GAN Precision-Recall Evaluation

Lex Meulenkamp
4721071

Abstract

1. Introduction

The evaluation of generative adversial networks (gan) is a challenging task. For gan evaluation the Fréchet Inception Distance (FID) is a popular metric. The authors of FID [1] have also shown that FID captures different kind of image artifacts. However, since it's a one dimensional score, the distinction between image fidelity and image diversity performance caan be hard to distinguish. Does a bad FID score indicate a gan with good fidelity samples or a gan with good sample diversity? Answers to such questions can only answered by manual inspection of the generated gan images. Also for detecting what kind of failure modes are present, one still needs to detect it by looking at the generated images. To address some of these issues, the authors from [4] created one of the first practical precision-recall (pr) metric for gan evaluation. On a high level, the pr metric captures a gans image fidelity by precision and image diversity by recall. Inspired by the framework of [4] multiple papers about pr-evaluation in the gan setting were published. Just like [4] the method of [5] outputs pr curve.
While the method from [2] and [3] don't output a curve but output two scores; one for precision and the other one for recall.

To be able to compare all metrics, we must settle on the dimensional output of the metrics. In other words, we must choose to go for either only curves or only 2D points. A starting point would be choose for only 2D points, then we only need to summarize curves into a 2D point. Extending the metrics which output a 2D point to also be able to output a curve methods would require more algorithmic engineering. Still, choosing a method to appropriate summarize the curves for different cases is not necessarily trivial. It could be a research question on its own.
Each evaluated gan will have its own curve for the curve methods. Scaling the experiments to evaluate a large set of gans will clutter our plots. Thus, we do really need to summarize here as well. The method from [2] outputs for each evaluated gan one precision score and one recall score. The method from [3] outputs for each evaluated gan one density score and one coverage score. This paper focus/compares the method from [2] and [3]

1.1. Current Questions

Main focus for now:

- *How does fake samples with a large radii within the fake manifold impact the recall/coverage scores?*
- *Impact of k for the metrics?*

Follow-up questions/related:

- *Do Gans tend to create unrealistic yet diverse data?*

2. Methods

For precision and recall metrics in the gan setting, we are comparing the similarity between two distributions, namely the real dataset R and the fake dataset F. Both methods do inference based on the samples distances to a specific (approximated) manifold. Those manifolds M are created by the knn algorithm and can be mathematically expressed as a union of ball objects (circles in 2D):

$$M(X_1, \ldots, X_n) = \cup_{i=1}^{N} B(X_i, NND_k(X_i) \quad (1)$$

Where B(x,r) is Euclidean n-ball space defined by center x and radius r. $NND_k(X_i)$ denotes the distance to the kth-nearest neighbour for sample X_i.

2.1. KNN

The pr metric by [2] is a two dimensional pr score. Where the k from the knn algorithm is a flexible hyperparameter and the feature extractor to go from image to feature space. ~~Both the manifolds from dataset real R and fake F are approximated by the knn algorithm.~~ These manifolds are used to estimate the pr score. For instance, a fake sample can be either rejected or accepted based on its distance from the approximated real dataset manifold. More specifically, the fake sample must have at least one real sample where its distance to this samples is smaller or equal to the real samples kth (real sample) neighbour.

First, the image samples are projected into a more meaningful space for the knn algorithm. For images, pretained cnn's as feature extractor are often used as starting point, but this aspect of the algorithm is domain dependent. Second, once the feature space is established the knn algorithm is used to approximate the manifold M.
Precision counts a fake sample as positive if it is contained in at least one real neighbourhood sphere. Based on the fake sample acceptation-rejection rate by the real manifold (knn approximated spheres) we can calculate precision as:

Precision =

$$\frac{1}{|F|} \sum_{j=1} \mathbb{1}_{F_j \in M(R_1, \ldots, R_N)} \quad (2)$$

The same goes for recall but our estimation is reversed, thus recall counts a real sample as positive if it is contained in at least one fake neighbourhood sphere.

