Depth First Learning **Wasserstein GAN**

Week 2: 10 Feb, 2019

Scribed by: James Allingham & Taliesin Beynon

2.1 Trade-offs between distance metrics

Refer to Figure 1 in *A note on the evaluation of generative models*.

- When choosing between different metrics (in a loose sense of the word) for probability distributions there are trade-offs.
- For example, minimizing the KLD between our model and true data will result in a model that places probability density over all of the modes of the true data distribution, at the cost of also placing probability density in areas where the true data distribution does not.
- On the other hand, JSD, Reverse KLD, and MMD do the opposite. The model will place all of its mass on the mode(s) that contains the most probability density in the true data distribution.
- The choice of whether we want to cover all the modes but draw low probability samples from our model, or to miss modes but only draw high probability samples, will depend on the application.
- The wording above has been slightly misleading since this issue can also come up with uni-modal data. For example, if the true data distribution has some interesting shape and we are trying to model it with a single isotropic Gaussian distribution.
- There is active research into precision & recall type metrics for measuring these trade-offs in practice.
- Note that even if we choose a model that is complex enough, in theory, to capture all of the modes of the true data distribution, without penalizing the model for missing modes the model might still miss some of the modes.
- We can understand why KLD does not penalize the model for assigning probability density to areas which have zero probability density under the true data distribution but considering the definition of KLD:

$$D_{KL}(P||Q) = \int_x P(x) \log \frac{P(x)}{Q(x)} dx. \tag{2.1}$$

  The $P(x)$ term means that the ratio term involving $Q(x)$ is ignored when $P(x) = 0$. However, $Q(x)$ can still have an effect on the KLD in areas where $P(x) = 0$ because having a high probability density for $Q(x)$ in these areas necessarily (for a normalized distribution) means that we can have less probability density in areas where $P(x) > 0$.

2.2 JSD

- JSD is a combination of forward and reverse KLDs. However, it is not a simple average of the two, but rather uses a distribution $M(x)$ that is defined as the average of $P(x)$ and $Q(x)$:

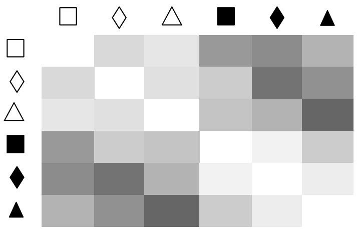
$$JSD(P, Q) = 0.5KLD(P||M) + 0.5KLD(Q||M). \tag{2.2}$$

The reason for using this averaged distribution might be to reduce the sensitivity to areas where either $P(x)$ or $Q(x)$ are low – as long as one of P and Q has non-zero density, the denominators in the KLD terms will not cause the KLDs to go to infinity.

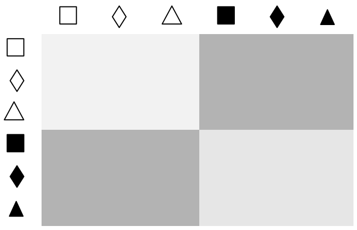
- The notation $||$ in the KLD is not related to the $|$ notation for describing conditional distributions.
- The square root of JSD is a proper metric, which means that a number of properties (for example the triangle inequality) hold. This is not true for most divergences.

2.3 MMD

- MMD is related to moment matching.
- An example to develop some understanding/intuition for MMD (adapted from Arthur Gretton's MLSS Africa talk on MMD-GANs):
  - Suppose we have samples from two distributions P and Q, $\{\Box, \Diamond, \triangle\}$ and $\{\blacksquare, \blacklozenge, \blacktriangle\}$ respectively, as well as a distance metric between samples k, and that we want to measure the distance between the two distributions.
  - We can use k to measure the distance between all of the samples from both P and Q, creating the following distance matrix:



  - In the matrix above, the darker the cell, the higher the distance between two samples. Note that the diagonal elements are white because the distance between a sample and its self is zero. Also, note that samples from the same distributions are on average smaller than samples from different distributions:



  - We can consider the difference between the average distances of samples from the same distributions, and the average distances between samples from different distributions to be a measure of the distance between the two distributions themselves.
  - This is basically what MMD, defined as

$$MMD(P, Q) = \mathbb{E}_{PQ}[k(x, x') - 2k(x, y) + k(y, y')]^{0.5} \tag{2.3}$$

where x and x' are samples from P, and y and y' are samples from Q, is doing. The first term in the expectation corresponds to the top left quadrant of the matrix above. Similarly, the second term corresponds to the top right and bottom left quadrants, and the last term corresponds to the bottom right quadrant.

2.4 Jensen's Inequality

- Jensen's inequality is used to derive equation 6 in *A note on the evaluation of generative models*, but it might not be immediately clear how it is actually used. As a reminder, here is what Jensen's inequality says

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}[x]) \tag{2.4}$$

  if f is a convex function and x is a random variable.

