GAN Precision-Recall Evaluation

Lex Meulenkamp
4721071

Abstract

1. Introduction

2. Research direction

[1] compared their precision-recall (pr) metric with the metric from [3]. The main difficulty is that the latter outputs pr value pairs to form the pr curve, while the former outputs a two dimensional pr score. To be able to compare those two metrics; the metric output from [3] must also be summarized into a two dimensional pr score. Another option would be to extend the metric from [1] to be able to output pr curve.
Just as in traditional pr evaluation we can summarize the pr curve with (generalized) f-scores. The default value used by [3] is $\beta = 8$, which means that recall is valued 8 times as important as precision. The following formula with the β value is used to calculate f-scores for each pr pair in the pr curve. The maximum of the set of f-scores is taken in order to output a two dimensional pr point.

$F_\beta = (1 + \beta^2)\frac{precision * recall}{(\beta^2 * precision) + recall}$
Precision $= F_{\frac{1}{\beta}}$
Recall $= F_\beta$

This has also been adapted by [1] for their comparison experiments. Whether the default beta value is a appropriate choice to summarize the evaluation power of the pr curve has not been experimented and/or reported.

2.1. Current Questions

The pr curve in practice is a set of multiple two dimensional pr pairs. Curve evaluation contain a lot of information about a system, but in this setting each gan has its own pr curve. Thus, it can get cluttered if we analyze a large set of gans. So before we do curve analysis we might want to start first analyzing at a higher level. Meaning, summarizing the curves by a single two dimensional point. Thus, we need to decide on one β value for generating f-scores and we need a statistic (e.g. max, median, avg) to summarize those f-scores in order to output the final two dimensional pr score.

- *What kind of statistic should be used to summarize multiple f-scores into one pr score?*
- *What kind of beta value should be used or is it just a matter of preference between recall/precision. (Maybe a range of trustworthy beta scores)*
- *Does the distribution of f-scores generated by multiple beta values give insightful information for choosing a beta value and statistic?*
- *Extra, can we say something interesting about the different metrics?*

A central question which is most likely important for all the questions above:
Can we analyze when certain choices fail/succeed under certain experimental conditions?

3. Methods

Compact explanation of methods.

3.1. Kmeans method

We have a mixture dataset from real and generated data Y which is clustered by kmeans. For each cluster, there is a count how many samples are coming from the real and generated dataset. Those counts are used for calculating the density histograms — bins equal to k-clusters — for the real and generated dataset.
Those two (density) histograms R and G are used to calculate the precision-recall pairs. We also create λ values for weighting the histograms to gain different precision-recall pairs. λ is linearly spaced angles between [0, pi/2]

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

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

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

3.2. Classifier method

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 1 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.

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

3.3. KNN

PR method by [1] makes manifolds by using k-nearest neighbours to estimate manifolds for datasets. According to [1]: *"The key idea is to form explicit non-parametric representation of the manifolds of real and generated data"*.
First, the image samples are projected into a more meaningful space for the distance metrics (knn). For images, using cnn as feature extractor is the most used method. The following function is used to determine a samples binary membership:
Where ϕ is one feature vector and Φ a set of feature vectors.

$$f(\phi, \Phi) = \begin{cases} 1, & \text{if } ||\phi - \acute{\phi}||_2 \leq ||\acute{\phi} - NN_k(\acute{\phi}, \Phi)||_2, \\ .. & \text{for at least one } \acute{\phi} \in \Phi \\ 0, & \text{otherwise} \end{cases}$$

$\text{Precision}(\Phi_r, \Phi_g) =$

$$\frac{1}{|\Phi_g|} \sum_{\phi_g \in \Phi_g} f(\phi_g, \Phi_r) \quad (2)$$

$\text{Recall}(\Phi_r, \Phi_g) =$

$$\frac{1}{|\Phi_r|} \sum_{\phi_r \in \Phi_r} f(\phi_r, \Phi_g) \quad (3)$$

3.4. Density and Coverage

The PR method by [2] is also a knn based method to estimate the manifold for a dataset.
According to [2] *Precision counts the binary decision of whether the fake data is contained in any neighbourhood sphere of the real samples. Density, instead, counts how many real-sample neighbourhood spheres contain fake samples* $(\Phi_r, \Phi_g) =$

$$\frac{1}{k|\Phi_g|} \sum_{\phi_g \in \Phi_g} \sum_{i=1}^{|\Phi_r|} f(\phi_g, \Phi_{r_i}) \quad (4)$$

Coverage is the ratio of the real samples that are covered by the fake sample, defined as
$\text{Coverage}(\Phi_r) =$

$$\frac{1}{|\Phi_r|} \sum_{\phi_r \in \Phi_r} 1 \text{ if } \exists j \text{ s.t. } \phi_j \in \Phi_g \text{ is in neighbourhood of } \phi_r \quad (5)$$

