Notes on Icebreaker

Shalin Shah
AI Sciences
Target Corporation
Sunnyvale, CA 94086, USA
shalin.shah@target.com

January 2020

Abstract

Icebreaker [1] is new research from MSR that is able to achieve state of the art performance on inference in which there is inherent missing data. Using mutual information, Icebreaker is able to suggest which values in the data to impute for maximum benefit. These notes are an amalgamation of information from various articles and tutorials including autoencoders, variational inference, variational autoencoders, the evidence lower bound, set based learning and finally leading to Icebreaker. References are provided whenever appropriate. There may be factual errors and typos in these notes. Please send them to the author.

1 Background

1.1 Bayes Rule

Posterior Inference using the Bayes rule is commonly used to do inference based on some data and a prior on the parameters.

The Bayes rule is given by: $P(M|D) = \frac{P(D|M)P(M)}{P(D)}$

Where, $P(M|D)$ is the posterior, $P(M)$ is the prior on the model parameters and $P(D|M)$ is known as the likelihood. The denominator $P(D)$ is the marginal likelihood also called the evidence.

The proof of this rule is directly observed from the definition of conditional probability.

$P(D, M) = P(D|M)P(M) = P(M|D)P(D)$

In most problems, it is often very difficult to compute the marginal likelihood because of the intractable integral which has to be computed across the entire parameter space. Monte Carlo integration methods can be used, but they might prove to be very resource intensive. If conjugate priors are used, the posterior has a closed form expression.

1.2 Autoencoders

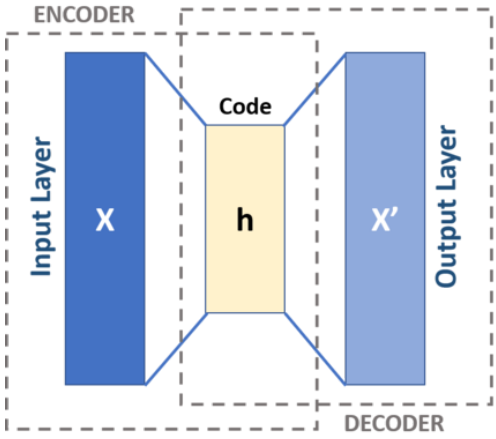

Figure 1: The Autoencoder (borrowed from Wikipedia)

Figure 1 shows a typical autoencoder. The data is fed to an encoder which compresses it to a low dimensional representation. This embedding is then fed to a decoder that tries to reconstruct the original input. Autoencoders can be used as a means of pre-training a more complicated neural network. They can also be used to generate low dimensional embeddings of very large dimensional data, and the embeddings can then be used as features for other tasks.

Autoencoders are generative models. By perturbation of the input to the decoder, subtle differences can be discovered and new data generated. This is especially useful for images. But the autoencoder can severely overfit the data and it may happen that the decoder produces garbage for inputs only slightly different from the encoded data. Training with noise (i.e. adding (Gaussian) noise to the observations during each iteration) somewhat solves the problem, which is equivalent to L2-regularization [2]. But full probabilistic inference is possible using variational autoencoders (section 1.4).

1.3 Variational Inference and ELBO

Posterior Inference can be accurately computed using sampling. However, for very large problems, a better method is variational inference. Instead of trying to sample from the posterior, a simpler distribution is learnt by minimizing the K-L divergence between the posterior and the simpler distribution. The parameters of the simpler distribution (Gaussian is a common choice) are called variational parameters.

Computing K-L divergence still requires computing the posterior and hence computing the intractable marginal likelihood. It is possible to optimize another quantity called the evidence lower bound and not compute the posterior at all. I have replicated some

of the math in [3].

$logP(X) = log \int_Z p(X, Z)$ (marginalizing across the latent variables)

$log \int_Z \frac{p(X,Z)q(Z)}{q(Z)}$ (multiplying and dividing by q(Z))

$= log(E_q(\frac{p(X,Z)}{q(Z))})$ (Importance Sampling to compute the integral using an expectation)

$>= E_q(log \frac{p(X,Z)}{q(Z)})$ (Jensen's Inequality)

$= E_q(log(p(X, Z)) - E_q(log(q(Z)))$ (Expectation of log of a distribution is its entropy H(Z))

$= E_q(log(p(X, Z) + H(Z)$

This last equation is called the evidence (or variational) lower bound (ELBO). Now, in variational inference, the K-L divergence between two distributions has to be computed. Let's see how ELBO fits into this.

$KL[q(Z)||p(Z|X)] = \int_Z q(Z) log \frac{q(Z)}{p(Z|X)}$

$= - \int_Z q(Z) log \frac{p(Z|X)}{q(Z)} = -ELBO + log(p(X))$

Thus minimizing the KL divergence is equivalent to maximizing the ELBO (which does not require computing the posterior). Note that if one is working with Gaussians, there is a closed form expression for the KL divergence between two Gaussians.