- In the proof for equation 6, Jensen's is used to make the following inequality:

$$\sum_{\mathbf{x}} P(\mathbf{x}) \int_{[0,1]^D} \log q(\mathbf{x}+\mathbf{u}) d\mathbf{u} \leq \sum_{\mathbf{x}} P(\mathbf{x}) \log \int_{[0,1]^D} q(\mathbf{x}+\mathbf{u}) d\mathbf{u}. \tag{2.5}$$

- It might not be clear that there are any expectations here, however, because the random variable is $\mathbf{u}$, which is distributed uniformly over the range $[0,1]^D$, we can say:

$$\int_{[0,1]^D} \log q(\mathbf{x}+\mathbf{u}) d\mathbf{u} = \int P(\mathbf{u}) \log q(\mathbf{x}+\mathbf{u}) d\mathbf{u} = \mathbb{E}[\log q(\mathbf{x}+\mathbf{u})]. \tag{2.6}$$

2.5 Log-likelihood

Near the start of section 3.1 of *A note on the evaluation of generative models*, the following statement is made: 'Since the discrete data distribution has differential entropy of negative infinity, this can lead to arbitrary high likelihoods even on test data'. This statement raised a number of questions.

- Firstly, why does the discrete data distribution have a differential entropy of negative infinity?
  - Recall that differential entropy is simply the continuous version of entropy – i.e. replacing the sum with an integral:

$$-\sum_{x} P(x) \log P(x) \to -\int_{x} p(x) \log p(x) dx. \tag{2.7}$$

  - Now consider what happens to the differential entropy as the variance of $p(x)$ goes to 0. More concretely, let us consider a normal distribution with arbitrarily low variance. In this case, the density at the mean becomes arbitrarily high and the integral tends to infinity, which means that the differential entropy tends to negative infinity.
- As an aside, this is why the KL-Divergence between any continuous and any discrete distribution is infinity. A discrete distribution can be viewed as a collection of Dirac deltas. Recall that the Dirac delta is defined as the limit of the Gaussian distribution as the variance tends to 0.
  - This is another one of the reasons that we often see noise added to discrete valued data, e.g. in GAN training.
  - Perhaps this is related to other cases of adding noise to discrete data, such as label smoothing.
- Secondly, how does differential entropy of negative infinity lead to arbitrarily high likelihoods (even for test data)?
  - Consider some training data point. We can place a Gaussian with its mean at the data point and then make the variance smaller and smaller, which will make the likelihood larger and larger. (Recall that as the variance goes to zero, our Gaussian becomes a Dirac delta).

  - Let us consider a more realistic example might be one where we are modelling the pixels of images. Here we could model the value of each pixel with a Gaussian. Consider training dataset where each image has a black pixel in the top left corner. We can model this pixel as a Gaussian with a mean of zero, and again make the variance arbitrarily low. Now if any test example has an image with a black pixel in the top left, the likelihood for that image will be arbitrarily high!
    - As an aside, this is why it is often beneficial to reduce the number of bits used to model an image, for example using 5 bits per colour channel of a pixel rather than 8, since it will effectively add more noise and prevent these arbitrarily high likelihoods from manifesting.

2.6 Samples and applications

- The essence of this section (and the previous one) is that you have a model with two components. And one of the components is what matters when you care about the samples, and the other component is what matters when you care about the likelihood. That is basically what you is being shown here, with some technical details about how these models come about.
- The point is that the quality of samples and the likelihood (as well as performance on actual applications) of a model are generally not necessarily related.

2.7 Evaluation based on samples and nearest neighbours

- The main idea in this section is to show that it can be tricky to determine if the model is producing good samples. More specifically, testing whether or not the model has simply memorized the training dataset.
- In particular, this section shows that the nearest neighbour test can easily be fooled by a model which performs simple transformations to the examples in the training set.
  - This is because nearest neighbours is based on Euclidean distance, which is not a good measure of the similarity between images, for example, and does not correspond to what the human eye/brain perceive as differences.
  - A potential solution to this problem might be to use other distance measures that correspond more closely with how the brain works, for example by looking at small patches of images or using convolutions.
- Because judging the quality of samples from a model is difficult, this makes this a bad proxy for the overall quality of the model, even in applications where all we care about are the samples.

2.8 Additional Comments

- The type of model you are using also has a big impact on how we should evaluate it, which isn't discussed in the paper.
  - For example, if your model is *correct*, then using maximum likelihood to estimate it will give a good result and log-likelihood will be a good measure of your model performance. However, if the model is wrong and you try use log-likelihood then you might run into the problems discussed in this paper.
  - Another example is energy based models, which when estimated using maximum likelihood can be very similar to GANs. Auto-regressive models, on the other hand, do not look anything like GANs when trained with maximum likelihood.

2.9 Summary

- This week was really about showing that it is difficult to evaluate generative models. We shouldn't necessarily just use evaluation metrics such as log-likelihood since they can be misleading.
- There are trade-offs to be made when choosing our evaluation metrics, and we should keep those in mind. For example, if your goal in learning a generative model is to produce good samples, then log-likelihood may not be the best choice of evaluation metric, however, if your goal is to minimize the KLD between the model and the true data distribution, then log-likelihood might be more reasonable.
- The choice of our optimization objective is also not so straightforward and involves trade-offs. For example, if we want to cover all the modes of the true data distribution, KLD might be a good choice, but if we care more about our model assigning a high probability to only the data points which have high probability in the true data distribution, then JSD or MMD might be better.
- Metrics such as sample quality and, in particular, Parzen window estimates can also be misleading.
- This is still an active area of research! For example, the paper *Improved Techniques for Training GANs* by Tim Salimans et. al. also contributed to this topic but did receive some criticisms so there is more to say here.