Deep Generative Models

Evaluating Generative Models

Hamid Beigy

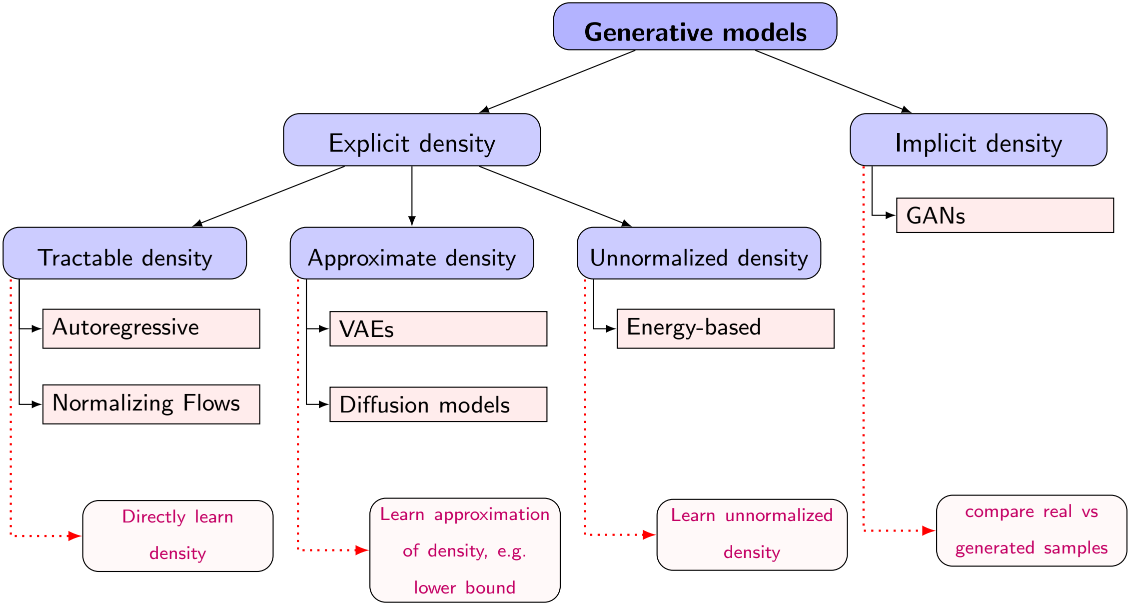
Sharif University of Technology

May 25, 2024

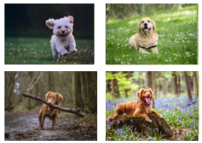
Table of contents

Introduction

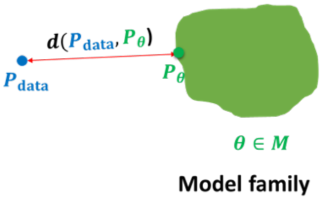
Generative models categories



1. Assume that the observed variable $\mathbf{x}$ is a random sample from an underlying process, whose true distribution $p_{data}(\mathbf{x})$ is unknown.



$x_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



2. We attempt to approximate this process with a chosen model, $p_\theta(\mathbf{x})$, with parameters θ such that $\mathbf{x} \sim p_\theta(\mathbf{x})$.

3. Learning is the process of searching for the parameter θ such that $p_\theta(\mathbf{x})$ well approximates $p_{data}(\mathbf{x})$ for any observed $\mathbf{x}$, i.e.

$$p_\theta(\mathbf{x}) \approx p_{data}(\mathbf{x})$$

4. We wish $p_\theta(\mathbf{x})$ to be sufficiently flexible to be able to adapt to the data for obtaining sufficiently accurate model and to be able to incorporate prior knowledge.

Credit: Aditya Grover

Evaluation of Generative Models

1. Evaluation of generative models is tricky
2. The key questions is about underlying task of the generative model.
   - Density estimation
   - Sampling / generation
   - Latent representation learning
   - More than one task.
3. How do we evaluate generative models?

