Controllable content generation

Cédric Beaulac
Department of Statistical Sciences
University of Toronto
Toronto, ON, Canada
cedric@utstat.toronto.edu

Jeffrey S. Rosenthal
Department of Statistical Sciences
University of Toronto
Toronto, ON, Canada

David K. Duvenaud
Department of Computer Science
University of Toronto
Toronto, ON, Canada

Abstract

We use the recent advances in generative models to a construct controllable content generator that extends the current procedural content generator abilities to democratize artistic content creation. Our goal is to give content creators the power to decide the properties of the content they would like to control, to give them tangible control over these properties and let generative models fill the gaps. Our main contribution is to define what control is and how to measure it. We also propose a model based on the Variation AutoEncoder architecture that is inspired from semi-supervised learning to create a controllable content generator.

1 Introduction

Recently, there has been an increase in the number of distribution platforms for visual arts, short movies, music and video games which gives content creators more tools to share and distribute their content themselves. In order to further help content creators, our current research is an attempt to give them more tools to enhance their content creation abilities.

The algorithmic creation of content allows us to generate an infinite amount of content quickly and inexpensively. Procedural content generation (PCG) [26, 28] is a family of techniques that generate content mostly in video games [1] and movies [14] such as art assets, story elements, dialogues and more. Our goal is not to replace content creators with generative models but rather to use generative models to democratize artistic creation. To achieve this goal, we focus our effort on giving the content creators control over the content generators so that they can truly create what they envision.

2 Control

In order for generative models to be used as design tools, we need to be able to control the generative process but the notion of control is absent in the generative models literature. The main contribution of this article is our proposed definition of control. We define various independent aspects of control that we deem useful for content creators. Given how hard it is to evaluate generative models [27, 13, 2], we try to define control in ways that can be measured. Control variables are at the centre of our proposed definition; these form a set of variables that allows the interaction between content creators and the content created, in other words, these are variables we adjust before generating the content that we use to design something specific. We fix the control **size** to a pre-specified number of control variables. Our definition of control is based on three properties of the control variables.

Preprint. Under review.

First, the control **strength** represents how strong is the effect of the control variables on the content generated. A control variable with high strength has a big impact on the produced content. We decided to use the mutual information [6] utilized to develop recent successful generative models such as InfoGAN [5, 8] and ExprGAN [7] to measure control variables strength.

Next is the **disentanglement** of the control variables which measures how interpretable control variables are and how disjoint are their effects. If we want to give artists true control, we have to make sure that the control variable effects are disjoint and interpretable. Disentanglement has been studied quite extensively recently [4, 21, 3] and researchers are attempting to produce a rigorous definition for it [11]. This recent progress allows us to evaluate the disentanglement of our control variables.

Finally, we will be looking at the **range** of the control variables. Supposing the control variables are disentangled and that they control individual properties of the content, we say that a variable has a large range if we can interpolate between what has been observed and if we can extrapolate outside of the spectre of observed instances. In recent work [23, 15] interpolation is evaluated informally by simply eyeballing the results, but strict evaluation metrics are needed in order to make more precise progress. We use the Inception Score [25] to automatically assess how far from the observed control variable values the generator can extrapolate without losing too much quality.

3 Generative models

In recent years Generative Adversarial Networks (GAN) [9] and Variational AutoEncoders (VAE) [17, 16] became the two state-of-the-art architectures for image generation or speech synthesis in the machine learning community [22, 24, 7, 10, 12]. Both of these models share similarities; they draw samples from a low-dimensional space and then process this low-dimensional sample through a neural network to generate content. Two reasons motivate a low-dimensional latent space. First, the generative process is much faster than directly generating from a high-dimensional correlated space. Second, we assume that the variability in the content can be explained with a small set of generative factors; everything else being random noise.

