Proposal: Marginal GAN Part 2

Kellin Pelrine
Department of Economics
Yale University
New Haven, CT 06520
Kellin.Pelrine@Yale.edu

Abstract

In the first part of this project, I trained GANs on individual classes and compared the output with normal GANs. Qualitative results were reasonable and quantitative metrics, particularly the class-aware Fréchet distance 7, suggested that this improved performance. However, experimental results were somewhat limited (MNIST and F-MNIST only) and there were no conceptual results. For the next part of the project, I propose to expand the experimental results in 3 directions. First, more datasets (CIFAR-10, SVHN, CIFAR-100). Second, compare with two GANs that uses labels (CGAN [8] and Class-Splitting GAN [5]). Third, examine if marginal GANs can be trained faster in real-time than normal GANs (with sufficient parallelization). I also propose a conceptual explanation for observed performance gain from this and other label-based methods, based on a tradeoff between taking advantage of separated classes and taking advantage of mutual information between classes. I test this explanation with a toy model and obtain consistent preliminary results.

1 Introduction and Summary of Part 1

The idea of Marginal GAN is very simple: instead of one GAN for all the data, train separate GANs on individual classes (or subsets of classes). Surprisingly, this idea does not seem to have been studied, even though using labels in any way typically improves GAN performance [3–5]. In the first part of this project, I took the WGAN code and parameters from [10] and trained one GAN per class in MNIST and F-MNIST. I compared it with the original WGAN, i.e. training on all classes simultaneously. The marginal GAN output was reasonable under human inspection. The quantitative results are shown in tables 1 and 2, reprinted from the project report.

Table 1: Quantitative Results: MNIST

Name	CAFD	FID	IS
Original GAN	107.4	40.1	2.23
Marginal GAN	90.7	39.7	2.14

Table 2: Quantitative Results: FMNIST

Name	CAFD	FID	IS
Original GAN	42.9	107.8	4.16
Marginal GAN	35.8	110.8	4.25

Preprint. Work in progress.

Here higher is better for the Inception score (IS) [9] and lower is better for the class-aware Fréchet distance (CAFD) [7] and Fréchet Inception distance (FID) [6]. The FID and IS scores are similar in each case. These metrics are included for comparison because they have been very popular, but they also suffer from substantial robustness problems and other issues [2, 1, 6, 7]. The CAFD is significantly better with marginal GAN. This metric is also the most relevant of the three, as it improves on many of the problems with previous metrics [7].

For further details on these metrics and the results of part 1, please refer to my project report.

The improved CAFD, along with equal FID and IS, suggest that training one GAN per class can be superior to the standard method of training one GAN for all classes. However, these results are limited because they only include two relatively simple datasets. They also only compare marginal GAN, which requires labels, with a GAN that does not use labels at all.

For the next part of this project I propose to expand these experimental results with more datasets and compare with two GANs which use labels (CGAN [8] and Class-Splitting GAN [5]). I also want to evaluate if training can be done faster in real-time with this technique than with standard GANs, which could be a useful practical result.

A significant limitation of the previous work on this project was a lack of theory or conceptual ideas for what was going on and why marginal GAN could yield performance improvements. In this proposal I include a conceptual explanation which can also give insight on why labels boost GAN performance in general. I test this with a toy model. Preliminary results are consistent with the proposed explanation.

The rest of this proposal is structured as follows: first, I discuss the new conceptual ideas and results. Next, I discuss the proposed experiments. Then I conclude with potential challenges to overcome and other possible directions for research.

2 Conceptual Ideas

Consider the following scenario. One wishes to estimate a mixture of two Gaussians. There are two options: one can either separate the points by "class," and estimate each Gaussian individually, or one can estimate them with all the data pooled together but potentially with some additional information on the relationship between the two Gaussians. For example, the means might be symmetric about 0 and the variance might be the same. More generally, the means and variances might be drawn from some common distribution. Which option is best?

The optimal option depends on the information gained from separating the classes (corresponding to marginal GAN) vs. using the shared or correlated information between classes (corresponding to original GAN). Small distance between classes increases the benefit of separating them (because if not, it's more difficult to distinguish which data point corresponds to which class). On the other hand, a strong correlation structure between class distributions, which the estimator can take advantage of, increases the benefit of pooling them.