Example (Evaluating density estimation)

When the given model has tractable likelihood, the evaluation is straightforward.

- Split dataset into train, validation, and test sets.
- Evaluate gradients based on the train set.
- Tune hyper-parameters based on the validation set.
- Evaluate generalization by measuring likelihoods on the test set.

1. Evaluating generative models requires metrics which capture
   - **Sample quality:**

     Are samples generated by the model a part of the data distribution?
   - **Sample diversity:**

     Are samples from the model distribution capturing all modes of the data distribution?
   - **Generalization:**

     Is the model generalizing beyond the training data?
   - **Interpretability and Controllability:** Understanding and controlling the latent representations learned by generative models.
   - **Sample Efficiency:** How many training samples do we need to train a generative mode with a good performance?
2. There is **no known metric which meets all these requirements**.
3. But various metrics have been proposed to capture **different aspects of the learned distribution**.

1. Generative models have become a popular topic in machine learning research.
2. The evaluation of generative models is crucial as it allows researchers and practitioners to assess the quality of generated samples.
3. Evaluating these models is challenging due to the lack of ground truths and the subjective nature of quality assessment.
4. How can we evaluate the effectiveness of a generative model?
   - **Quantitative methods:**

     These methods calculate some numerical scores based on some criteria.
   - **Qualitative methods:**

     These methods inspect the generated data visually or auditorily.
   - **Hybrid methods:**

     These methods combine quantitative and qualitative methods.

Example 1: Face Generative Model

1. The output of the generative model is synthesized facial images.
2. How can one decide whether the model output is acceptable or not.
3. The following methods can be utilized for evaluating the performance of such model:
   - **Quantitative methods:**

     Use the metrics such as Frechet Inception Distance and Inception Score to evaluate the quality of the generated images.
   - **Qualitative methods:**

     Visual Inspection by a human to qualitatively determine realism looking for unnatural facial features, artifacts, and/or inconsistencies.
4. We can also use geometrical facial features such as distance between facial landmarks (corners of the eyes, mouth, nose, eyebrows).
5. We can also use other types of image features such as texture, eye color, skin color, and hair color and compare them to population norm.

Example 2: Text Summarization Model

1. The goal of a text summarization model is to generate a concise, coherent, and comprehensive summary of a long body of text that is significantly shorter in length and is able to capture the main essence of the original text.
2. There is no single true answer for such a task, which makes the evaluation difficult.
3. The following methods can be utilized for evaluating the performance of such model:
   - **Quantitative methods:** Use the metrics such as ROUGE and BLEU to evaluate the quality of the generated summary text.
   - **Qualitative methods:** Use human evaluators using a standard scales.
4. In addition, we can use the following methods to evaluate the generated text:
   - Calculate the distance between sentence embeddings.
   - N-fold validation by running text summarization on N different permutations of the original text and expecting to achieve similar results.
   - Using Q&A model on both original and summary text and expecting to receive identical or very similar answers.

Qualitative methods

Qualitative methods

1. In qualitative methods, we can use methods such as
   - visual inspection,
   - pairwise comparison, or
   - preference ranking

   to assess how realistic, coherent, and appealing the generated data is.
2. We can also use methods such as interpolation, latent space exploration, or conditional generation to test how the generative model responds to different inputs or parameters.
3. Qualitative methods can provide intuitive and subjective feedback on generative model performance.
4. These methods have some **drawbacks**, such as being
   - time-consuming,
   - biased, or
   - inconsistent.
5. These methods usually considered as a supplementary method for evaluating generative

Human evaluations

1. One intuitive metric of performance can be obtained by having human annotators judge the visual quality of samples.
2. This process can be automated using Amazon Mechanical Turk (Salimans et al. 2016).
3. The task is to ask annotators to distinguish between **generated data** and **real data**.
4. For **MNIST** dataset and GAN model, annotators were able to distinguish samples in **52.4%** of cases (2000 votes total), where **50%** would be obtained by random guessing.
5. For **CIFAR-10** dataset and GAN model, annotators were able to distinguish samples in **78.7%** of cases.
6. A downside of using human annotators is that the metric varies depending on the setup of the task and the motivation of the annotators.
7. Also, results change drastically when we give annotators feedback about their mistakes.
8. By learning from such feedback, annotators are better able to point out the flaws in generated images, giving a more pessimistic quality assessment.