3.1 Variational AutoEncoders

Let us briefly introduce the architecture our proposed model is built on: VAE. Assuming we have $\mathbf{x}$, any type of content of size D, for instance an image and $\mathbf{z}$ the latent representation of the image of size $d << D$. An autoencoder learns an encoding function $q : \mathcal{X} \to \mathcal{Z}$ and a decoding function $p : \mathcal{Z} \to \mathcal{X}$ simultaneously. The VAE [17, 16] architecture assumes a prior distribution $p_\theta(\mathbf{z})$, a decoding distribution for $p_\theta(\mathbf{x}|\mathbf{z})$ and an encoding distribution $q_\varphi(\mathbf{z}|\mathbf{x})$ and optimizes the parameters of these distributions using maximum likelihood. To generate an image, we sample from the prior $p_\theta(\mathbf{z})$ and then process this sample through the decoding distribution $p_\theta(\mathbf{x}|\mathbf{z})$.

3.2 Controllable Variational AutoEncoders

Finally, let us introduce the proposed approach we are currently implementing in Python: Controllable Variational AutoEncoder (ConVAE). We propose a VAE built on two distinct sets of latent variables $\mathbf{z}$ and $\mathbf{c}$ instead of one. When generating content, we can adjust the control variables $\mathbf{c}$ to explicitly design something of our choice, sample from $\mathbf{z}$ to randomize details and finally generate the content $\mathbf{x}$ using $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})$.

To train the interpretable and desirable control variables $\mathbf{c}$ we will manually assign labels for a few observations and using the recent advances in semi-supervised learning [18, 19, 16, 15] the model will be able to learn from unlabelled data as well, thus using the large amount of data available for any type of content.

4 Conclusion

In this preliminary work, we attempt to fill the gap in the literature regarding the notion of control in generative models. We also propose a model that we believe offers great properties with respect to our notion of control. This model is being implemented and we expect results to come in shortly.

Acknowledgments

We would like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada and Ontario Graduate Scholarships.

Ethical implications

We are putting at the centre of our work the democratization of content creation. We understand the benefits of art creation for the human brain, and for the development of societies and we aim at creating tools so that everyone can express their creativity regardless of their technical skills. We also believe this will help independent video game designers, cartoon producers and such produce the content they envisioned thus improving the diversity of such content available.

References

[1] Alba Amato. *Procedural Content Generation in the Game Industry*, pages 15–25. Springer International Publishing, Cham, 2017.

[2] Ali Borji. Pros and cons of GAN evaluation measures. *CoRR*, abs/1802.03446, 2018.

[3] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv e-prints*, page arXiv:1804.03599, Apr 2018.

[4] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2615–2625. Curran Associates, Inc., 2018.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2172–2180. Curran Associates, Inc., 2016.

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006.

[7] Hui Ding, Kumar Sricharan, and Rama Chellappa. Exprgan: Facial expression editing with controllable expression intensity. *CoRR*, abs/1709.03842, 2017.

[8] K. Evtimova and A. Drozdov. Understanding Mutual Information and its Use in InfoGAN. 2016.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[10] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Arbitrary facial attribute editing: Only change what you want. *CoRR*, abs/1711.10678, 2017.

[11] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations. *arXiv e-prints*, page arXiv:1812.02230, December 2018.

[12] Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuan Cao, and Yuxuan Wang. Hierarchical generative modeling for controllable speech synthesis. 2019.

[13] Gao Huang, Yang Yuan, Qiantong Xu, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. 2018.

[14] IDV Inc. Speedtree, 2019.

[15] Joel Jaskari and Jyri J. Kivinen. A novel variational autoencoder with applications to generative modelling, classification, and ordinal regression. *ArXiv*, abs/1812.07352, 2018.

[16] D. P. Kingma. *Variational Inference & Deep Learning : A New Synthesis*. PhD thesis, Universiteit van Armsterdam, 10 2017.

[17] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[18] Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014.

[19] Abhishek Kumar, Prasanna Sattigeri, and P. Thomas Fletcher. Improved semi-supervised learning with gans using manifold invariances. *CoRR*, abs/1705.08850, 2017.

[20] Cortes C. LeCun, Y. and C.J.C Burges. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2019-09-03.

[21] Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling Disentanglement in Variational Auto-Encoders. *arXiv e-prints*, page arXiv:1812.02833, December 2018.

[22] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv e-prints*, page arXiv:1411.1784, Nov 2014.

[23] C. Nash and C. K. I. Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. *Comput. Graph. Forum*, 36(5):1–12, August 2017.

