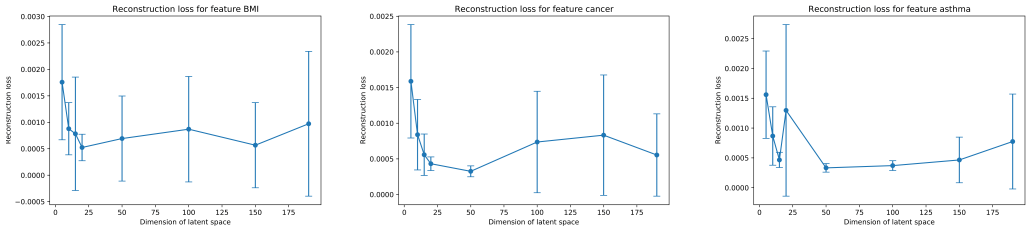
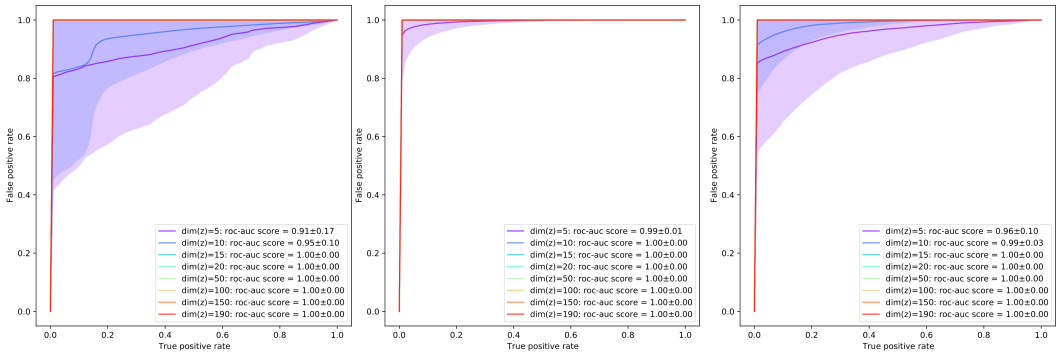


Figure 4: Reconstruction loss and total loss for VAEs trained on 223185 snapshots of EHR from 8367 patients with different sized latent spaces as well as the theoretical upper bound for reconstruction loss.



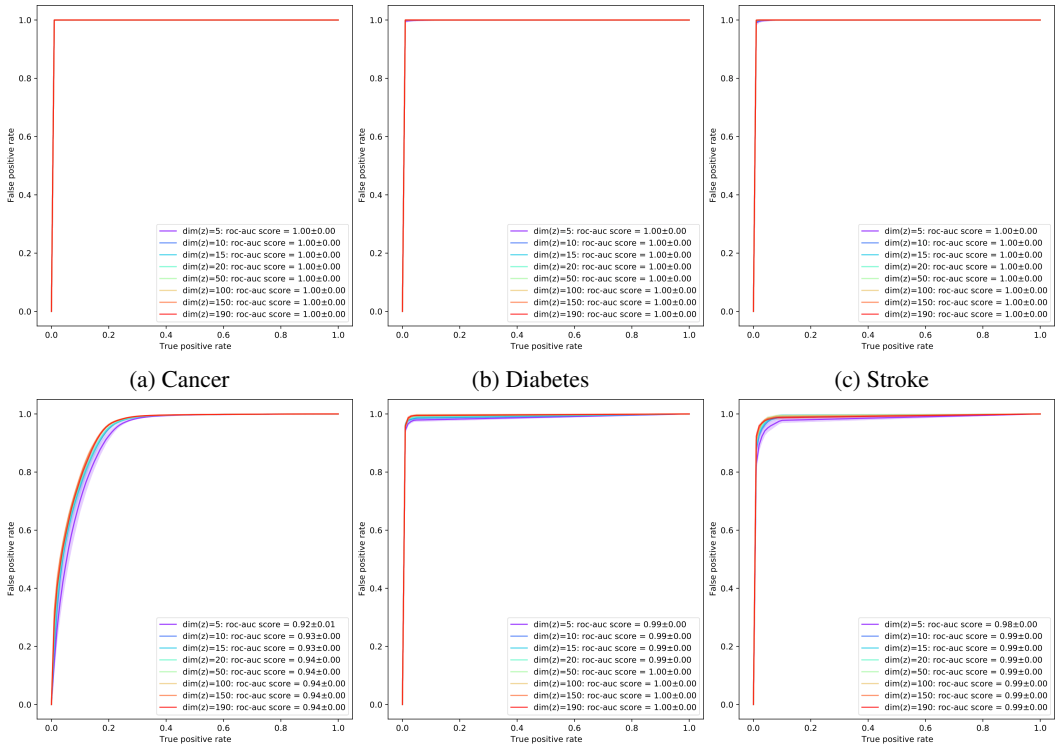
(a) Body Mass Index (BMI) (b) Cancer (c) Asthma

Figure 5: Reconstruction loss for selected features averaged for VAEs with different latent sizes. The loss is the average of 10 training iterations on 223185 snapshots of EHR from 8367 patients.