Quantitative methods

Quantitative methods

1. Quantitative methods involve calculating numerical scores based on some criteria.
2. These methods can be categorized as:
   - **Likelihood-based methods**
   - **Raw data-based methods**
   - **Feature-Based Metrics**
   - **Task-Based Metrics**
   - **Novelty-Based Metrics**
   - **Statistical Tests**
3. These methods can provide objective and standardized measures of DGM performance.
4. These methods have some limitations, such as
   - requiring a reference dataset,
   - being sensitive to model architecture, or
   - being hard to interpret.

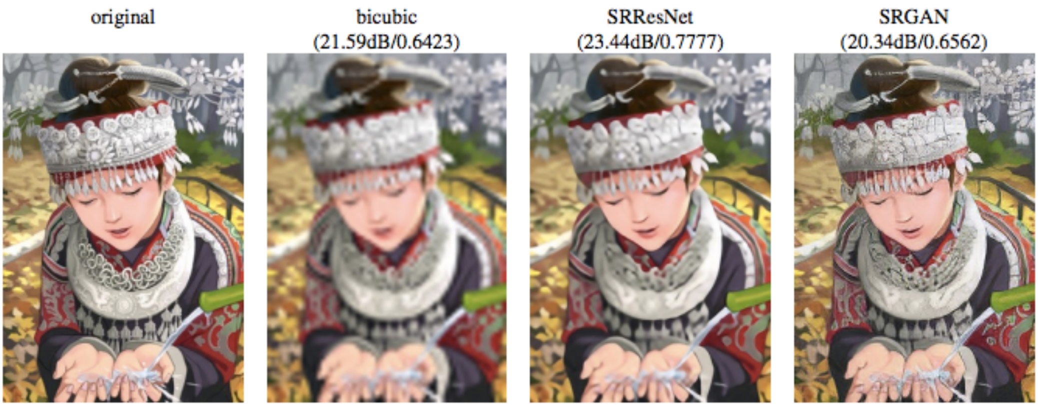
Quantitative methods

Likelihood-based methods

Likelihood-based methods

1. We have a dataset that sampled from p_{data} and generated samples from p_g.
2. Evaluating deep generative models (DGM) is hard because
   - the distributions of interest are often high dimensional,
   - the likelihood functions are not always available or easily computable.
3. A common way to evaluate a DGM is to measure how close p_{data} is to p_g.
4. Since sample complexity of traditional measure such as KL divergence or Wasserstein distance is exponential in the dimensionality of the distribution, they cannot be used for real world distributions.
5. The reduced sample complexity comes at the cost of reduced discriminative power.
6. These metrics cannot tell the difference between a model that memorizes the training data and a model that generalizes.

1. Some generative models, such as VAE, have intractable likelihoods.
2. For example, in VAE we can compare the evidence lower bounds (ELBO) to log-likelihoods.
3. For general case, kernel density estimates only via samples can be used.
4. Consider the following generated images, which of them is better?



5. Likelihood is not related to sample quality.

Quantitative methods

Raw data-based methods

Raw data-based methods

1. These methods assess generative models by comparing the generated sample with real ones from the same domain.
2. These methods are application-dependent. For example, for generating images, we can use the following pixel-based metrics:
   - mean squared error (MSE),
   - peak signal-to-noise ratio (PSNR), or
   - structural similarity index (SSIM).
3. These metrics dig deep into a pixel level, taking into account that the closer the pixels, the higher the image quality.
4. Pixel-based metrics also have some limitations, including
   - sensitivity to image transformations,
   - ignoring high-level semantic features, and
   - overlooking the aspects of diversity and innovation.

