Resampling Generative Models: An Empirical Study

by

Prachi R. Jadhav

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Science

The University of Memphis

May 2023

Abstract

Despite Generative AI's rapid growth (e.g., ChatGPT, GPT-4, Dalle-2, etc.), generated data from these models may have an inherent bias. This bias can propagate to downstream tasks e.g., classification, and data augmentation that utilize data from generative AI models. This thesis empirically evaluates model bias in different deep generative models like Variational Autoencoder and PixelCNN++. Further, we resample generated data using importance sampling to reduce bias in the generated images based on a recently proposed method for bias-reduction using probabilistic classifiers. The approach is developed in the context of image generation and we demonstrate that importance sampling can produce better quality samples with lower bias. Next, we improve downstream classification by developing a semi-supervised learning pipeline where we use importance-sampled data as unlabeled examples within a classifier. Specifically, we use a loss function called as the semantic-loss function that was proposed to add constraints on unlabeled data to improve the performance of classification using limited labeled examples. Through the use of importance-sampled images, we essentially add constraints on data instances that are more informative for the classifier, thus resulting in the classifier learning a better decision boundary using fewer labeled examples.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Generative learning using deep neural networks [41] has gained significant attention over the last several years. Specifically, in generative learning, the goal is to model the data distribution, and thus, in theory, generative models can generate unlimited samples that can be used in downstream tasks such as classification. Deep generative models (DGMs) have been successful in generating realistic images [11], videos [20], text [5], audio [43] etc.

In the last decade, several types of DGMs have been developed. Prominent among those include Variational Auto Encoders (VAEs) [22], Generative Adversarial Networks (GANs) [11], Autoregressive models (ARMs) [38] and normalizing flow models (NFMs) [8]. All of these models use different approaches to model the data distribution. Specifically, VAEs learn a latent variable model using deep network layers to encode and decode the latent vectors. GANs use a *likelihood-free* approach where the idea is to train a generator-discriminator pair in an adversarial manner with the generator learning better representations of the data to generate samples for the discriminator and the discriminator learning to discriminate between real and generated samples. ARMs generate sequential data based on autoregessions that was widely used for predictions in time-series models. ARMs can compute the likelihood in a tractable manner since they assume the autoregressive property. On the other hand, VAEs cannot compute the likelihood tractably but rather use variational inference to approximate the likelihood using deep encoders and decoder layers. This allows VAEs to learn complex feature representations. In NFMs, the idea is to

combine the properties of both VAEs and ARMs. Specifically, NFMs can compute the likelihood in a tractable manner and at the same time, they can also learn complex feature representations like VAEs. This is done by using simple density functions with tractable likelihoods and then mapping them to more complex probability distributions using the data samples.

Regardless of the type of generative model, it is infeasible for the model to learn the data distribution exactly [36]. That is, each type of generative model makes underlying assumptions about the data that may or may not be always valid. For instance, in VAEs, to make variational inference feasible through the neural network layers, we assume that the latent vectors that represent the underlying characteristics of the data are normally distributed [22]. In general, if the data distribution is simple enough, then clearly, we can sample directly from the distribution and may not need complex approaches such as DGMs. Therefore, there is a need to improve the quality of samples that are generated from DGMs. In particular, one way to quantify the quality of samples generated by a DGM is based on the *bias* in the model's distribution. This type of bias can result in problems such as *mode-collapse* [37, 29, 33] where the DGM samples from a single mode of a multi-modal distribution and this results in generating samples that are very similar. A recently proposed approach by Grover et al. [13] addressed this problem and tried to reduce the bias of the samples generated by a DGM. Specifically, here, the main idea is to use an approach called *importance sampling* [25] to generate weighted samples instead of unweighted samples. The importance weights encode the importance of the generated samples with respect to the true distribution. For example, as shown in Fig. 1.1, if the goal is to generate samples from a complex distribution that is hard to sample, in importance sampling, we instead generate samples from a simpler distribution called the *proposal distribution* and weight these samples based on a *ratio* of probabilities. Specifically, for each sample, we divide the probability of the sample original distribution with its probability in the proposal distribution. This ratio acts as the importance weight for a sample. In [13], using the idea of importance sampling, we draw samples from the DGM and weight it based on the data distribution which is the true distribution from which we want to draw samples from.