These two effects can interact with the sample size. Suppose one is just estimating the means of an equally weighted two-Gaussian mixture with identical known variance. Suppose that the "separating" estimators (when we separate the data corresponding to each Gaussian) are the sample means, a natural choice. Imagine also we make a poor choice for the pooled estimator, ignoring the fact that the data comes from two Gaussians, and just fit a single Gaussian using the (pooled) sample mean. Nonetheless, even with this poor choice of pooled estimator, if the sample size is tiny - in the extreme case, one point from each Gaussian - and the two true means are close together, it may be better to fit a single Gaussian in order to reduce the variance. As the sample size grows, it will eventually be better to switch to the separating estimator.

The situation is more complicated in a real GAN scenario, because it is not clear how "close" or "correlated" different classes may be, and how that interacts with the GAN's structure and learning process. However, this line of thought would explain why labels seem to almost always improve performance [4]: given that we often place significant importance on human-identifiable output classes, there is probably a substantial amount of information to be gained from knowing which data points correspond to which class. Conversely, this suggests that since marginal GAN is at the other extreme, it may suffer from not taking advantage of any cross-class information. The experimental results, especially compared to other GANs that use labels, can indicate whether existing GANs go far enough in separating classes.

In order to test this idea in a controlled GAN setting, I constructed a toy model based on code (not mine) available here: `https://github.com/MatthieuBizien/AdversarialNetworks`. The original code constructs a simple GAN that estimates a Gaussian distribution or mixture of two Gaussian distributions. It uses a logistic regression discriminator and updates the generator directly using gradients of the likelihood function.

I wrote new code within this framework to estimate a Gaussian "quasi" mixture model - an equally weighted mixture of two Gaussians with identical variance and means that are symmetric about 0. I then compare a separating estimator - separating the data depending on which distribution it came from and estimating each Gaussian individually - with a pooling estimator. The pooling estimator does not know which distribution each data point is from, but knows the correct likelihood function and thus only estimates one variance and one mean (which is reflected to give the second mean).

The results of two tests are shown in figures 1 and 2. In both, the true density is shown in black, while the pooling estimate is in red and the separating estimate is in blue.