Quantitative methods

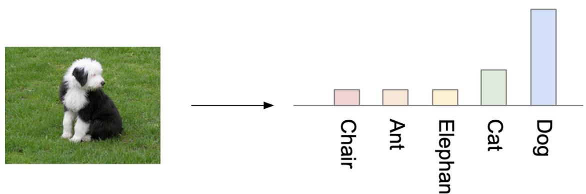
Feature-based methods

Feature-based methods

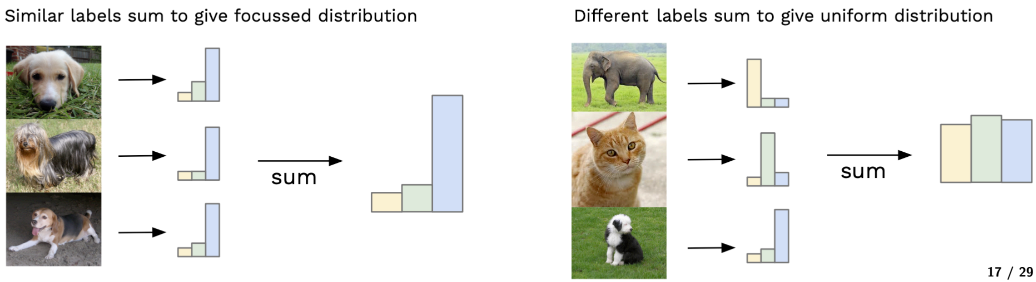
1. Deep learning methods, such as convolutional neural networks (CNNs), responsible for finding high-level features, such as shapes, textures, colors, and styles.
2. These methods do not directly compare raw data (e.g., pixels) but use a neural network to obtain features from the raw data.
3. Then, compare the feature distribution obtained from model samples with the feature distribution obtained from the dataset.
4. The metrics related to this method are
   - Inception score (IS),
   - Kernel Inception distance,
   - Fréchet inception distance (FID),
   - Perceptual path length (PPL),
5. These metrics compare the feature distributions of the generated and real images and determine how well this model preserves the quality and diversity of the original domain.

Inception score

1. The **inception score** takes a list of images and returns a single number, the **score**.
2. The score is a measure of how realistic the output of a generative model (GAN) is.
3. The score measures two things simultaneously:
   - The images have variety.
   - Each image distinctly looks like something.
4. If both things are true, the score will be high; otherwise, the score will be low.
5. The lower bound of this score is zero and the upper bound is ∞.
6. The inception score takes its name from the Inception classifier, an image classification network from Google.
7. Classifier takes an image, and returns probability distribution of labels for image.



1. If image contains just one well-formed thing, then output of classifier is a narrow distribution.
2. If image is a jumble, or contains multiple things, it's closer to the uniform distribution of many similar height bars.
3. The next step is combine the label probability distributions for many of generated images (50,000 images).
4. By summing the label distributions of our images, a new label distribution (marginal distribution) will be obtained.
5. The marginal distribution tells the variety in the generator's output:



Inception score

1. The final step is to combine these two different things into one single score.
2. By comparing label distribution with marginal label distribution for images, a score will be obtained that shows how much those two distributions differ.
3. The more they differ, the higher a score we want to give, and this is the inception score.
4. To produce the inception score, the KL divergence between label distribution and marginal label distribution is used.
   - Construct an estimator of the Inception Score from samples $\mathbf{x}^{(i)}$ by constructing an empirical marginal class distribution,

$$\hat{p}(y) = \frac{1}{m}\sum_{i=1}^{m} p(y \mid \mathbf{x}^{(i)})$$

   - Then an approximation to the expected KLdivergence is computed by

$$IS(G) \approx \exp\left(\frac{1}{m}\sum_{i=1}^{m} D_{KL}(p(y \mid \mathbf{x}^{(i)}) \parallel \hat{p}(y))\right)$$

