STAT 157 Final Project

Zachary Mackin

May 2024

1 Introduction and Context

Dimensionality reduction is a technique for data preprocessing utilized in data analysis and machine learning domains to reduce the dimensionality of the input while still preserving the nature of the data. There is an obvious information theoretic perspective to take here in that we want to transform the data into a lower dimensionality while still preserving as much information as possible. Dimensionality reduction falls into two main categories, linear techniques of combining the features (i.e Principal Components Analysis), and nonlinear techniques (i.e Local Linear Embeddings) [Mainali et al., 2021]. These nonlinear techniques are often more effective given that data in the "real world" is unlikely to follow any linear structure, but rather fall on some underlying nonlinear structure [van der Maaten et al., 2007].

Throughout this paper, we will explore some methodologies that fall into both of these categories but are overall unified on their motivation coming from an information-theoretic framework. These methodologies will seek to overcome an overarching weakness of statistical methods of dimensionality, that they potentially miss any nonlinear manifolds of the data that are not computable in any statistical quantity [Cilibrasi and Vitányi, 2007]. We will see a multitude of adaptations of various informational theoretical quantities in this domain, such as measuring information loss by modifying features. This gives us an optimization problem where we want to maximize the reduction of dimensions while maintaining the lowest possible information loss. Additionally, we will explore how we can treat various aspects of the dimensionality reduction process as random variables, and compute the conditional entropy between these RVs in order to inform our dimensionality reduction techniques. This approach will allow us to explore specific examples from the literature, showcasing how these techniques are applied in real-world scenarios. We'll also compare their advantages over dimensionality reduction methods that lack the grounding in information theory. Ultimately, this will lead us to other related topics such as pre-training, unsupervised learning, or even deep neural networks in general. This will enable us to not only illustrate the universality of information theory but also allow us to draw more profound connections between these domains that may just seem correlated.

Ultimately the goals of this paper, which I have slightly refined since initial proposals to ensure they are both precise and reflective of the research outcomes are the following. The first is to help explain how Information Theory is utilized to create metrics that allow us to construct algorithms and make decisions throughout the dimensionality reduction process. The second is to exemplify how from a geometric perspective, Information Theory can construct new representations of our data that assist us in the dimensionality reduction process. Finally, I will look at similar processes such as Variational Autoencoders and Pre-training, and compare how Information Theory is utilized in these domains.

2 Literature Review

2.1 Conditional Entropy

Frequently, Information Theory is utilized to develop metrics within dimensionality reduction. One particularly helpful metric is conditional entropy. The conditional entropy of our random variable Y given a random variable X notated by $H(Y|X)$ which can be interpreted as the uncertainty left in Y once we observe our random variable X [Cover and Thomas, 2009]. The intuition behind utilizing this as a metric between features can be motivated by thinking about extreme cases, say when the conditional entropy between two features X_1 and X_2 is zero or when it is extremely high. In the case where $H(X_1|X_2)$ is zero X_2 completely determines X_1 which if X_1 is our target feature means X_2 is a great predictor for X_1. However, if these are both features we're choosing for our model, this would mean that having both of these features is unnecessary as X_2 determines X_1 for us. This is the exact opposite if we have an extremely high conditional entropy. Let's see how we can leverage this utilizing dimensionality reduction techniques.

2.1.1 Using List of Features

Say we have some list of features $\mathcal{X} = \{X_1, \ldots, X_n\}$ where $|\mathcal{X}| = N$. Let's say our ultimate goal is to reduce this feature list such that this new selection of features $\mathcal{X}^*$ has $|\mathcal{X}^*| = M$ s.t $1 \leq M < N$. To minimize the predictors we want to select features that are as independent from other features as possible, which means that they will have high conditional entropies given the other predictors [Mainali et al., 2021]. To give a good estimation of just how independent these features are from other features they propose a metric:

$$avg_i = avg\{H(X_i|X_j) : 1 \leq i \neq j \leq N\}$$

From there we can choose the M features with the largest average to achieve our goals. One modification of this process proposed is a more iterative process where we first choose our first feature utilizing the process we shared before (the feature that maximizes the average single conditional entropy), notating this X_{i1}. On the k_{th} time step where we will have selected $X_{i1}, \ldots, X_{ik}$ we can, instead of maximizing just the single entropy, maximize the following value

$$avg_j\{(H(X_i|X_{i1}, \ldots, X_{ik}, X_j) : 1 \leq i \neq j \leq N\}$$

which assists us in taking in account the features we have already chosen [Mainali et al., 2021]. In practice, we will generally utilize the iterative process until we have chosen our M_{th} feature.