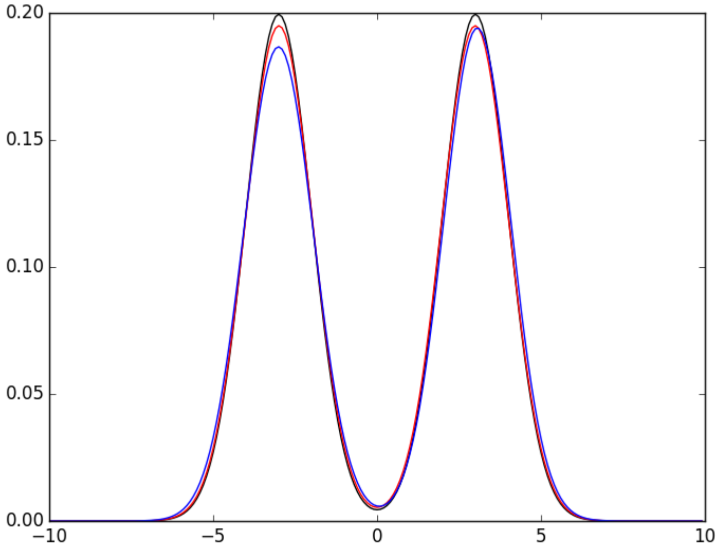

Figure 1: Means +3 and -3, Variance 1

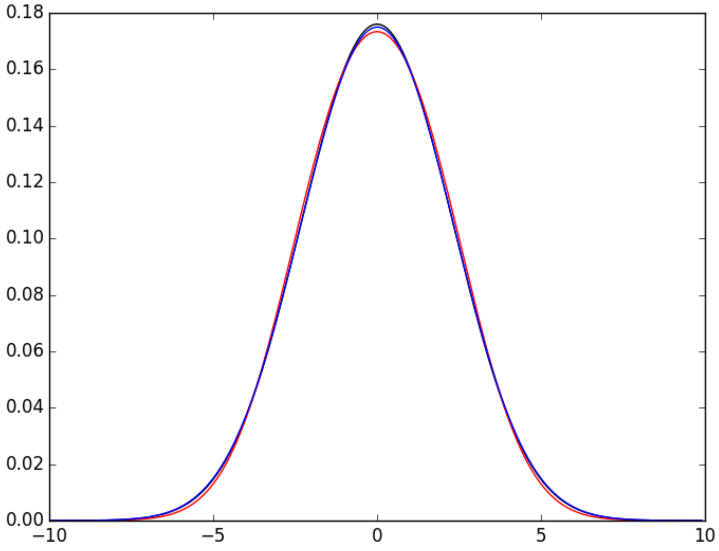

Figure 2: Means +1 and -1, Variance 4

In the first case, the pooling estimator (red) outperforms the separating estimator. This is not surprising because there is very little additional information gained from knowing which distribution each data point came from, because the separation of the means is large enough (and the variance small enough) that there is almost no overlap. Therefore, taking advantage of the symmetry gives a better estimate.

In contrast, in the second case, the distributions are much closer and there is substantial overlap (illustrated in figure 3, which shows the separating estimators individually). Here the separating estimator outperforms the pooling estimator, even with a massive sample size (I tested a variety; the graph shown uses 1000 draws from 1 million samples per distribution). In fact, although the pooling estimator's distribution is still close, the parameters are very far off, as it thinks the means are +1.266 and - 1.266, and the variance is 1.793. In addition to illustrating the ideas above, the huge sample size and substantial difference from the true parameters suggests that the GAN may get stuck in a local optimum.

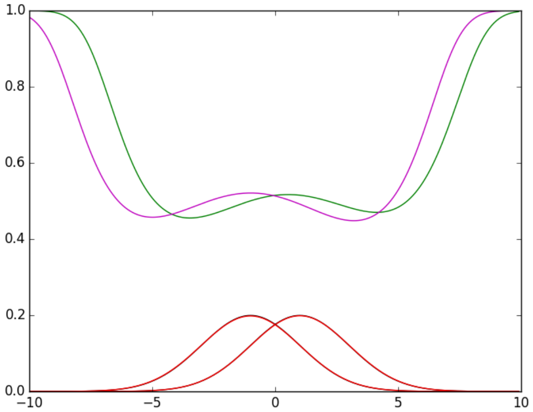

Figure 3: Means +1 and -1, Variance 4. Purple and green lines are the discriminators. The true densities are present in black, but are indistinguishable from the estimates.

These ideas and examples are hopefully illustrative, but the presentation could be improved. On the experimental side, rather than just showing two cases, it would be more convincing to include a metric like an estimate of the mean integrated squared error. Ideally, I would also like to show a transition from the sample size, where an identical data generation process results in the pooling estimator performing better at some sample size and the separating estimator at another.

3 Proposed Experiments

The most obvious way to extend the project's existing experimental results is to test more datasets, since both MNIST and F-MNIST are fairly simple black-and-white images. The next dataset I propose to test is CIFAR-10. As the name indicates and like MNIST/F-MNIST, this dataset has also has 10 classes. However, these images are in color, so it is substantially more challenging.

Another dataset is SVHN, which like MNIST contains small images of numbers, but has more complexity and includes color like CIFAR-10. These pictures are house numbers taken by Google street view. The difficulty is probably not higher than CIFAR-10, perhaps lower, but this will provide another data point. This dataset appears to be somewhat less commonly used than CIFAR-10 though, potentially making comparisons more difficult, which is why I put it in the fourth slot for testing rather than the third.

Beyond these two, I would also like to test CIFAR-100. This dataset is structured like CIFAR-10 but has 10 times more classes and 10 times fewer images per class. This may present additional challenges - for example, will the small sample size per class prevent training a marginal GAN on each class? - so I am less confident here. But failure to get a working marginal GAN may also yield useful information.

In addition to more datasets, it also seems important to compare marginal GAN with other GANs which use labels. A simple, standard one is Conditional GAN [8]. Another one, which tries to create even more labels than are provided, is Class-Splitting GAN [5]. Depending on the results, others may be worth testing as well.