(a) Logistic regression (b) Random Forest (c) XGBoost

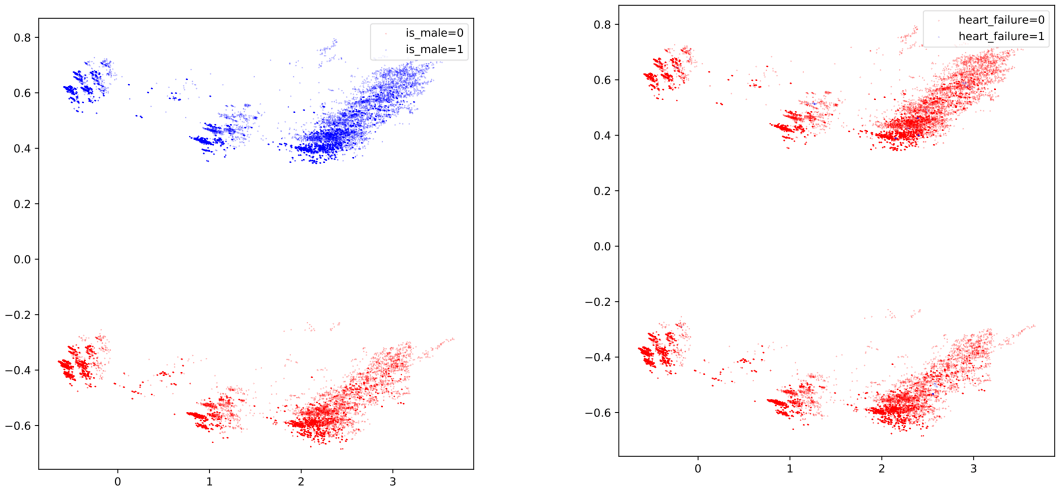
Figure 6: ROC-curves for 3 different types of classifier models for predicting gender feature of original sample from VAE latent representation. Each type of classifier is trained and evaluated for $dim(z) = 5, 10, 15, 20, 50, 100, 150, 190$.



(a) Cancer (b) Diabetes (c) Stroke

(d) Covid in 5 days (e) In ventilator in 5 days (f) Dead in 5 days

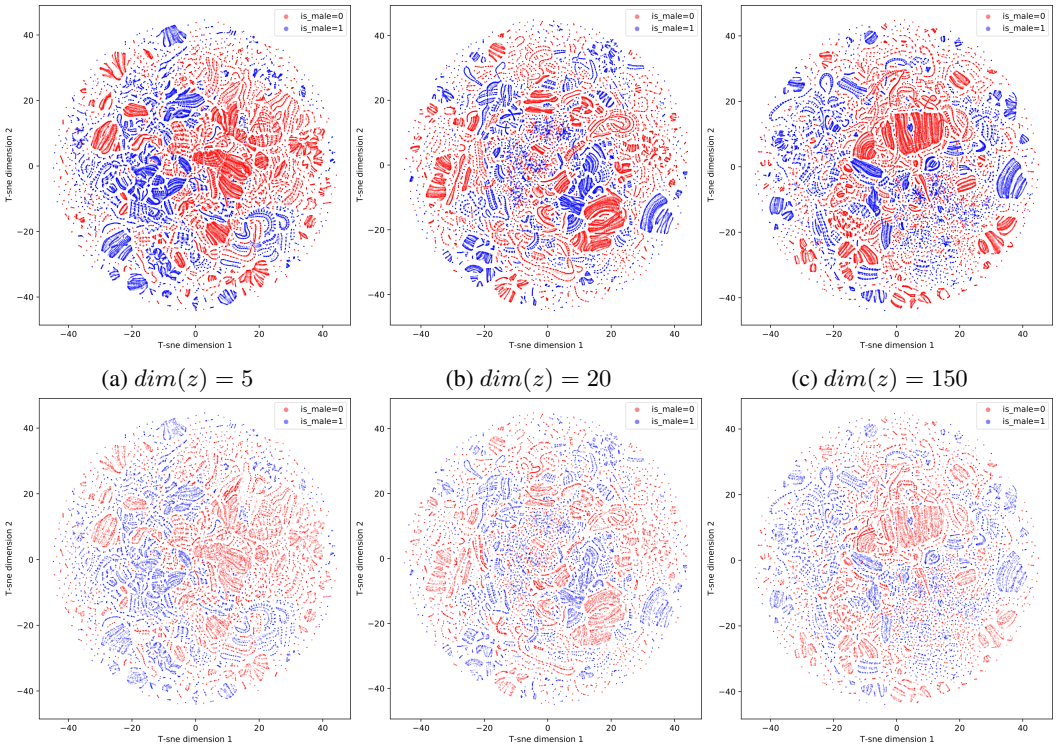
Figure 7: ROC-curves for random forest models predicting a specific feature of original sample from VAE latent representation. For each $dim(z) = 5, 10, 15, 20, 50, 100, 150, 190$ the ROC curve is a linear interpolation of the ROC curves of 10 training iterations.