Recall =

$$\frac{1}{|R|} \sum_{i=1} \mathbb{1}_{R_i \in M(F_1, \ldots, F_N)} \quad (3)$$

2.2. Density and Coverage

The density and coverage metric by [3] is also a knn based metric which outputs a two dimension pr score, which is largely based on the pr metric from [2]. They argue that the fake manifold which is used by [2] to estimate recall is often unreliable.
"Since models tend to generate many unrealistic yet diverse samples, the fake manifold is often an overestimation of the actual fake distribution". [3]
One could argue if the word "tend" is appropriate in the context of gan algorithms. The general behaviour of gans is highly volatile to hyperparameters, architecture, loss functions, gan tricks etc. One of the main problems of gans is a lack of diversity while the image fidelity performs better than most generative algorithms. However, if a gan generates diverse (unrealistic noisy) samples than the recall estimation approach by [2] becomes problematic. In such a case, the radii of multiple spheres from the fake manifold are quite large, thus increasing the chance that a real sample will be accepted even though its smallest distance to a fake sample in the feature space might be large. Instead of using recall [3] uses coverage, where coverage counts the fraction of real samples which have at least one fake sample contained in their neighbourhood sphere.

Coverage =

$$\frac{1}{|F|} \sum_{i=1}^{|F|} \mathbb{1}_{\exists j \ s.t. \ F_j \in B(R_i, NND_k(R_i))} \quad (4)$$

Precision metric by [2] counts a fake sample positive if it is contained in any real sample neighbourhood. This way of counting makes it more susceptible for outlier samples. Therefore, [3] proposes to use the density metric instead, where it counts how many real-sample neighbourhood spheres contain fake samples.Therefore, the weight of an outlier sample is reduced. Important to note, this way of counting makes it possible for the density measure to be larger than one.

Density =

$$\frac{1}{k|R|} \sum_{j=1}^{|R|} \sum_{i=1}^{|F|} \mathbb{1}_{F_j \in B(R_i, NND_k(R_i))} \quad (5)$$

3. Experiments

For the experiments radii r refers to distance to the kth-nearest neighbour for a sample within its own data distribution.

3.1. Fake data with diverse samples and constant radii

The purpose of this experiment is to test the influence of fake (unrealistic) diverse data. We don't necessarily care about the type of the fake distribution as long as the radii of the fake samples is relatively large compared to the real dataset samples.

$$\mathrm{R} \sim N\left[\mathbf{0} \in R^D, \quad \mathbb{I} \in R^{DxD}\right]$$

The first fake sample starts at the zero vector with a constant radii. The constant radii determines the minimal distance between this sample and the following one.
Thus, each new samples has:

- No circle overlap; meaning distance $\geq 2 * r$ to its previous sample.
- New direction is randomly sampled angle between 0 and 360.
- Last sample position plus sampled direction is the new location.

The recall score will still be close to one even if we remove a lot of fake samples. For instance, figure 2 the fake distribution sample size is reduced to $|F| = 10$. The space defined by intersection of the fake circles is still large enough to cover most of the real samples which causes recall to be close to one.
Getting a good coverage score is quite difficult once the real dataset is confined in a close space. For instance, if we use the following Gaussian for our experiments.

$$\mathrm{R} \sim N\left[\mathbf{0} \in R^D, \quad \tfrac{1}{10}\mathbb{I} \in R^{DxD}\right]$$
$$\mathrm{F} \sim N\left[\mathbf{0} \in R^D, \quad \mathbb{I} \in R^{DxD}\right]$$