All the experiments above can be evaluated quantitatively using the same three metrics as before.

I also want to test if marginal GANs can be trained faster than normal GANs. If separating classes creates simpler subproblems, perhaps computation speed can be improved? If so, this would be a practical result, because there are situations where a practitioner may have a surplus of parallel computation available but a deficit of real-life time to complete the training (e.g. to meet an upcoming deadline, or because of the scale of the problem at hand).

The first and simplest way to test this is by reducing the maximum training time until performance of the GANs decay (as evaluated by the usual metrics) and comparing the marginal ones with the original. This method may not give the whole story but is certain to work and give some results.

A more complicated way to test this is by adjusting the hyperparameters of the GANs. Perhaps the marginal GANs can work with a smaller network size, which would thus be faster to evaluate. This is trickier to test, since there are many parameters to tune and no clear guide, but depending on the results of the previous test might also be worth testing (especially some simple modifications like decreasing the size of every layer by a constant factor).

4 Potential Challenges and Further Extensions

For the conceptual side of the project, the goal is to present a clear argument and evidence that it matches how GANs work in practice. Although the former part also needs work, the latter is particularly challenging. I hope that the toy model described above can provide a reasonable illustration, but this is open to debate, and there may be a better approach.

More ambitiously, there might be some connection here to mode collapse. In Figure 2, the two modes constructively interfere and combine into a single mode. The pooled GAN also fails to effectively capture the parameters of this distribution, and in particular underestimates the variance. If this occurred simultaneously in many dimensions, could this collapse the variance dramatically? Could there be a cascading effect if other dimensions were correlated with this one? I do not have a clear idea here yet, but there might be potential for further investigation.

On the empirical side, the main challenge will be coding and implementing all these different GANs. I plan to use existing code or parameters, with modification as needed, but debugging can always be surprisingly hard.

Another challenge is evaluating GAN performance in the best way. As argued in the part 1 report, the CAFD appears to be a good metric, but there are many others and more being created. At some point, a second pass over the literature on quantitative GAN metrics may be worthwhile, both to reconsider existing metrics and check for new information.

Overall, the aim of this project is to give insight on how to better train GANs. The first part suggested there was something here, since marginal GAN seemed to outperform the standard GAN. But results were limited. The second part of the project, as proposed here, will hopefully provide substantial evidence on how well marginal GAN works and also offer an explanation for the results.

5 Note on Replication

I have included the code for the toy model with this proposal. The file that runs is "demo_gaussian_quasi_mixture.py." Output graphs are saved in .png files (Combined, Pooling, Separating). Note the figure title for separating can be misleading because it corresponds to only one of the estimates (the one computed last), not both. A conda specification file (specfile.txt) is included detailing the environment (works with the cluster Grace).

References

[1] S. Barratt and R. Sharma. A Note on the Inception Score. 2018. URL `https://arxiv.org/pdf/1801.01973.pdf`.

[2] A. Borji. Pros and Cons of GAN Evaluation Measures. 2018. URL `https://arxiv.org/pdf/1802.03446.pdf`.

[3] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. 2015. URL `https://arxiv.org/pdf/1506.05751.pdf`.

[4] I. Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. 2017. URL `https://arxiv.org/pdf/1701.00160.pdf`.

[5] G. L. Grinblat, L. C. Uzal, and P. M. Granitto. Class-Splitting Generative Adversarial Networks. 2018. URL `https://arxiv.org/pdf/1709.07359.pdf`.

[6] 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. 2018. URL `https://arxiv.org/pdf/1706.08500.pdf`.

[7] S. Liu, Y. Wei, J. Lu, and J. Zhou. An Improved Evaluation Framework for Generative Adversarial Networks. 2018. URL `https://arxiv.org/pdf/1803.07474.pdf`.

[8] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. 2014. URL `https://arxiv.org/pdf/1411.1784.pdf`.

[9] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. 2016. URL `https://arxiv.org/pdf/1606.03498.pdf`.

[10] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: protecting classifiers against adversarial attacks using generative models. 2018. URL `https://arxiv.org/pdf/1805.06605.pdf`.