(a) Colored by gender, blue for male, red for female.

(b) Colored by whether the patient has had heart failure at any point before the time of snapshot, blue for male, red for female.

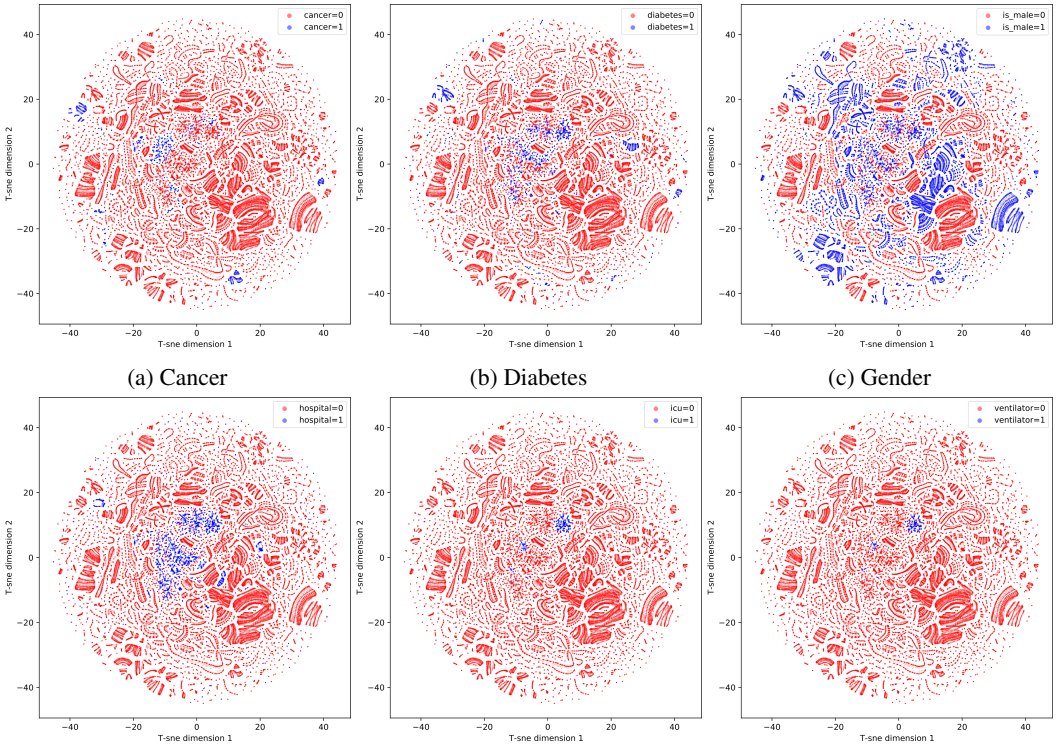
Figure 8: PCA reduced encoded space for 223185 snapshots of EHR from 8367 patients using encodings from a model with $dim(z) = 20$.



(a) $dim(z) = 5$ (b) $dim(z) = 20$ (c) $dim(z) = 150$

(d) $dim(z) = 5$ (e) $dim(z) = 20$ (f) $dim(z) = 150$

Figure 9: t-SNE visualizations of encodings of 223185 snapshots of EHR from 8367 patients encoded using models with $dim(z) = 5, 20, 150$. (a), (b), and (c) contain the training samples while (d), (e), and (f) contain the test set.



(a) Cancer (b) Diabetes (c) Gender

(d) In hospital (e) In intensive care unit (ICU) (f) In ventilator

Figure 10: t-SNE visualizations of encodings of 223185 snapshots of EHR from 8367 patients, encoded by VAE with $dim(z) = 20$, colored by value of selected features.

Discussion

Quantitative evaluation

We see in figure 4 that the VAE is able to do better reconstructions the more information is able to be preserved in the latent space, from the increase in number of dimensions. The large distance between the theoretical upper limit of the reconstruction loss, and the reconstruction loss of all the models, including the small dimensional model, show that we can construct a small set of variables that capture the correlations in the original samples. The decrease in reconstruction loss becomes minimal after encoding into a 15-dimensional latent space, which points to the fact that the 190 original features are very well captured in just a 15-dimensional learned representation. It is also apparent that the reconstruction loss and the total loss are shifted by some constant-like factor for all the models. This constant factor is the regularization loss, which by being very close among all the models, means that all models learned representations which approximate the assumed prior which consists of independent Gaussians. The divergence of the reconstruction loss curves in figure 5 from the reconstruction loss curve in 4, and the high variance across different models of the same architecture, implies that even though the overall reconstructions become better, the different latent representation of different models, even with the same architectures, prioritize preserving different features.

A comparison of the 3 different models trained to recreate single features from the encodings seen in figure 6 imply that models can be trained to recreate parts of the decoder. The good performance of XGBoost and even better performance of random forest models compared to the logistic regression, imply that the decoding process is non-linear. From figure 7 we see that features used in the encoding are easy to predict form the encodings, and that the correlation to the target features are also preserved in the latent representation. We also see that the dimensionality of the latent space has little effect on the ability to reconstruct the single features, as it does with the reconstruction loss, implying that a larger decoder with more and larger layers might allow for smaller latent representations.