2.1.2 Using the Target Feature

Generally in dimensionality reduction, we care just as much about the relationship between our features as we do the relationship between our features and our target which we denote Y. In this case, we compute $H(Y|X_i)$ to measure just how informative our feature is, with low entropy implying that X_i is very informative as mentioned in the background. Thus we will select the top M features that minimize this entropy to reduce the number of features we have selected while still maintaining as much predictive power of the target feature as we can [Mainali et al., 2021].

2.2 Information Geometry

Another metric that Information Theory develops that assists us in dimensionality reduction commonly is the KL-Divergence. The intuition behind this is relatively straightforward. Ultimately, if our initial distribution is p and the output of our dimensionality reduction algorithm is q, ideally, we would want p and q to be indistinguishable from each other. Thus, we can think about utilizing KL-Divergence, which is a measurement of distinguishability, as something we should minimize as we reduce the dimensionality of our data. This raises the question of which divergence do we take, $KL(p||q)$ or $KL(q||p)$, which given the absence of symmetry in the metric, is a relevant consideration. The best practice is to utilize Fisher Information Distance, an extrinsic measure, that measures the amount of information an observable random variable X carries about an unknown parameter θ of which X depends on [Srivastava et al., 2007]. We want to use KL-divergence to estimate this quantity, which given the symmetry of Fisher Information Distance, we can estimate using the following

$$\sqrt{KL(p||q) + KL(q||p)}$$

which converges to our Fisher information distance as $p \to q$.

Now that we have this intuition for utilizing KL-Divergence let's look at a method of dimensionality reduction developed by Carter, Raich, and Hero called Fisher Information Nonparametric Embedding (FINE). This dimensionality reduction procedure differs from most in that it, through use of this metric, relies on information geometry rather than Euclidean geometry [Carter et al., 2008]. Developing this method requires the introduction of some notation. We define a statistical manifold M to be a set whose elements are probability distributions [Carter et al., 2008]. We also will notate our approximation of Fisher information distance as $\hat{D}_F$. To perform FINE we need a family of datasets, of which we can use our $\hat{D}_F$ metric to obtain a lower dimensional embedding. However, in the "real world" we don't have the true parameterization and thus the PDFs to utilize our Metric. We instead will have a family of datasets which we will denote $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$. From there, for each dataset, we calculate some $\hat{p}_i(\mathbf{x})$ which represents a density estimate of $\mathbf{X}_i$. This requires the assumption that these datasets are realizations of some underlying PDF and they lie on some manifold [Carter et al., 2008]. This gives us a set of PDFs on our manifold which we can call $\mathcal{P} = \{p_1, \ldots, p_N\}$. Using our $\hat{D}_F$ we can develop a G matrix where $G(i, j)$ is the shortest approximate Fisher information path for each pair p_i, p_j (called the geodesic approximation) which the methodology of FINE develops as the following function:

$$G(p_1, p_2; \mathcal{P}) = \min_{M,\mathcal{P}} \sum_{i=1}^{M-1} \hat{D}_F(p_{(i)}, p_{(i+1)}), \quad p_{(i)} \to p_{(i+1)} \ \forall i$$

where here M is the number of segments used for the approximation and $p_{(i)}$ is the ith point along a geodesic path between p_1 and p_2. From there we use a multidimensional scaling method that takes our G matrix and embeds it in Euclidean space of the desired dimension.

To understand why we do this, we can compare FINE to Isomap, a commonly used dimensionality reduction technique. Isomap does not utilize information geometry, instead opting to utilize Euclidean geometry in a methodology akin to FINE [Tenenbaum et al., 2000]. This limits it as a dimensionality reduction technique because there are many domains in which there is not always a "straightforward and meaningful Euclidean representation of the data" [Carter et al., 2008]. Thus in those domains, Isomap does not have much to go off of, and FINE is much more effective at generalizing the data given the statistical manifold. FINE effectively enables us to have a joint embedding of all of these datasets into a single Euclidean space. This resultant low-dimensional state can often be far lower in dimensionality than if we had just maintained Euclidean space like in Isomap.