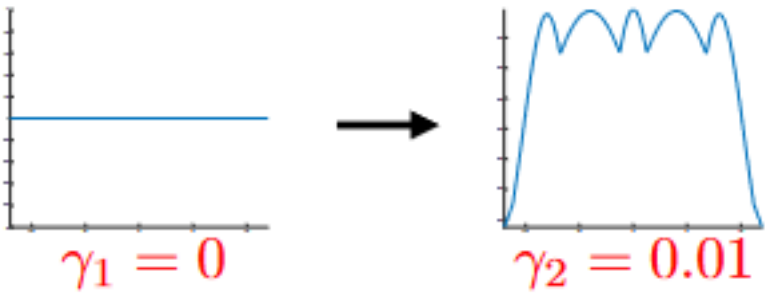

Figure 1.1: Sampling Importance Resampling, adopted from [24]

Unfortunately, to compute importance weights, we need to compute the ratio between the probability of a sample in the true distribution and its probability in the proposal distribution. This is infeasible in DGMs since we do not know how to compute this probability in a tractable manner, i.e., the data distribution is assumed to be a complex unknown distribution. In [39], a clever trick was proposed to weigh samples based on a *calibrated probabilistic classifier* [31, 14]. That is, this classifier is trained to compute the importance weights for samples that are generated from a DGM. While this approach seems generally applicable to *any* DGM, depending on the type of samples that are generated by the DGM, the trained probabilistic classifier could obtain very different results. Thus, the quality of the importance weights depends on how well we can train the probabilistic classifier.

In this thesis, we train a *calibrated* probabilistic classifier to perform importance sampling for images generated from VAEs and ARMs. Specifically, we use the well-known CIFAR10 image dataset [23] to train the generative models. We use convolutional VAEs as our VAE model for image generation and for ARMs, we use the most well-known image generation model called PixelCNN++ [38]. We implement sample importance resampling (SIR) [26, 9] as suggested in [13] to resample from the distribution learned by the probabilistic classifier for the generated samples. However, in [13], it was suggested that SIR can be implemented using a standard *Multinomial roulette* sampling method. in our experiments, we observed that for the weights that were generated, this approach failed to scale and yielded poor results. Therefore, we implemented a novel SIR where we approximate the Multinomial distribution over importance weights which

are probability ratios (computed from the probabilistic classifier) using a *Gumbel-softmax* distribution [19, 27]. This allows us to resample images from the importance distribution more efficiently. To empirically compare the quality of the the generated images with and without importance sampling, we use different standard metrics such as the inception score [37], the Fretchet distance [17] and the kernel inception distance [4].

Next, we develop a novel approach to use the generated images to improve downstream classification. Specifically, in [44], a loss function called the *semantic loss* was proposed to add symbolic knowledge to deep network training. Here, we add exactly-one constraints (as specified in [44]) over generated images. Specifically, the idea is that we want each generated image to belong to exactly one class (among all possible classes). Thus, as in semi-supervised learning, we can now treat the generated images as unlabeled data and add a limited number of labeled examples to train the classifier. The semantic loss function learns to separate the labeled examples and at the same time assign classes to the unlabeled examples. Thus, assuming that the unlabeled examples has information about the classes, this will result in a more general classifier even using limited labeled examples. Further, by adding constraints on informative generated images, we can learn a more effective classifier. Using our SIR approach, we resample the generated images which are the most informative for the classifier. We evaluate our approach by comparing the performance of classification on CIFAR10 using limited labeled and a large number of generated images. Our results show that utilizing the importance sampling, we are able to learn a more accurate classifier trained on the semantic loss funtion as compared to using the generated images directly.

To summarize, our contributions in this thesis are as follows.

- We empirically evaluate the approach proposed in [13] to understand how well importance weights can be estimated for VAEs and PixelCNN using the CIFAR-10 dataset.
- We implement the Gumbel-softmax sampling to efficiently perform SIR based on the importance weight distribution from a calibrated probabilistic classifier.

- We apply the semantic loss function proposed in [44] to add constraints over images generated (and resampled) from generative models such that the unlabeled generated data can augment labeled examples within a semi-supervised classifier.
- We develop an open-source implementation in Pytorch that integrates the semantic loss function with importance sampling for generative models for image datasets.