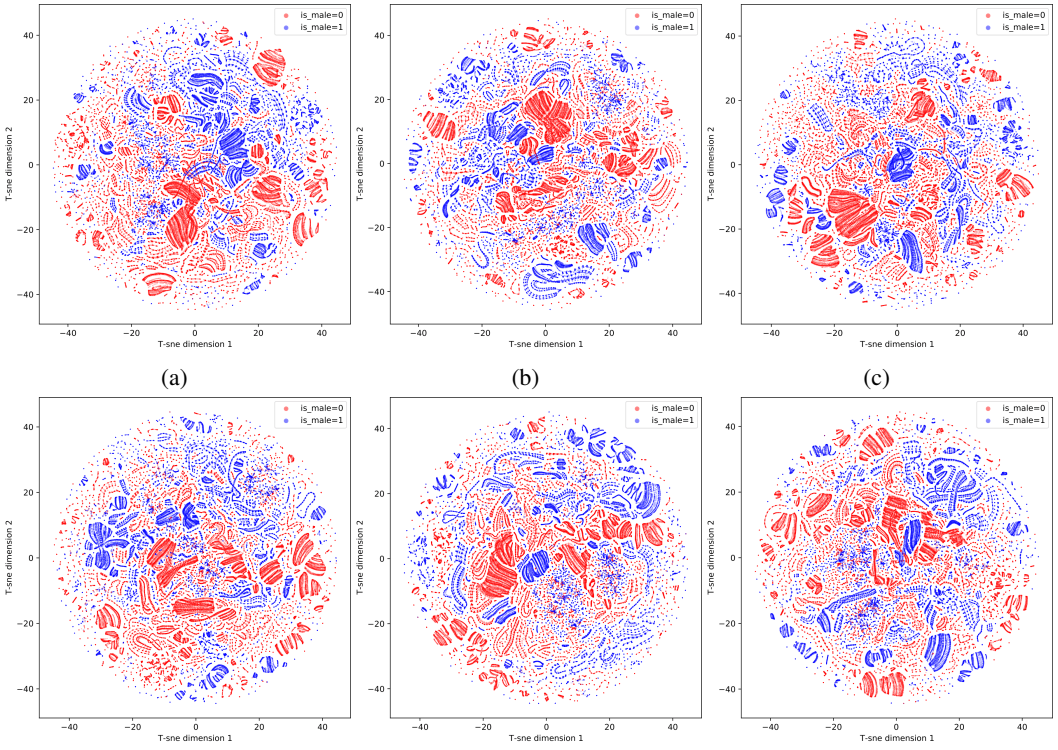


Figure 11: t-SNE visualizations of 6 different encodings of 223185 snapshots of EHR from 8367 patients, encoded by VAEs with $dim(z) = 50$ trained with 6 different random weight initializations

Qualitative evaluation

When comparing PCA and t-SNE for dimensionality reduction for visualization, we see in figure 8 that preserving the global variance divides that encoded space by gender first and foremost, while preserving local variances with high probability provides a more clustered visualization space as seen in figure 10. The choice to use t-SNE gives the ability to more easily examine how features are represented.

When a representation space successfully encode a feature, we would expect the feature to exhibit a structure in the representation space. We can confirm visually from figure 10 that the representation spaces preserves information about encoding features. The features show as multiple clusters in the representation space, supported both by the superior performance of non-linear classifiers in figure 6 and the multiple clusters in the t-SNE visualizations.

The dataset consists of multiple snapshots of the same patient over time, breaking the i.i.d assumption of learning algorithms. This lack of independency presents visually in the encodings as snapshots of the same patient can appear as a line in the t-SNE visualization as seen in figure 1.

From figure 9 we see that the learned representations generalize to unseen data, showing that the learned mapping is indeed a representation, and not a memorization of the training data. The learned representation, however, can vary for different models. As seen in figure 9 the dimensionality of latent space allows for easier separation of groups of patients, as is expected since more information is preserved in the encodings. The representation also vary across different trained instances of the same model as seen in figure 11, where we see some general patterns across the different models. But, as previously discussed and seen in figure 5, trained instances of the same model might learn different correlation structures and thus different representations. Dissimilarities arrive both from these differing representations, as well as the random permutation of learned features, as is common in deep learning, which causes the rotation, flipping and stretching of the recurring structures in figure 11.

Conclusion

Information from EHR processed into snapshots can be compressed quite a lot while still preserving important information, measured using reconstruction. We've also seen that reproducing original covariates can be accomplished using non-linear machine learning models, even when the dimensionality of the representation space becomes very low, even when the total reconstruction loss begins to increase. We also see that the structure of the representation spaces, although robust in the sense that samples are separated by the value of the features, can differ on the same model, trained with different random initializations. This difference can in part be explained by the random scaling, shifting, and permutations of learned features in deep learning, but is also in part because each training instance will learn to represent and reconstruct features differently. The measures used are notions of amount of information preserved, because more the analytical measure of the information bottleneck was impractical in this setting.