2.3 Information Loss

Another manifestation of Information Theory being utilized to construct metrics is through the concept of "Information Loss". We can treat dimensionality reduction as an optimization problem, in which we are trying to reduce the dimension of our feature space to some $N \geq 1$. Throughout this process, we want to diminish the loss of information as much as possible. To utilize Information Loss our algorithm employs a technique similar to the previous G matrix, in that it constructs a new representation of the data. This representation is going to be a labeled graph $G = (V, E)$ with vertices $V(G) = \{1, \ldots, n\}$ and our edges which construct an undirected graph with no self-loops defined and each edge is an unordered pair $E(G) \subset E_c(G) = \{(x, y) | x, y \in V\}$ where $E_c(G)$ represents all possible edges in an undirected graph without self-loops [Zenil et al., 2023]. Using this graph we define the Information Loss to be $I(G, e_i) := C(G) - C(G \backslash e_i)$ where C(x) is the Kolmogorov Complexity of x and $e_i \in E(G)$. Worth noting is that the Kolmogorov Complexity, though not directly developed from Shannon's Information Theory, is a quantity s.t $K(x)$ is the length shortest effective binary description of x. If this sounds familiar, it's because it's reminiscent of Entropy; indeed, expected Kolmogorov complexity equals Shannon Entropy [Grünwald and Vitányi, 2004].

Now that we have our tools, a natural question to ask is what do these vertices and edges represent? The answer is, we are incredibly flexible, and making the necessary modifications, the vertices can be anything ranging from features, to rows, to pixels of an image, all the way up to representing the datasets themselves (similar to the previous example). Making no assumptions about the particulars of our algorithm let's analyze the methodology. We will assume the number of edges represent the dimensionality that we are trying to reduce, that is we will run while $|E(G)| > N$. At each time step, we will calculate the Information Loss $I(G, e_i)$ for each i. From there we remove the edge $e_i \in E(G)$ that has the lowest Information Loss. This utilization of minimizing information loss is incredibly intuitive, and Information Theory gives us the tools to determine just how to represent 'information' as a mathematical quantity.

2.4 Related Applications: Pre-training

Pre-training has a very similar motivation to dimensionality reduction. They both are attempting to construct a new representation of the data that is ultimately a better *fons et origo* for our model to utilize. The general methodology we take to do this is certainly different, in general, we first pre-train a model that can be used to complete some surrogate task and run our input through this, this gives us a representation of our data. We then fine-tune this model to complete the task that we are trying to accomplish [Sahai, 2022a]. In the Natural Language Processing (NLP) domain, this looks like a model that predicts the next word given a string of text. However, say our goal is to output the sentiment of the text. We utilize the surrogate model minus the part that decodes it into the output (more on what this can look like in the next section) to give us a representation of the text. We take this representation and pass it into a model that will output the sentiment of the text. The thought is this surrogate model will give us some meaningful representation since to achieve its goal it needs to understand the data. We can see this is a very similar perspective to that of dimensionality reduction, as the pre-training step develops some simpler representation of the data that maintains the essence of the data before running our model.

To see how Information Theory is utilized in pre-training let's look at a methodology called *unsupervised bin-wise pre-training*. This method can be broken down into four steps [H. et al., 2020]:

1. **Estimation of Mutual Information**: We start by constructing a DNN with some effective initialization (the exact process I'll omit due to irrelevancy). We ideally want the mutual information between our input data and the activation of each neuron, however, we can not compute this given the difference in dimensionality between each of these. So instead the mutual information will be computed between each dimension of the input and the activations of each neuron. Then, assuming that each of the d dimensions are independent of each other the equation $I(X, Y) = \sum_{i=1}^{d} I(X_i, Y)$ can be utilized [H. et al., 2020]. Thus we have an effective estimate of the mutual information. The intuition here is that the larger the average mutual information for a certain neuron is, the more important it is in capturing the given data point.

2. **Select number of bins**: From there a method called Partial Information Decomposition is utilized. This looks for neurons that capture similar information, which is marked by having similar mutual information values given a data point, and "binning" them together.

