Comparing Novel Approaches to Learning: The Information Sieve vs. Generalized Low Rank Models

Kendrick D. Cancio, Skylar W. Carfi

Algorithm Development Project

Abstract. *We review a novel algorithm for unsupervised learning called the Information Sieve. We first discuss its information theoretic background and the algorithm, then compare this model to another novel approach to learning called Generalized Low Rank Models. We attempt compare results between the two models on 3 benchmark applications: Lossy Compression and Inpainting with MNIST digits, and classification using the MADELON dataset. In lossy compression, our GLRM produced visually more accurate representations of Digits than IS using the same compression rate. We failed, for various reasons, in both inpainting and classification tasks to yield satisfying results.*

1. Background

For our project, we were interested in the algorithm development option. We were especially interested in learning about any new approaches to unsupervised learning, a hot topic in Machine Learning. Presently, there is an increased focus on learning deep representations of unlabeled data. This unsupervised learning context is where we find the popular artificial neural network for example. Hierarchical clustering is another example where we find multiple layers of information learned from our data. Methods like these involve the discovery of latent variables, which is essentially the meaningful information contained in the data. In our research, we found a new algorithm presented at ICML in 2015[2], and further updated at ICML in 2016 with the additional paper "The Information Sieve"[1]. The Information Sieve (IS) was appealing to us due to its seemingly approachable nature. A large part of the appeal of the IS is the intuitive explanation of the methodology.The author claims that, as opposed to many other models where signal is learned from the data "all at once," the Information Sieve algorithm is more reminiscent of human learning[3].

2. The Information Sieve Algorithm

The Information Sieve derives its name from the conceptual idea that one can progressively sift out information from a data set in the form of latent variables. Given a data-set, a function is constructed that explains as much of the dependence in the data as possible. Then, using this function, the data set is transformed into the "remainder information". We can then iteratively pass the remainder information through the Information Sieve to extract factors that explain progressively a smaller amounts of the dependence in the data. In this way, the process can be compared to sifting the data set through progressively finer filters to extract the information.

3. Mathematical Background

The Information Sieve is based on information theoretical concepts, particularly the Infomax Principle. ICA is an example of an infomax technique, as is the IS.

3.1. Mutual Information

This principle states that a learned mapping from some input X to output Y should be constructed so as to maximize the "mutual information" (I) between X and Y subject to some constraints. Naturally, $I(X, Y)$ seeks to quantify the amount of information in I that is contained in Y. Correlation is a simple example of a mutual information that is limited to $\mathbb{R}$. In the IS we define $I(I; O) = H(I) - H(I|O)$, where $H(\cdot)$ is Shannon Entropy.

3.2. Entropy

Shannon Entropy (Entropy, H) is a measure of how unpredictable the information is that comes from some source (distribution). Over a given domain, the distribution with maximum entropy is the uniform distribution since any value is equally likely to be observed. The formula for Entropy of a single random variable x with probability distribution P is:$H(P(x)) = E[-ln(P(x)]$. If we write out the formula for expectation we get:

$$-\int_D P(x) * \log P(x)dx \text{ or } -P(x) * \sum_{i=1}^{n} \log P(x)$$

There also exists the notion of "joint" entropy amongst multiple variables. In the discrete case we define $H(X_1, ..., X_n) = -\sum_{x_1} \cdots \sum_{x_n} P(x_1, ...x_n) log_2[P(x_1, ...x_n)]$

We make use of the joint entropy in defining "conditional entropy:"

$$H(X|Y) = H(Y, X) - H(Y)$$

3.3. Total Correlation

To extend the idea of mutual information to n random variables, the IS uses the information theoretic measure of Total Correlation (TC). TC essentially measures how similar a multivariate distribution is to the product of the distributions of the component variables. TC is defined as the Kullback-Leibler Divergence between these two quantities. Equivalently, we can find that $TC(X) = \sum_{i=1}^{n} H(X_i) - H(X)$ where X_i is a single variable in the data, X.

4. Sieve Algorithm

The Information Sieve algorithm is comprised of two main parts and an iteration step:

1. Optimizing $TC(X^{k-1}; Y_k)$
   Here we construct a variable Y_k, an arbitrary function of X^{k-1}, such that it explains as much of the dependence in the data as possible.

2. Remainder Information
   Construct the remainder information X_i^k as a probabilistic function of X^{k-1} and Y_k such that $I(X^{k-1}; Y_k) = 0$ and $H(X^{k-1}|X_i^k, Y_k) = 0$

3. Iterate
   Run the remaining information again through steps (1) and (2) and terminate either when $TC(X^{k-1}; Y_k) = 0$ (the remaining information is independent) or when the optimization step stops producing positive results.