Frechet Inception Distance

1. The **Inception score** solely relies on class labels, and thus **does not measure** overfitting or sample diversity outside the predefined dataset classes.
2. To address this drawback, the Fréchet Inception distance or FID score are use.
3. Frechet inception distance (FID) is a metric for quantifying the realism and diversity of images generated by generative models.
4. Realistic could mean that generated images of people look like real images of people.
5. Diverse means they are different enough from the original to be interesting and novel.
6. Unlike the earlier Inception score (IS) evaluates only the distribution of generated images.
7. Unlike IS, the FID compares the distribution of generated images with the distribution of real images that were used to train the model.

1. Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be the distribution of some neural network features of the images generated by the generative model.

2. Let $\mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ be the distribution of the same neural network features from the world / real images used to train the model.

3. The FID metric is the squared Wasserstein metric between two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$.

4. Thus, FID equals to

$$FID = \|\boldsymbol{\mu} - \boldsymbol{\mu}_w\|_2^2 + \operatorname{tr}\left(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_w - 2\left(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}_w\boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right)$$

5. FID has been shown to have a high bias, with results varying widely based on the number of samples used to compute the score.

6. To mitigate this issue, kernel Inception distance has been introduced.

Quantitative methods

Task-Based Metrics

Task-Based Metrics

1. Generative models can be evaluated using task-oriented metrics.
2. These metrics measure how well the generated sample serve downstream functions.
3. For example, the generated images can be evaluated in tasks like classification, segmentation, captioning, or retrieval.
4. These metrics offer insights into the practicality and suitability of the generative model for specific tasks and domains.
5. Examples of task-based metrics (for image generation) include
   - classification accuracy,
   - segmentation accuracy,
   - captioning BLEU score, or
   - retrieval precision and recall.
6. The effectiveness of task-based metrics hinges on the choice and performance of downstream models and may not encompass the broader aspects of sample generation.

Quantitative methods

Novelty-Based Metrics

Novelty-Based Metrics

1. These metrics measure the **novelty** and **diversity** of generated samples in comparison to existing ones within the same or different domains.
2. Novelty-based metrics provide insights into the creativity and originality of the generative model.
3. Examples of novelty-based metrics include
   - nearest neighbor distance,
   - coverage, or
   - entropy.
4. While these metrics highlight creativity, they may not consider the realism and relevance of the created sample and might favor unrealistic or irrelevant results.

Quantitative methods

Statistical Tests

Statistical Tests

1. Statistical tests have long been used to determine whether two sets of samples have been generated from the same distribution.
2. These types of statistical tests are called two sample tests.
3. Define null hypothesis as the statement that **both set of samples are from the same distribution**.
4. We then compute a statistic from the data and compare it to a threshold, and based on this we decide whether to reject the null hypothesis.
5. Statistical tests have their own advantages and disadvantages:
   - Users can specify Type 1 error (the chance they allow that the null hypothesis is wrongly rejected).
   - Statistical tests tend to be computationally expensive and thus cannot be used to monitor progress in training; hence they are best used to compare fully trained models.

Another classification for metrics

1. Several metrics have been proposed for evaluation of generative models (Thanh-Tung and Tran 2020).
2. Divergence based evaluation metrics
   - Inception score
   - Fréchet inception distance
   - Neural net divergence
3. Precision-Recall based evaluation metrics
   - k-means based Precision-Recall
   - k-NN based Precision-Recall
4. Other evaluation metrics
   - Metrics for class-conditional models
   - Topological/Geometrical approaches
   - Non-parametric approaches

Summary

Learning deep latent variable models

1. Marginal distribution on x obtained by integrating out z

$$p(z) = \mathcal{N}(z; 0, I)$$
$$p_\theta(x) = \int_z p(z)p(x|f_\theta(z))$$

2. Problem: Evaluation of $p_\theta(x)$ intractable due to integral involving flexible non-linear deep net $f_\theta(z)$.