4. Experiments

4.1. 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:

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

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

$$x = [x_0, \ldots, x_D]$$

$$x_i \sim \text{Ber}(0.5) \quad D \in \mathbb{N}^+$$

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 $Precision = Median(F_{\frac{1}{8}}) \approx 0.16$

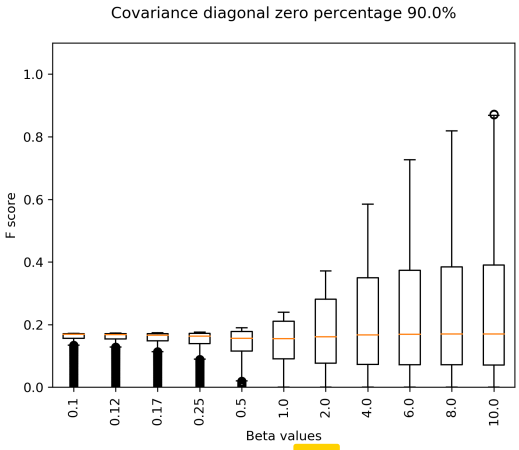

Figure 1: K-means method (k=20)

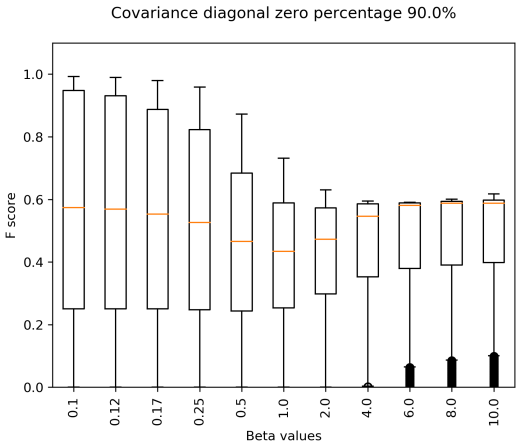

Figure 2: Classifier method (logistic model)

TODO classifier methods results The classifier plots of the distribution are more left-skewed. Also, if we take a classifier which has an easy time to distinguish the samples the

error rate are low, thus also all the other linear combination, resulting in a degenerate pr curve (0,0). In such a case, we might argue that the classifier (adaboost) is too strict.

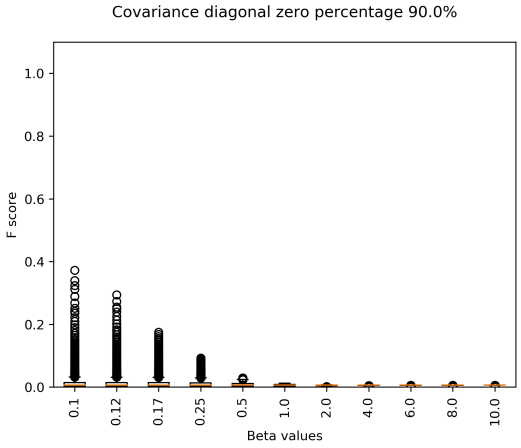

Figure 3: Classifier method (ada boost)

4.2. 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.

4.3. Mode dropping

[2] 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 $D \in \{2, 4, 6, 8, 16, 32, 64\}$.

TODO: shorter or gone
Note that:
As the dimensions scale up, there is more room for the mode centers to be sampled further apart by $U(-1, 1)$. Missing samples due to simultaneous mode dropping should be easier to spot by th e metrics in high dimensions, which should results generally in lower precision-recall pairs.
In the Appendix section other dimensions are displayed.

Remarks
The Gaussian clusters have a lot of overlap in the problem where $D = 2$. This makes it for every metric more difficult to notice the drop changes. A better classifier (random forest) helps in such a case as it has more complex decision boundaries compared to logistic regression. The cluster method has in general more troubles to recognize the gradual drop. For most experiments it is less sensitive to the gradual dropping of fake samples.

Remarks about experiment setup:
Instead of dropping data for fake, try to resample the fake dataset until both sets are balanced. Resampling fake then happens with adjusted mode weights until the fake dataset is as large as the real dataset.

- Find or test/show multiple aggregations for pr curves.
- Test if single point pr-method can be adjusted to also make pr curves
- split images once analysis are done

Experiments Included Synthetic multivariate Gaussian data.

- Decrease covariance matrix;
  to simulate fake data in a subspace of the real data.
- Shifting of mean vector
- Robustness to noise
- Gradual mode dropping
- Outlier test

5. Appendix

References

[1] 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.

[2] 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.

[3] 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.