5. Problem

Our original intent in this project was to implement the IS algorithm in Julia, and assess its performance on new data sets. Although the algorithm seems relatively clear on the surface, we had great trouble understanding it in full enough detail to implement it, even for simple cases. Reading reviews of the paper, we realized that we were not alone in the confusion. One ICML reviewer had this to say:

> The algorithm is not properly presented...In particular, the description of the procedure is intertwined with explanations of why various steps work and how they can be modified. It is frustrating that the main algorithm is obscured, especially in a submission to what is primarily a Computer Science conference.

After experiencing these frustrations we shifted the focus of our project. In the two IS papers, the IS was bench marked against other learning algorithms on a number of tasks. However, we realized that one algorithm that it was not compared to is Generalized Low Rank Models (GLRM). This is likely because GLRM is, itself, a novel approach to learning.

We aim to compare the performance of these models at the three primary tasks of: Lossy Compression, Imputing (inpainting), and Classification by performing these tasks with GLRM ourselves using the Juia implementation[5] and comparing our results to those presented in the IS papers.

In lossy compression, as the name suggests, the aim is to store the target information in a smaller amount of memory than the original. In this case, it is not necessary to maintain the fidelity of the file. Compression ratio is prioritized over accuracy so long as the original information may be understood. A common use is the lossy compression of images where a loss in quality isn't necessarily detrimental to the image. Here we directly compare both methods at compressing 60000 digits in the MNIST dataset.

The second task is inpainting which is a subset of the problem of imputing data. The bottom half of images will be removed and must be reconstructed using only knowledge of other complete images and the top half of the incomplete images. Because there is quantitative benchmark of success provided in the sieve paper, a qualitative visual approach will be taken to determine success. Again, we will compare both methods on inpainting the digits in the MNIST dataset.

Finally, we will compare both methods on a difficult classification problem using the MADELON dataset. In the paper where the Information Sieve is first introduced, "Sifting Common Information from Many Variables" (Ver Steeg), the IS was evaluated against some standard models on a couple of classification tasks. One, where the IS performed better than all other evaluated competitors was using the MADELON dataset.

6. Datasets

6.1. MNIST

The MNIST database consists of 70,000 examples of handwritten digits centered in a 28x28 pixel image. These images are presented in a compressed file format that, when extracted into a csv file, yields rows that represent individual images of digits. The first column is an integer from 0 to 9 representing the handwritten digit encoded in the remaining 784 columns which represent grayscale values from 0 to 255 of the pixels of the image. After downloading, we process this data identically as described in the sieve paper by first normalizing the values and then rounding them at a threshold of 0.5 so that the image is effectively composed of black and white pixels. Of these 70,000 examples, the database is split into 60,000 training examples and a test set of 10,000 samples. This dataset was made available to download directly in csv format for both the training and test datasets.

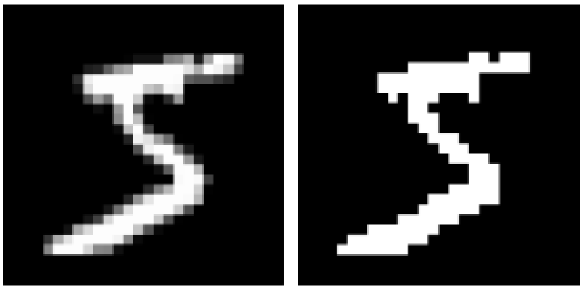

Figure 1. Thresholding to Create Binary Images

6.2. MADELON

The MADELON dataset is an artificially created binary classification dataset designed to have many confounding factors[6]. The data is generated by first creating 32 independent Gaussian clusters and introducing some covariance between them. These clusters are then placed at random on the vertices of a five dimensional hypercube and randomly labeled either +1 or -1. The five dimensions represent the 5 informative features to which 15 linear combinations of them are added to form additional redundant features. Finally, 480 additional features were added with no predictive power for a total of 500 features. This dataset was also made available in downloads of the training and test datasets but with separate downloads for the classification labels.

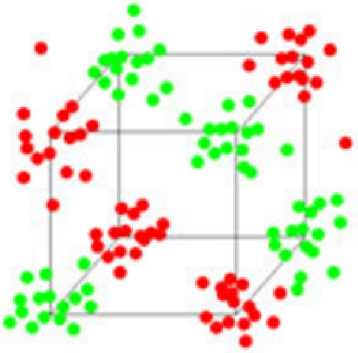

Figure 2. MADELON Dataset[6] Visualization

7. Methodology

7.1. GLRM for Lossy Compression of MNIST Digits

The first comparison we ran against the Information Sieve was the task of lossy compression on the MNIST dataset using GLRM. The sieve claims a reduction of file size of about 70% when compressing 50,000 digits using 100 sieve representatives. This brings the bits used to store an MNIST digit from 784 to 243. Because the performance of GLRM compression depends on the rank of the model, a rank had to be chosen that would produce a similar compression rate per MNIST digit so that the visual quality of each image can be directly compared.