3. Solutions: by different unsupervised deep learning paradigms

   - Avoid integral: Generative adversarial networks (GAN)
   - Approximate integral: Variational autoencoders (VAE)
   - Tractable integral: Constrain $f_\theta(z)$ to invertible flow.
   - Avoid latent variables: autoregressive models

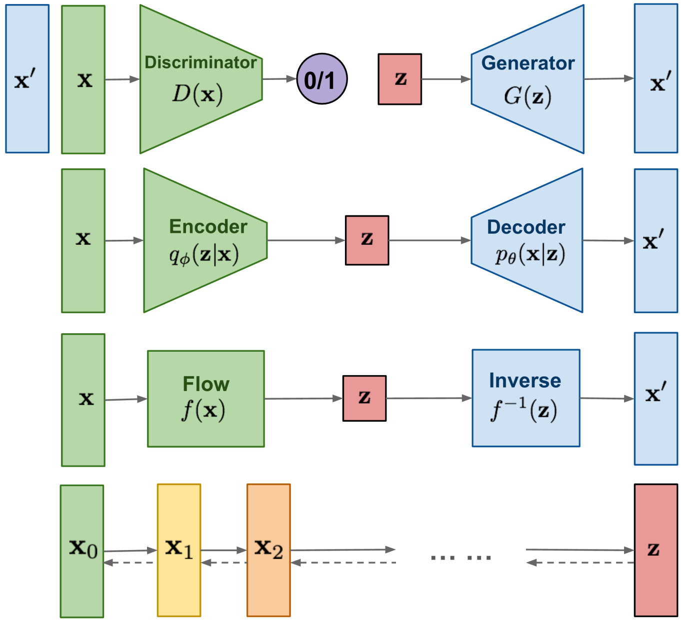
Different generative models using latent variables

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



Some metrics used to evaluate generative models

1. Average Likelihood
2. Inception Score
3. Frechet Inception Distance (FID)
4. Precision and Recall
5. Perceptual Path Length (PPL)
6. Generative Adversarial Metric (GAM)
7. Spectral Analysis
8. Classifier Two-Sample Tests
9. Classification Accuracy
10. FCN Score
11. Nearest Neighbors
12. Time to Distinguish Real and Fake Images
13. Hype and Hype Infinity
14. Disentanglement Analysis

References

Reading

1. Paper Pros and Cons of GAN Evaluation Measures (Borji 2018).
2. Paper Assessing Generative Models via Precision and Recall (Sajjadi et al. 2018).
3. Paper Precision Recall Cover: A Method For Assessing Generative Models (Cheema and Urner 2023).
4. Section 20.4 of Probabilistic Machine Learning: Advanced Topics (Murphy 2023).

Borji, Ali (2018). "Pros and Cons of GAN Evaluation Measures". In: *CoRR* abs/1802.03446.

Cheema, Fasil and Ruth Urner (2023). "Precision Recall Cover: A Method For Assessing Generative Models". In: *International Conference on Artificial Intelligence and Statistics*, pp. 6571–6594.

Murphy, Kevin P. (2023). *Probabilistic Machine Learning: Advanced Topics*. The MIT Press.

Sajjadi, Mehdi S. M. et al. (2018). "Assessing Generative Models via Precision and Recall". In: *Advances in Neural Information Processing Systems*, pp. 5234–5243.

Salimans, Tim et al. (2016). "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*, pp. 2226–2234.

Thanh-Tung, Hoang and Truyen Tran (2020). "Toward a Generalization Metric for Deep Generative Models". In: *arXiv* abs/2011.00754.

Questions?