Although the representations were not directly used for case control studies, we have demonstrated that sufficient information can be preserved in low dimensional learned representations, making matching on multiple covariates more feasible. To more accurately asses this method of case control matching, it should be tested in a epidemiological case control study setting.

The goal of making global re-usable representations is limited by the need for no intervening variable to be unaccounted for, meaning that the global representation is insufficient for studying cause and effect relationships that may be confounded by variables not present in the representation. Meaning that even though these representation could be used for case control studies, they are always limited by the variables observed.

References

[1] J. Antonakis, S. Bendahan, P. Jacquart, and R. Lalive. On making causal claims: A review and recommendations. *The leadership quarterly*, 21(6):1086–1120, 2010.

[2] P. C. Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424, 2011.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[4] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[5] N. E. Breslow. Statistics in epidemiology: the case-control study. *Journal of the American Statistical Association*, 91(433):14–28, 1996.

[6] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[7] K. W. Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[8] W. Gao, S. Kannan, S. Oh, and P. Viswanath. Estimating mutual information for discrete-continuous mixtures. *Advances in neural information processing systems*, 30, 2017.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[11] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

[12] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[13] F. Li, Y. Jin, W. Liu, B. P. S. Rawat, P. Cai, H. Yu, et al. Fine-tuning bidirectional encoder representations from transformers (bert)–based models on large-scale electronic health record notes: an empirical study. *JMIR medical informatics*, 7(3):e14830, 2019.

[14] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[16] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[18] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):1–13, 2021.

[19] Y. Si, J. Du, Z. Li, X. Jiang, T. Miller, F. Wang, W. J. Zheng, and K. Roberts. Deep representation learning of patient data from electronic health records (ehr): A systematic review. *Journal of Biomedical Informatics*, 115:103671, 2021.

[20] E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

[21] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[22] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[24] L. Weng. From autoencoder to beta-vae. `https://lilianweng.github.io/posts/2018-08-12-vae/`. Accessed: 2022-06-08.

[25] Z. Zhang. Estimating mutual information via kolmogorov distance. *IEEE Transactions on Information Theory*, 53(9):3280–3282, 2007.

A Derivation of the VAE loss function

We start by considering the VAE loss

$$\mathcal{L}_{VAE} = \mathbb{E}_{z\sim q_\theta}[-\log p(x|z)] + KL(q_\theta(z|x)||p(z))$$

We start by showing that the reconstruction loss can be written as the mean squared error $\mathbb{E}_{z\sim q_\theta}[-\log p(x|z)] = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x_i})^2$. We assume that $x|z \sim \mathcal{N}(D(z), I)$ where the predicted reconstruction $D(z)$ is assumed as the distribution mean and the variance and dimensionality $dim(z)$ are assumed to be constant.

$$\begin{aligned} x|z &\sim \mathcal{N}(D(z), I) \\ \log p(x|z) &= \log \frac{\exp\left(-\frac{1}{2}(x - D(z))^T\Sigma^{-1}(x - D(z))\right)}{\sqrt{(2\pi)^{dim(z)}|\Sigma|}} \\ &= \left(-\frac{1}{2}(x - D(z))^T\Sigma^{-1}(x - D(z)\right) - \log\left[\sqrt{(2\pi)^{dim(z)}|\Sigma|}\right] \end{aligned}$$

Because of our assumptions we see that we can minimize this expression by minimizing the term $-(x - D(z))^2 = -(x - \hat{x})^2$. Thus the reconstruction loss becomes

$$\begin{aligned} \mathcal{L}_{reconstruction} &= \mathbb{E}_{z\sim q_\theta}[-\log p(x|z)] \\ &= \mathbb{E}_{z\sim q_\theta}\left[(x - \hat{x})^2\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x_i})^2 \end{aligned}$$

Since $\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x_i})^2$ is an unbiased estimate of $\mathbb{E}_{z\sim q_\theta}\left[(x - \hat{x})^2\right]$.

We now show that the regularization term of the variational loss can be rewritten as

$$KL(q_\theta(z|x)||p(z)) = \frac{1}{2}\left[\log\left(\det(\Sigma_\theta(x))\right) - k + \text{tr}\left(\Sigma_\theta(x)\right) + (\mu_\theta(x))^T\mu_\theta(x)\right]$$

We start by considering any

$$\begin{aligned} q_\theta(z|x) &= \frac{\exp\left(-\frac{1}{2}(z - \mu_\theta(x))^T\Sigma_\theta(x)^{-1}(z - \mu_\theta(x)\right)}{\sqrt{(2\pi)^{dim(z)}|\Sigma_\theta(x)|}} \\ p(z) &= \frac{\exp\left(-\frac{1}{2}(z - \mu_0)^T\Sigma_0^{-1}(z - \mu_0\right)}{\sqrt{(2\pi)^{dim(z)}|\Sigma_0|}} \end{aligned}$$