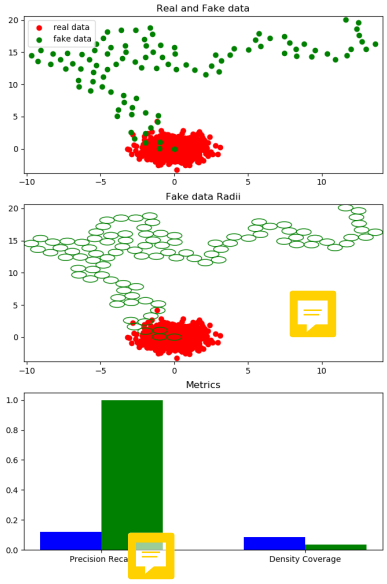


Figure 1: Diverse fake data with constant radii 0.5

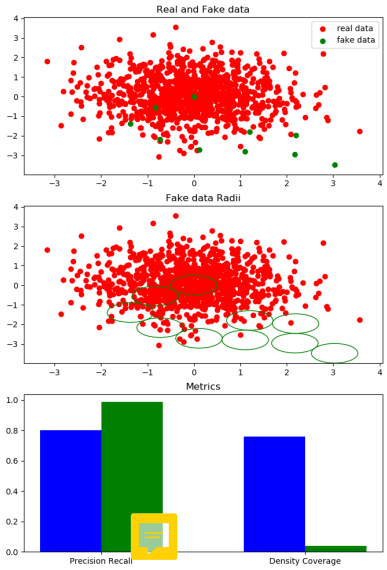
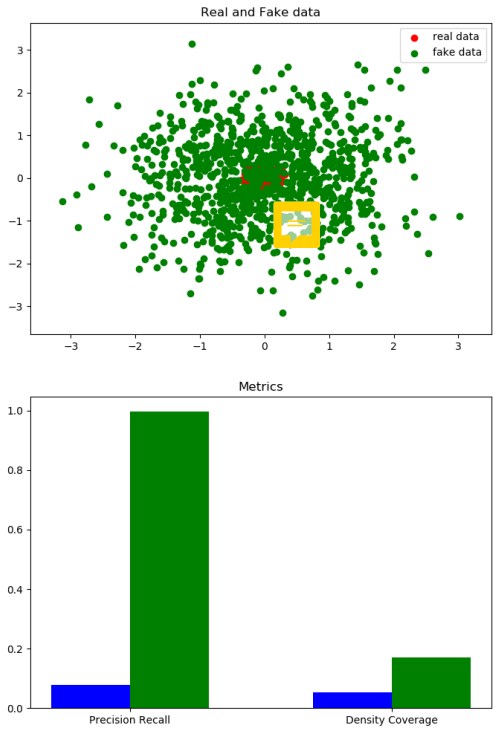


Figure 2: Diverse fake data with constant radii 0.5 and 10 fake samples



$$\text{Real data} \sim N\left[\begin{pmatrix}0\\0\\\vdots\\0\end{pmatrix} \in R^D, \quad \mathbb{I} \in R^{DxD}\right]$$

$$\text{Fake data} \sim N\left[\begin{pmatrix}0\\0\\\vdots\\0\end{pmatrix} \in R^D, \quad x\mathbb{I} \in R^{DxD}\right]$$

$$x \in \mathbb{R}$$

$$D \in \mathbb{N}^+$$

Figure 3: Dense Gaussian for R

3.2. Scale experiments

The following experiment include the metrics which rely on the knn algorithm from section 2.1 precision and recall and 2.2 density and coverage for manifold estimation. For this experiment we want to test the volatility of those metrics once the fake data attains high levels of variance compared to the real distribution. As the authors from method 2.4 [3] mention *"Since models tend to generate many unrealistic yet diverse samples, the fake manifold is often an overestimation of the actual fake distribution"*.
We would like to find out in which circumstances the fake manifold becomes too diverse/unreliable too rely on it for inference. This is done by increasing the x scale factor – which is multiplied over the identity matrix – over time.
Important to note: Recall from 2.3 relies on fake manifold inference and Coverage from 2.4 doesn't.