1.4 Variational Autoencoders and Amortized Inference

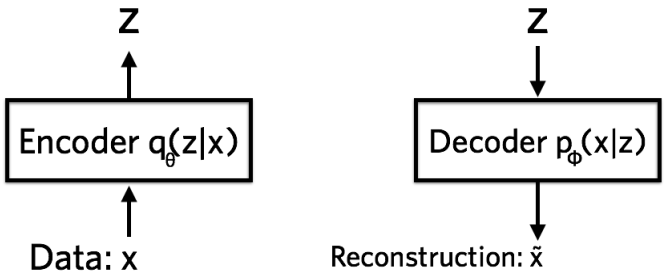

Figure 2: Variational Autoencoder (borrowed from jaan.io)

Similar to an autoencoder, a variational autoencoder is also a generative model, but differs from an autoencoder in that the encoder produces the parameters of a Gaussian

instead of an embedding with limited interpretability. The variational autoencoder may have several layers of non-linearities in the encoder as well as the decoder and the outputs of the encoder are the means and the variances of the variational distribution. This results in much better learning, many times better than a denoising autoencoder. By sensitivity analysis, the interpretation of each dimension of the variational distribution can be understood which is very useful for image reconstruction (e.g. adding a pair of glasses to an image of a face (See `https://bit.ly/37yDTs1`)).

By amortizing the inference using a neural network, the number of parameters does not scale linearly with the data size. This makes it highly tractable to learn the parameters of the encoder as well as the decoder. Icebreaker uses a modified version of the variational autoencoder which can learn using a variable number of inputs. This comes from the deep sets [4] approach which is in section 1.7. If both the input data and the output are assumed to be Gaussians, there is a closed form cost function for the variational autoencoder.

1.5 Mutual Information

This section will be useful to understand the reward function expressions of Icebreaker. Mutual information between two random variables measures the reduction in uncertainty in one variable when the other is known.

$I(X;Y) = \sum_y \sum_x p(x,y) log(\frac{p(x,y)}{p(x)p(y)})$

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

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

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

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

These alternative forms of mutual information will be useful to understand the reward function in Icebreaker. Also, note that as in the Icebreaker paper, $H(Y|X)$ is actually an expectation. Please see Wikipedia on conditional entropy. Another important thing is that mutual information is symmetric. So, $I(X;Y) = I(Y;X)$

1.6 Short note on MVN

There is a property of the multivariate normal distribution with a diagonal covariance matrix in which since all covariances are 0, the multivariate normal distribution factorizes into the product of individual univariate normal distributions. For a proof, see section 3 of `http://cs229.stanford.edu/section/gaussians.pdf`.

1.7 Deep Sets

As Frege said, a set is a collection of objects with some common property. So, two sets may have differing number of elements and the order of the objects does not matter.

Deep Sets [4] is recent work that learns from the set formulation of data.

The Kolmogorov-Arnold representation theorem says that any multivariate function can be represented as univariate function compositions and additions (See `https://bit.ly/2TjMryL`). This is closely related to the neurons in a neural network, and this theorem proves that a neural network is a universal approximator.

This recent work [4] proves that even permutation invariant functions (like a sum of common non-linearities) can be universal approximators.

This work says that the following function can be a universal approximator:

$\rho(\sum_x(\phi(x))$ (permutation invariant function)

For suitable functions ρ and ϕ. The sum in this equation makes it permutation invariant which makes it possible to learn on variable number of features and the order of the features does not matter as ϕ is shared.

The basic idea in the proofs appears to be that if $\sum_x \phi(x)$ is injective (countable case) or homeomorphic (uncountable continuous case) then ρ can always be chosen such that the permutation invariant function is a universal approximator (see `https://bit.ly/36tBTzR` for a discussion). For details, please refer to the appendix of [4].

This is useful for Icebreaker as because of missing features, a set based approach is very applicable.

2 Icebreaker and Applications

Icebreaker [1] uses a variational autoencoder with a set based formulation like deep sets. It alternates between sampling the decoder weights (using stochastic gradient Hamiltonian Markov chain Monte Carlo (SGHMC)) and updating the encoder (which generates the latent information). For feature imputation, it then assigns a reward to each unobserved feature based on mutual information and chooses the one that reduces the uncertainty in the decoder weights the most. It keeps choosing unobserved features and notes that the results are much better than their earlier work [5].

The cost function to update the encoder is similar to the ones described in section 1.3.

The reward function for imputation is similar to the ones described in section 1.5.

The reward function for imputation is:

$$R_I(x_{i,d}, X_O) = H(p(\theta|X_O)) - H(p(\theta|X_O, x_{i,d}))$$

where the second term in the RHS is an expectation. This is similar to the entropic formulation of mutual information:

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

For prediction (when there is a target variable), the reward function takes another form in which the uncertainty in the target is included.

$$R_P(x_{i,d}, X_O) = E_{p(x_{i,d}|X_O)}(H(p(y_i|x_{i,d}, X_O)) - E_{p(\theta, x_{i,d}|X_O)}(H(p(y_i|\theta, x_{i,d}, X_O))$$

The paper recommends taking a linear combination of the two reward functions R_I and R_P.

As they note, this work is very applicable to medicine in which it can recommend which further medical tests to conduct of a patient to maximize the amount of information gained from the test. Tests are invasive, risky (radiation) and costly and it is wise to minimize the number of tests done. Another application might be a recommender system in which each product shown to the user can be chosen in such a way that the amount of information about a guest is maximized. The reward function can then be a linear combination with greater emphasis on the target (purchases, views etc.).

References

[1] Wenbo Gong, Sebastian Tschiatschek, Sebastian Nowozin, Richard E Turner, José Miguel Hernández-Lobato, and Cheng Zhang. Icebreaker: Element-wise efficient information acquisition with a bayesian deep latent gaussian model. In *Advances in Neural Information Processing Systems*, pages 14791–14802, 2019.

[2] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[3] Xitong Yang. Understanding the variational lower bound. 2017.

[4] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[5] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018.