We then set $\Sigma_0 = I$ and $\mu_0 = 0$ as per the choice of prior. The Kullback-Liebler divergence can now be written up as

$$\begin{aligned} &KL(q_\theta(z|x)||p(z)) = KL(\mathcal{N}(\mu_\theta(x)), \Sigma_\theta(x))||\mathcal{N}(0, I)) \\ &= \int\left[\frac{1}{2}\log\left(\frac{\det I}{\det\Sigma_\theta(x)}\right) - \frac{1}{2}(x - \mu_\theta(x))^T\Sigma_\theta^{-1}(x)(x - \mu_\theta(x)) + \frac{1}{2}x^Tx\right] \times p(x)dx \\ &= \frac{1}{2}\log\det\Sigma_\theta(x) - \frac{1}{2}\text{tr}\ \left(\mathbb{E}[(x - \mu_\theta(x))(x - \mu_\theta(x))^T]\ \Sigma_\theta^{-1}(x)\right) + \frac{1}{2}\mathbb{E}[x^Tx] \\ &= \frac{1}{2}\log\det\Sigma_\theta(x) - \frac{1}{2}\text{tr}\ (I) + \frac{1}{2}\mu_\theta(x)^T\Sigma_\theta^{-1}(x)\mu_\theta(x) + \frac{1}{2}\text{tr}\left(\Sigma_\theta(x)\right) \\ &= \frac{1}{2}\left[\log\det\Sigma_\theta(x) - k + \text{tr}\left(\Sigma_\theta(x)\right) + (-\mu_\theta(x))^T(-\mu_\theta(x))\right] \\ &= \frac{1}{2}\left[\log\det\Sigma_\theta(x) - k + \text{tr}\left(\Sigma_\theta(x)\right) + \mu_\theta(x)^T\mu_0\right] \end{aligned}$$

Thus the regularization loss becomes

$$\begin{aligned} \mathcal{L}_{regularization} &= KL(q_\theta(z|x)||p(z)) \\ &= \frac{1}{2}\left[\log\det\Sigma_\theta(x) - k + \text{tr}\left(\Sigma_\theta(x)\right) + \mu_\theta(x)^T\mu_0\right] \end{aligned}$$

Thus giving the final loss expression

$$\begin{aligned}\mathcal{L}_{VAE}(x, \hat{x}) &= \mathcal{L}_{reconstruction} + \mathcal{L}_{regularization} \\ &= \mathbb{E}_{z\sim q}[-\log p(x|z)] + KL(q(z|x)||p(z)) \\ &= \sum_{i=1}^{N}(x_i - \hat{x}_i)^2 + \frac{1}{2}\left[\log\left(\det(\Sigma)\right) - k + \text{tr}\left(\Sigma\right) + \mu^T\mu_\theta(x)\right]\end{aligned}$$

B Model architectures

Model with $dim(z) = 5$

Layer (type)	Output Shape	Param #
Linear-1	[-1, 100]	19,100
LeakyReLU-2	[-1, 100]	0
Linear-3	[-1, 50]	5,050
LeakyReLU-4	[-1, 50]	0
Linear-5	[-1, 5]	255
Linear-6	[-1, 5]	255
Linear-7	[-1, 50]	300
LeakyReLU-8	[-1, 50]	0
Linear-9	[-1, 100]	5,100
LeakyReLU-10	[-1, 100]	0
Linear-11	[-1, 190]	19,190
Sigmoid-12	[-1, 190]	0

Model with $dim(z) = 10$

Layer (type)	Output Shape	Param #
Linear-1	[-1, 100]	19,100
LeakyReLU-2	[-1, 100]	0
Linear-3	[-1, 50]	5,050
LeakyReLU-4	[-1, 50]	0
Linear-5	[-1, 10]	510
Linear-6	[-1, 10]	510
Linear-7	[-1, 50]	550
LeakyReLU-8	[-1, 50]	0
Linear-9	[-1, 100]	5,100
LeakyReLU-10	[-1, 100]	0
Linear-11	[-1, 190]	19,190
Sigmoid-12	[-1, 190]	0

Model with $dim(z) = 15$

Layer (type)	Output Shape	Param #
Linear-1	[-1, 100]	19,100
LeakyReLU-2	[-1, 100]	0
Linear-3	[-1, 50]	5,050
LeakyReLU-4	[-1, 50]	0
Linear-5	[-1, 15]	765
Linear-6	[-1, 15]	765
Linear-7	[-1, 50]	800
LeakyReLU-8	[-1, 50]	0
Linear-9	[-1, 100]	5,100
LeakyReLU-10	[-1, 100]	0
Linear-11	[-1, 190]	19,190
Sigmoid-12	[-1, 190]	0

Model with $dim(z) = 20$

Layer (type)	Output Shape	Param #
Linear-1	[-1, 100]	19,100
LeakyReLU-2	[-1, 100]	0
Linear-3	[-1, 50]	5,050
LeakyReLU-4	[-1, 50]	0
Linear-5	[-1, 20]	1,020
Linear-6	[-1, 20]	1,020
Linear-7	[-1, 50]	1,050
LeakyReLU-8	[-1, 50]	0
Linear-9	[-1, 100]	5,100
LeakyReLU-10	[-1, 100]	0
Linear-11	[-1, 190]	19,190
Sigmoid-12	[-1, 190]	0