GLRM finds a low rank factorized representation of the observed data Y. Thus our compressed representation of our images is stored in two matrices X and W whose product is approximately our original data Y. Matrices X and Y are real valued and each entry requires 64 bits of memory to hold. If we use approximate values, this memory requirement can be reduced to 16 bits. The total memory required is now the sum of the number of entries in X and W multiplied by 16 bits. Matrix X is of size $n * k$ where n is the number of compressed MNIST digits and k is the rank of the model. Matrix W is of size $k * 784$ where 784 is the number of bits in the original image (each bit represents a feature in the dataset). Having established this, we can calculate a formula for the approximate file size of storing n MNIST digits using GLRM of rank k : $(16n + 12544)k$ where the cost of storing an additional digit is then $16k$.

One interesting feature arises from this calculation: there is an initial fixed cost to storing information using GLRM. This indicates that it is a method best used for storing large amounts of data. Now, considering the compression rate of the sieve, we may calculate that using a model of rank 15 will yield a similar average compression rate to the sieve of approximately 244 bits per digit.

Overall, the author used 50,000 digits of the MNIST data to perform this lossy compression. Our compression utilized all 60,000 training digits however, this discrepancy would only serve in the sieve's favor because more observations are fitted within the same rank model. The author also points out that spatial information was not utilized and emphasizes this point by applying a consistent random shuffling of the pixels of each digit. We held ourselves to the same constraint in fitting our GLRM because glrm does not take into account spatial proximity of features.

7.2. InPainting

Another Application of the Information Sieve presented by Ver Steeg is "In-Painting". Using MNIST digits once again, the author removed the bottom half of the image for a subset of the digits and then attempted to fill them back in using the latent variables derived from the algorithm. With the Information Sieve, "missing data is handled quite gracefully" as you are simply able to optimize results over the observed values only.

GLRMs are also well designed for handling missing data. With GLRMs we are able to impute unobserved values. We designated that bottom portion of the image as unobserved for 10% of our digits. We then fit a rank 50 model to the data using Hinge Loss and no regularization.

Unfortunately, due to limitations in computing power, we did not use 50,000 training digits as Ver Steeg did. We used only 7,500 digits.

7.3. Classification

In the paper where the Information Sieve is first introduced, "Sifting Common Information from Many Variables" (Ver Steeg), the IS was evaluated against some standard models on a couple of classification tasks. One, where the IS performed better than all other evaluated competitors was using the MADELON dataset.

We see in Figure 3 that beyond about rank 3, The IS outperforms all of the methods to which it was prepared. The closest competitor being "factor analysis". However, GLRMs, being a new method itself, were not evaluated again the IS.

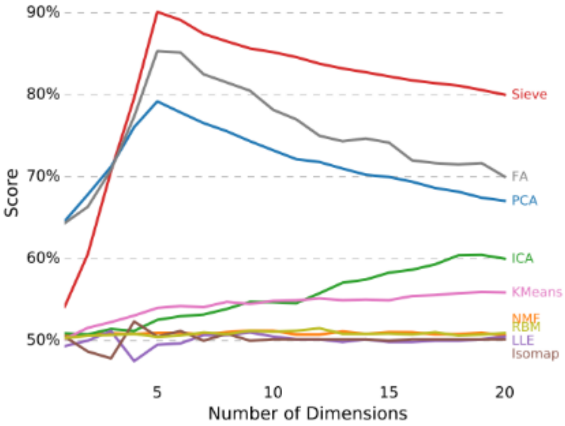

Figure 3. IS Outperformed Alternative Models on MADELON

8. Results

8.1. Lossy Compression

Ultimately, GLRMs performed noticeably better, than the Information Sieve at the same compression rate. Below is a figure showing a comparison of randomly chosen digit representatives before and after compression using GLRMs. The true digits are on top.

