ORIE 4741 Midterm Report

Kendrick D. Cancio, Skylar W. Carfi

Algorithm Development Project

1. Motivating a New Methodology

In Machine Learning today, 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 feed 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. We often try to boost performance by extracting the latent information a priori, and then feed the transformed data into our learning algorithm.

One prominent method for extracting latent variables is through Independent Component Analysis (ICA). As the name suggests, this technique seeks to transform data into statistically independent parts. One shortcoming to ICA is that it does not permit a quantifiable way to determine how much benefit there is from a modeling perspective to include an additional layer/latent variable in a deep representation of data. The appeal of the IS is that it decomposes the data one step at a time. This allows a modeler to analyse the value of each subsequent pass through the IS in improving model performance.

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.
These are the main components of the theoretic background needed to understand the IS algorithm.

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. Applications Thus Far

The primary application of the Information Sieve is in independent component analysis. As the paper claims, the algorithm provides and exponentially faster method of performing ICA (although without guarantees of global optimality). Thus, we can apply the IS to signal processing problems where incoming information can be decomposed into underlying signals. Other applications of ICA include face recognition, stock market prediction, inpainting, and recovery of spatial clusters.

Aside from these, the Information Sieve is also capable of lossy and lossless compression. Once the latent variables have been determined, if the algorithm is run in reverse, it will take the independent components and construct the original information. This results in lossless compression if the algorithm is run until termination or lossy compression if the remaining information is discarded by terminating the algorithm early.

6. Future Plans

6.1. Attempt to Collaborate

This week we contacted Gregg Ver Steeg, the primary author and code creator. We asked him about the computational complexity of the algorithm specifically. We are hoping that he replies and that we can continue to correspond with him and seek guidance on what he sees as possible next improvements to, or applications of the IS.

6.2. Computational Complexity

As the nexus of this course is big data algorithms that scale linearly with data, it is of primary importance to find out the computational complexity of the IS. We are told that it scales linearly with d, but we are not sure about n. While we hope that Dr. Steeg will be willing to explain it to us, we may be able to uncover it ourselves either analytically in Big-O or at least empirically by running the code and different size data sets. In fact, quote from an anonymous review of the paper for the ICML conference affirms our confusion:

> It is frustrating that the main algorithm is obscured, ... - What is the runtime of the algorithm? How does it scale? Empirically and big-O. It is mentioned that the method is linear in the number of variables in Sec4, but this needs to be backed up and also related to cardinality of variables, size of dataset and other parameters$_2$.

We think it would be a reasonably achievable task to try to demystify some of these things, and will certainly grow our understanding of how to analyze algorithms.

6.3. Test on New Data Sets

In Dr. Steeg's paper, the IS was put to the test on a few common applications and data sets where its performance could be bench-marked against traditional methods such as ICA. All of these tests however were on small data sets. Our plan is to test the IS on a couple of large data sets using his publically available code on github. We will seek to find, as the author did, some common data sets for which the performance of some learning algorithm is known so that we can assess the relative performance of the IS.

References

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

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

[3] Wikipedia