Model with $dim(z) = 50$	Layer (type) Output Shape Param # Linear-1 [-1, 190] 36,290 LeakyReLU-2 [-1, 190] 0 Linear-3 [-1, 100] 19,100 LeakyReLU-4 [-1, 100] 0 Linear-5 [-1, 50] 5,050 Linear-6 [-1, 50] 5,050 Linear-7 [-1, 100] 5,100 LeakyReLU-8 [-1, 100] 0 Linear-9 [-1, 190] 19,190 LeakyReLU-10 [-1, 190] 0 Linear-11 [-1, 190] 36,290 Sigmoid-12 [-1, 190] 0
Model with $dim(z) = 100$	Layer (type) Output Shape Param \# Linear-1 [-1, 190] 36,290 LeakyReLU-2 [-1, 190] 0 Linear-3 [-1, 150] 28,650 LeakyReLU-4 [-1, 150] 0 Linear-5 [-1, 100] 15,100 Linear-6 [-1, 100] 15,100 Linear-7 [-1, 150] 15,150 LeakyReLU-8 [-1, 150] 0 Linear-9 [-1, 190] 28,690 LeakyReLU-10 [-1, 190] 0 Linear-11 [-1, 190] 36,290 Sigmoid-12 [-1, 190] 0
Model with $dim(z) = 150$	Layer (type) Output Shape Param \# Linear-1 [-1, 190] 36,290 LeakyReLU-2 [-1, 190] 0 Linear-3 [-1, 150] 28,650 LeakyReLU-4 [-1, 150] 0 Linear-5 [-1, 150] 22,650 Linear-6 [-1, 150] 22,650 Linear-7 [-1, 150] 22,650 LeakyReLU-8 [-1, 150] 0 Linear-9 [-1, 190] 28,690 LeakyReLU-10 [-1, 190] 0 Linear-11 [-1, 190] 36,290 Sigmoid-12 [-1, 190] 0
Model with $dim(z) = 190$	Layer (type) Output Shape Param # Linear-1 [-1, 190] 36,290 LeakyReLU-2 [-1, 190] 0 Linear-3 [-1, 190] 36,290 LeakyReLU-4 [-1, 190] 0 Linear-5 [-1, 190] 36,290 Linear-6 [-1, 190] 36,290 Linear-7 [-1, 190] 36,290 LeakyReLU-8 [-1, 190] 0 Linear-9 [-1, 190] 36,290 LeakyReLU-10 [-1, 190] 0 Linear-11 [-1, 190] 36,290 Sigmoid-12 [-1, 190] 0