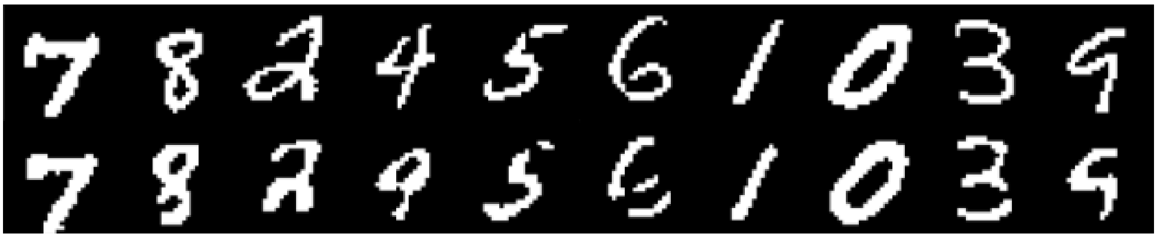

Figure 4. Random Lossy Compression Results Using GLRM

We can see that the digit's structure is mostly conserved through the compression, 1 being the most preserved of all. As an artifact of the process, the strokes of the line are noticeably thickened as in the case of the digits 8, 2, and 9. This distortion seems to have connected features in the digit 4 so as to make it appear as a 9.

In comparison, the recovered digits from the information sieve appear noisy (Fig 2). Rather than thickening the strokes, the lines appear pixellated with gaps and holes. The sieve appears to have trouble identifying the digits 8 and 9 while also confusing the digits 0 and 6.

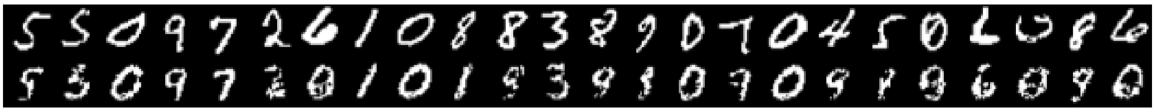

Figure 5. Lossy Compression Results Presented in "The Information Sieve"

8.2. Inpainting

The results of Inpainting were far less than ideal. Although there were some discernable shapes in the lower half of our images, in no way did they complete the image to produce a discernable number. We ultimately

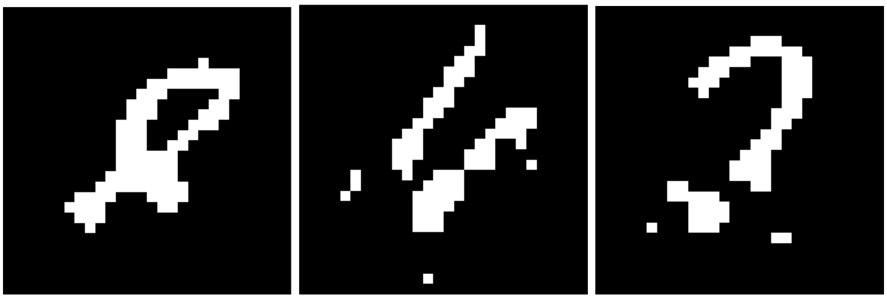

Figure 6. Inpainting Using GLRM Yielded Poor Results

8.3. Madelon

Ultimately technical issues hindered any real analysis or comparison between the results of the Information Sieve and GLRMs. Across many attempts to fit a low rank model to predict the classification, none of them yielded a success rate above 52% percent. As we Investigated our poor results, we noticed that the model was predicting TRUE for all observations.

9. Conclusion

Overall, GLRMs outperfomed the Information Sieve at lossy compression. The difference is visible just by inspection however, difficulties in implementing the remaining two comparison tests severely limit the degree to which we can adequately benchmark the Information Sieve against GLRMs. Looking forward, it is important to note improvements in the methodology for future comparisons. It is, firstly, imperative to utilize the full dataset. This applies to the inpainting exercise. Our effort was to first produce a working model at low scale so future improvements to the project would include using more information. The second improvement is using cross validation to adequately scale the loss parameters of the models. This applies particularly to the MADELON dataset where although all the observations could be loaded, our initial model did not sufficiently penalize misclassified points. Since this feature was boolean, the hinge loss was relatively small in comparison to the quadratic loss of the remaining 500 features which take on values in the order of 200-400. Automatically scaling the loss functions through the glrm package yielded errors so we resorted to manually using different scales for the hinge loss: a technique which never achieved above a 52% success rate.

Overall, we discovered that although seemingly intuitive on the surface, the Information Sieve algorithm is a complex procedure that we were not able to replicate. Further, comparing IS to GLRM also proved difficult in the remaining time.

References

[1] Galstyan and Steeg. 2016. The Information Sieve. ICML

[2] Steeg, Gao, Reing, and Galstyan. 2016. Sifting Common Information from Many Variables. ICML

[3] Steeg. The “information sieve” with bonus eigen-faces! https://apparenthorizons.com/2015/07/20/the-information-sieve-with-bonus-eigen-faces/

[4] ICML Submission 66 Review. http://icml.cc/2016/reviews/66.txt

[5] Udell, Horn, Zadeh, and Boyd. Generalized Low Rank Models.

[6] MADELOLN Dataset. https://archive.ics.uci.edu/ml/datasets/Madelon