3. **Construct a Hypergraph**: Once these neurons are binned together it allows us to create a hypergraph, which similar to earlier methods, gives us a different way to represent our data.

4. **Update Parameters**: We then update parameters by computing a "divergence factor" which is measured by computing the KL-Divergence between the bin with the highest average mutual information and the bin that is currently having its parameters updated [H. et al., 2020]. This is a natural metric to utilize as we want the worst bins to become more and more like the best bins and improve in their ability to represent the data.

After these parameters converge, the output of the model can be used for the model that is fine-tuned for the problem's objective. We see once again, Information Theory is utilized to create a more effective representation of the data and give us a way to quantify how vital certain features are in representing the data.

2.5 Related Applications: Variational Autoencoders

Continuing our dive into techniques utilized in deep learning we find one of the seminal methods of utilizing Information Theory in Variational Autoencoders (VAEs). VAEs have a structure that is reminiscent of the general communication diagram we get in Information Theory as you can see in figure 1. The general process of the encoder is to take each point of a dataset, like say an image, and pass it through a deep neural network (DNN). This DNN will output not a single point, but rather a distribution within latent space [Kingma and Welling, 2019]. This latent space relies on a very similar assumption to our assumptions within FINE, that there is some lower dimensional manifold that we can reduce our input to. Thus, this encoder outputted distribution is a compressed representation of the data that we hope represents the essentials of our initial point, an idea we have seen within both dimensionality reduction and pre-training. From this latent distribution the decoder, which is also a DNN, takes in our latent distribution and attempts to reconstruct our initial point from our latent representation.

To see how Information Theory is utilized it might help to look at the specifics of the VAE framework, which are motivated by Bayesian Statistics. Let x represent our data and z represent our latent representation of the data. We want to develop the following distribution for $p_\theta(x)$:

$$p_\theta(x) = \int_z p_\theta(x, z)dz = \int_z p_\theta(x|z)p_\theta(z)$$

which we can view from a Bayesian perspective as $p_\theta(x|z)$ is a Gaussian multiplied by a vector of real numbers, thus our $p_\theta(x)$ distribution is a mixture of Gaussians [Asperti et al., 2021]. This can then inform the three parts of our Bayesian framework, our prior $p_\theta(z)$, our likelihood $p_\theta(x|z)$, and posterior $p_\theta(z|x)$. However, given the intractability of calculating our posterior for more complex distributions (which we commonly have) we estimate it using our encoder which we call $q_\phi(z|x)$. Our decoder is an estimation of our likelihood [Sahai, 2022b]. To develop our parameterization for our encoder and decoder we generally utilize stochastic gradient descent to maximize our evidence-based lower bound (ELBO). ELBO is defined as the following:

$$ELBO_{\theta,\phi}(x) = \mathbb{E}_{z\sim q_\phi(\cdot|x)}[\ln(\frac{p_\theta(x, z)}{q_\phi(z|x)})]$$

Knowing that our KL-Divergence is defined as $\mathbb{E}_{z\sim q_\phi(\cdot|x)}[\ln(\frac{q_\phi(z|x)}{p_\theta(x,z)})]$ we can rewrite ELBO as the following expression:

$$ELBO_{\theta,\phi}(x) = \ln(p_\theta(x|z)) - KL(q_\phi(\cdot|x)||p_\theta(\cdot))$$

.

Given we are maximizing ELBO we are attempting to maximize the first term and minimize the second term. The first term we can view as the log-likelihood of x (our data) given z (our latent distribution),

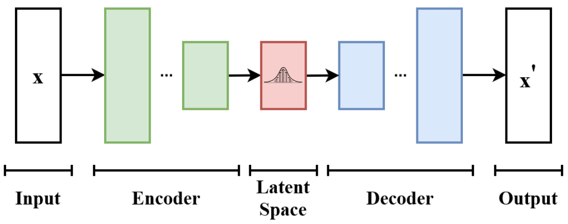

Figure 1: A diagram depicting the VAE system [Commons, 2021]

and the second term is once again our KL-Divergence or the distinguishability of our encoder and the prior distribution. Once again, the application of KL-Divergence is natural, as we want latent representations that are as indistinguishable as possible from our prior. This utilization of KL-Divergence demonstrates how Information Theory informs our auto-encoding process, and how we essentially view VAEs from the communication lens developed by Shannon. Many modern techniques are derived from these VAEs, which demonstrates once again the indispensable role Information Theory has played within modern modeling. It also allows for profound interplay with other domains, like in this example statistics, given the Bayesian framework and maximum likelihood utilized in VAEs.