C Additional figures

C.1 Training metrics

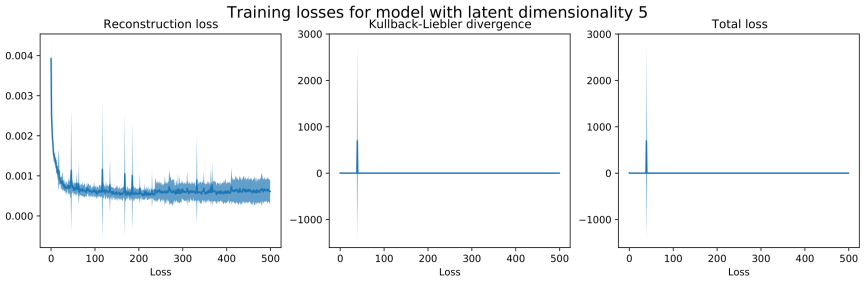


Figure 12: Loss per epoch for model with latent dimensionality 5

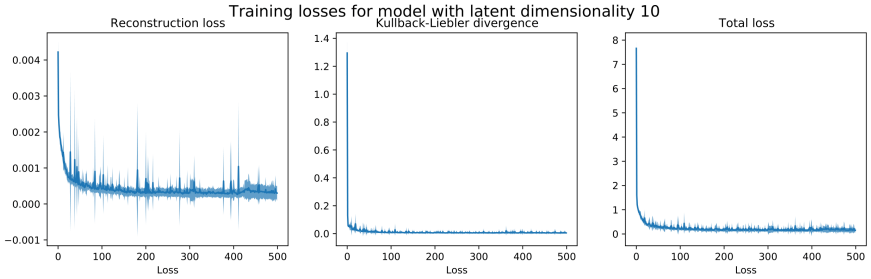


Figure 13: Loss per epoch for model with latent dimensionality 10

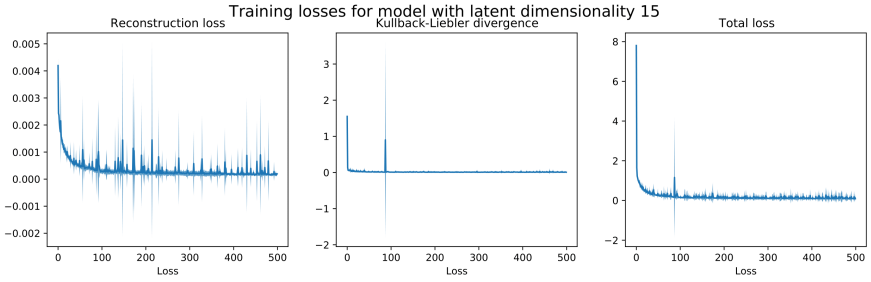


Figure 14: Loss per epoch for model with latent dimensionality 15

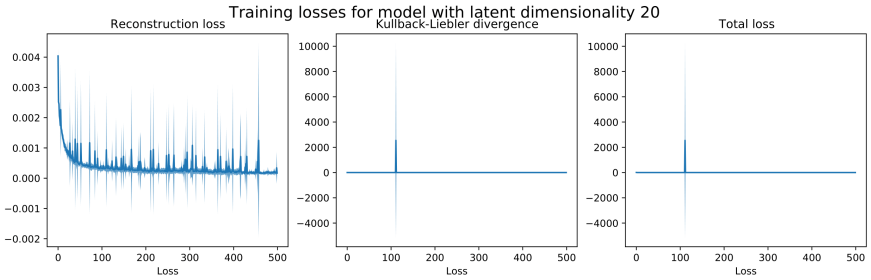


Figure 15: Loss per epoch for model with latent dimensionality 20

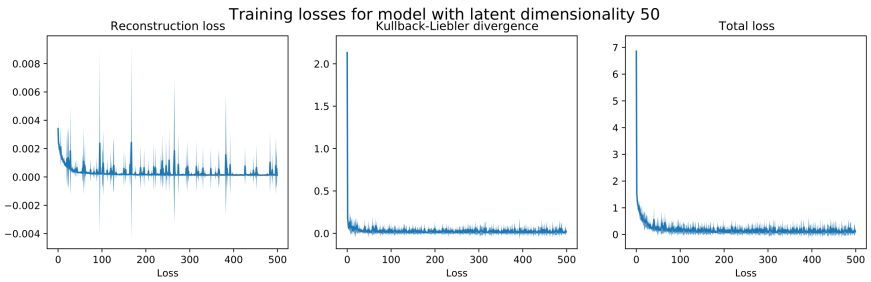


Figure 16: Loss per epoch for model with latent dimensionality 50

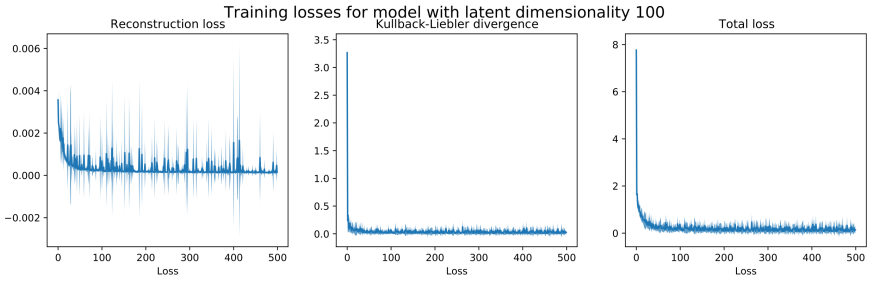


Figure 17: Loss per epoch for model with latent dimensionality 100

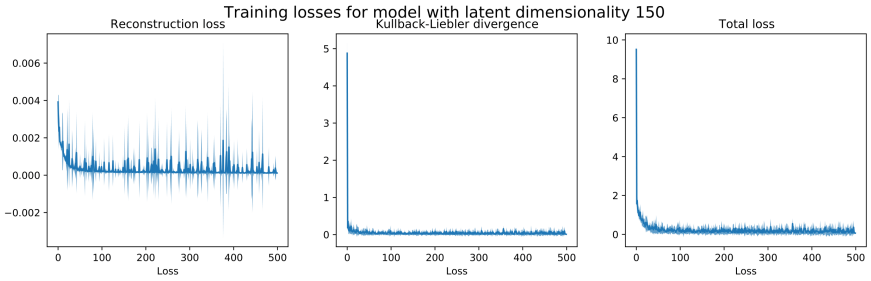


Figure 18: Loss per epoch for model with latent dimensionality 150

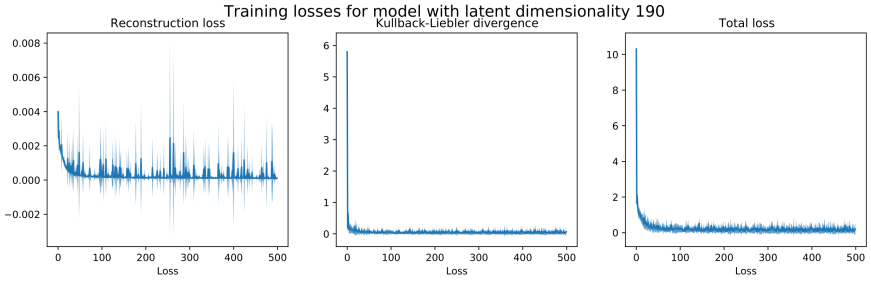


Figure 19: Loss per epoch for model with latent dimensionality 190