4. Appendix

4.1. Different Metrics

4.2. Kmeans method

[4] is one of the first papers with both a theoretical and practical algorithm for precision and recall curve analysis for gans. Intuitively, we express both distributions as one component which can be generated by the other distribution, and the other component which can't be generated by the other distribution. Those components are mainly composed by the sets created by the intersection operator.
First the support sets:
$A = supp(R) \cap supp(F)$.
$B = supp(R) - A$.
$C = supp(F) - A$.

Based on the support sets, there are multiple distributions configurations possible for the three probability distributions μ_A, μ_B, and μ_C. Where the underscore denotes the support set of the probability distribution. A precision α and recall β pair, both α and $\beta \in (0, 1]$, are defined by the metric if the following equations are true.

$$R = \beta\mu_A + (1 - \beta)\mu_B \tag{6}$$

$$F = \alpha\mu_A + (1 - \alpha)\mu_C \tag{7}$$

For instance, perfect recall $\beta = 1$ is only possible if R can be decomposed just by the intersection distribution μ_A.

Getting the precision α and recall β values from equation 6 and 7 is difficult because we have to check whether there are suitable distributions μ_A, μ_B, and μ_C for all possible precision α and β values. The practical approach by [4] get the precision-recall pairs by using the kmeans method on the union feature space by a cnn from datasets R and F.
In more detail, by fixing $\alpha = \lambda\beta$ we can iterate over multiple λ values to obtain precision and recall pairs for a certain weighting factor λ. Equations 8 and 9 are used to calculate precision and recall pairs which are intersection operation equations for discrete histograms. Those histograms are approximated by the kmeans algorithm on the union feature space. For each dataset, we obtain its histogram by counting the amount of samples per cluster, thus its histogram is a one dimensional vector with its length equal to the amount of k clusters from kmeans. Those count histograms can then be normalized into a density histogram which essentially is an approximated distribution.

$$\text{Precision}(\lambda \in \Lambda) = \sum_{\omega \in \Omega} min\big(\lambda R(w), G(\omega)\big) \quad (8)$$

$$\text{Recall}(\lambda \in \Lambda) = \sum_{\omega \in \Omega} min\big(R(\omega), \frac{G(\omega)}{\lambda}\big) \quad (9)$$

$$\Lambda = \left\{ \tan\left(\frac{i}{(m+1)} . \frac{\pi}{2} \right) \mid i = 1, 2, \ldots, m \right\} \quad (10)$$

$$\textit{Curve} = \left\{ \Big(\text{Precision}(\lambda), \text{Recall}(\lambda) \Big) \mid \lambda \in \Lambda \right\} \quad (11)$$

All the $\lambda \in \Lambda$ form an equiangular grid of values which are used to obtain the precision and recall pairs from equation 8 and 9. Where $m \in \mathbb{Z}^+$ is the radial resolution which determiners the amount of precision-recall pairs we sample for the curve. Also looking at it differently, $\alpha = \lambda\beta$ is essentially a line equation such as $x = \lambda y$ where the line intersection with the distribution is the precision recall pair we need.

4.3. Classifier method

The authors from [5] state that the histogram approach by [4] is restricted to discrete probability distributions. They remove this restriction by defining precision and recall curves for arbitrary probability distributions. The precision recall curve is defined as mixed error rate of binary classifiers obtained as likelihood ratio tests. Where the main idea is that the mixture data from the real and fake dataset is hard to classify if the gan has learned the real distribution. This will result into a high type I and type II error for the evaluation classifier. Those errors rate are used to obtain multiple precision recall pairs.
For instance, we have a mixture dataset from real and generated data Y, labeled generated with $Y = 0$ and real with $Y = 1$. A classifier $\hat{Y}$ which is trained on Y. Just as the kmeans method we use $\lambda \in \Lambda$ equation 10 values to get different precision-recall pairs. The weighting of the λ values is now used to weight the type I and type II error instead of the histograms.

$$\text{Type I error} = \alpha = P(\hat{Y} = 0 \mid Y = 1)$$
$$\text{Type II error} = \beta = P(\hat{Y} = 1 \mid Y = 0)$$
$$\text{Precision} = \lambda\alpha + \beta$$
$$\text{Recall} = \alpha + \frac{\beta}{\lambda}$$

4.4. Old experiments

Determine if we needs these for the future.