3 Conclusion

One of the major concepts we can see that information theory generates throughout this process is the use of information geometry. Commonly we visualize and handle our data through Euclidean geometry, because as humans that is what is intuitive to us. Not only is it how we view the world, but our mathematical systems are developed largely on an Euclidean basis. However, for many domains, like say genomic data, an Euclidean interpretation does not make much sense and is likewise ineffective. Thus, we can use Information Theory to give us a new geometric perspective, often constructing information geometry representations of our data. These representations, like the geodesic approximation matrix, enable us to have lower dimensional dimensionality reductions while still maintaining much more of the initial information than an Euclidean counterpart.

We also see how many Information Theoretic quantities we can utilize to demonstrate how we make decisions during the dimensionality reduction process. These can be split roughly into two different categories. The first are quantities that demonstrate just how similar our representation is to the initial data. Generally, the metric used here is KL-Divergence, which is an extremely motivated quantity as we want our representation to be indistinguishable from the initial data while significantly more compressed. The other are various quantities ranging from entropy to mutual information, however, they all measure just how much of the data's information is maintained from step to step. It is very interesting to me just how commonly these metrics have been utilized, and even methodologies that stray away from information theory seem to rely on these concepts. I would be interested to see if there is any dimensionality reduction process that could exist completely independent of information theory.

Seeing how Information Theory has informed new methodologies within the young field of deep learning could be an interesting place for future work. This is interesting both from a perspective of understanding the current state of the field, but also potentially informing future developments in the field given the prevalence of intellectual arbitrage within deep learning and the intuitive applicability of Information Theory. I'm very curious if the utilization of Information Theory is something that has been exhausted or if there are further motivations that can be developed through utilizing information theory.

References

[Asperti et al., 2021] Asperti, A., Evangelista, D., and Piccolomini, E. L. (2021). A survey on variational autoencoders from a greenai perspective.

[Carter et al., 2008] Carter, K. M., Raich, R., and III, A. O. H. (2008). An information geometric framework for dimensionality reduction.

[Cilibrasi and Vitányi, 2007] Cilibrasi, R. and Vitányi, P. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

[Commons, 2021] Commons, E. W. (2021). Vae basic. https://commons.wikimedia.org/wiki/File:VAE_Basic.png.

[Cover and Thomas, 2009] Cover, T. and Thomas, J. (2009). *Elements of Information Theory*. Wiley-Blackwell, 2 edition.

[Grünwald and Vitányi, 2004] Grünwald, P. and Vitányi, P. M. B. (2004). Shannon information and kolmogorov complexity. *CoRR*, cs.IT/0410002.

[H. et al., 2020] H., A. G., C., V., and V.S., S. S. (2020). Unsupervised bin-wise pre-training: A fusion of information theory and hypergraph. *Knowledge-Based Systems*, 195:105650.

[Kingma and Welling, 2019] Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

[Mainali et al., 2021] Mainali, S., Garzon, M., Venugopal, D., et al. (2021). An information-theoretic approach to dimensionality reduction in data science. *International Journal of Data Science and Analytics*, 12:185–203.

[Sahai, 2022a] Sahai, A. (2022a). Pre-training and fine tuning. Lecture at University of California, Berkeley. EECS 182 Lecture 20.

[Sahai, 2022b] Sahai, A. (2022b). Self-supervision, autoencoders. Lecture at University of California, Berkeley. EECS 182 Lecture 14.

[Srivastava et al., 2007] Srivastava, A., Jermyn, I. H., and Joshi, S. (2007). Riemannian analysis of probability density functions with applications in vision. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

[Tenenbaum et al., 2000] Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2023.

[van der Maaten et al., 2007] van der Maaten, L., Postma, E., and Herik, H. (2007). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research - JMLR*, 10.

[Zenil et al., 2023] Zenil, H., Kiani, N. A., Adams, A., Abrahão, F. S., Rueda-Toicen, A., Zea, A. A., and Tegnér, J. (2023). Minimal algorithmic information loss methods for dimension reduction, feature selection and network sparsification.