4.5. Variance drop

Sample diversity is often a problem for gans. The gan loss-function does not directly incentives for more sample diversity. Just a few very convincing generator samples can get the gan stuck in a local-minimum. **(Find quotes to support given statements)** A related diversity problem in gan, namely mode collapse is also often studied, where parts of the real distribution can't be generated by the gan. Gans often end up in a state where the generated samples form a (small) subspace of the real distribution.**(Prefer a specific quote to make define mode collapse more rigidly)**. Therefore, experiments which test the metrics sensitivity for diversity differences between distributions is important.
For the experiment, the real and fake distributions are both sampled from a multivariate Gaussian:

$$\text{Real data} \sim N\left[\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in R^D, \quad \mathbb{I} \in R^{DxD} \right]$$

$$\text{Fake data} \sim N\left[\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in R^D, \quad A = diag(x) \in R^{DxD} \right]$$

$$x = \big[x_0, \ \ldots, \ x_D\big]$$
$$x_i \sim \text{Ber}(0.5) \quad D \in \mathbb{N}^+$$

The density of the fake distribution becomes more (compact) dense as the experiment x vector contains more zero's. However, the mean vector from the fake distribution remains the same as the real distribution. So most samples from the fake distribution should still be generated by the real distribution.**(Test how true this statement is? or word it differently based on experiments)**
Thus, the expectation for the metrics in general is that recall will at least decrease significantly faster than precision. That means that the beta values lower than 1 (precision more weight) have high f scores and beta values higher than 1 (recall more weight) have low f scores.

Among the four metrics, k-means shows an inverse evaluation compared to the rest. Meaning that precision decreases harder than recall. This example also shows why taking the maximum is not necessarily a good approach for all evaluation settings. For traditional pr-evaluation we might want

to take the classifier that gives the maximum f-score, but its important to keep in mind that in this evaluation setting, one pr-curve and all the score derivations from it corresponds just to one single system (gan). For some settings a different (e.g. median) approach might capture the general evaluation consensus better among most pr-points. However, taking the maximum from a set f-scores is not necessarily better/worse than the median. For most cases, we can only analyze which approach is appropriate given that we have (almost) complete knowledge of both distribution. The variance between the approaches can be significant. For instance, if we take default $\beta = 8$ value then by maximum approach we get:
$Recall = Maximum(F_8) \approx 0.8$ and $Precision = Maximum(F_{\frac{1}{8}}) \approx 0.16$. While the median gives:
$Recall = Median(F_8) \approx 0.2$ and $Precision = Median(F_{\frac{1}{8}}) \approx 0.16$

4.6. Mean shifts

First we start with two identical multivariate Gaussian distributions real and fake. The mean shifts experiments shifts the Gaussian mean by a fixed translation vector.

4.7. Mode dropping

[3] did an Gaussian mixture experiment where the real dataset and fake dataset are identical at first. The fake dataset gradually drops all mode data simultaneous except for the first mode. The following experiment is the same but with some slight variations.
The experiments are generated by a Gaussian mixture $\in R^D$ with 10 modes. Each mode uses the same identity matrix I^{DxD} and the mean vectors u^D are sampled uniformly by $U(-1, 1)$, where $D \in \{2, 4, 6, 8, 16, 32, 64\}$.

References

[1] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[2] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pages 3927–3936, 2019.

[3] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo. Reliable fidelity and diversity metrics for generative models. *arXiv preprint arXiv:2002.09797*, 2020.

[4] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, pages 5228–5237, 2018.

[5] L. Simon, R. Webster, and J. Rabin. Revisiting precision and recall definition for generative model evaluation. *arXiv preprint arXiv:1905.05441*, 2019.