[24] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. *CoRR*, abs/1611.06355, 2016.

[25] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[26] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[27] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *Proceedings of ICLR 2016*, page arXiv:1511.01844, November 2016.

[28] Georgios N. Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. Springer, 2018.

Appendices

Mutual information

In information theory, the mutual information (MI) of two variables is a measure of mutual correlation between the variables. It quantifies the amount of information obtained about one variable when the other is observed. For two discrete random variables X and Y, the MI is defined as :

$$\text{MI}(X,Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p_{(X,Y)}(x,y)\log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right)$$

The MI is often presented as a difference of two entropy terms :

$$\text{MI}(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

where $H(X)$ is the entropy of X defined as the rate at which information on X is obtained in information theory. For X a discrete random variable taking k different values the entropy is defined as :

$$H(X) = -\sum_{i=1}^{k} p(x_i)\log(p(x_i))$$

VAE model

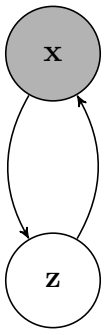

Figure 1: A graphical representation of the VAE architecture where $\mathbf{x}$ is the content (image) of size D and $\mathbf{z}$ its latent representation (code) of size $d << D$.

We assume the following model :

- A prior on $p_\theta(\mathbf{z})$ (Usually isotropic Normal)
- A decoding distribution $p_\theta(\mathbf{x}|\mathbf{z})$ ($\theta = NN_1(\mathbf{z})$)
- An encoding distribution $q_\varphi(\mathbf{z}|\mathbf{x})$ ($\varphi = NN_2(\mathbf{x})$) that serves as an approximation of the true but intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$

Evidence lower bound

To optimize the model parameters θ and φ, we use maximum likelihood. More precisely, we maximize the Evidence Lower BOunds (ELBO) $\mathcal{L}(\varphi, \theta) = \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}, \mathbf{z}) - \ln q_\varphi(\mathbf{z}|\mathbf{x})]$, which is a lower bound of the observed data log likelihood :

$$\begin{aligned}
\ln p(\mathbf{x}) &= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x})\right] \\
&= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p(\mathbf{x}, \mathbf{z}) q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x}) p(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] + \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathcal{L}(\varphi, \theta) + KL(q||p) \\
&\geq \mathcal{L}(\varphi, \theta).
\end{aligned} \tag{1}$$

ConVAE model

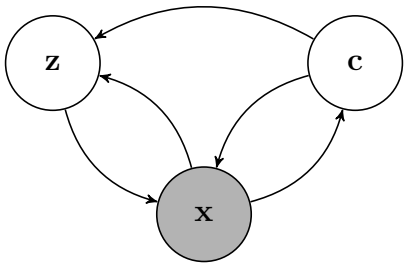

Figure 2: A graphical representation of the ConVAE architecture where $\mathbf{x}$ is the content (image) of size D, $\mathbf{c}$ the set of control variables and $\mathbf{z}$ the latent space of size $d << D$.

Figure 2 illustrates our proposed architecture. It induces the following factorization :

$$p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{c}) = p_\theta(\mathbf{c}) p_\theta(\mathbf{z}|\mathbf{c}) p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c}). \tag{2}$$

This graphical representation also induces the conditional independence of $\mathbf{z}$ and $\mathbf{c}$ given $\mathbf{x}$ which allows us to factorize the encoding distribution as well :

$$q_\varphi(\mathbf{z}, \mathbf{c}|\mathbf{x}) = q_\varphi(\mathbf{z}|\mathbf{x}) q_\varphi(\mathbf{c}|\mathbf{x}). \tag{3}$$

Example 1: Generating hand-written digits

The MNIST [20] data set frequently used in recent years is our first example. As ConVAE user, we could determine that we would like to control the digit c_1 and the style (cursive or straight) c_2. Those two categorical variables form our set of control variables $\mathbf{c}$. We have to manually assign the values of c_1 and c_2 for a few images before training the model.

The latent space $\mathbf{z}$ will take care of other aspects of the image, for example the digit thickness, its angle and its exact location. When we decide to generate an image of a digit, we pick the digit itself and its style by fixing c_1 and c_2 then randomly generate other aspects of the images by sampling from $p_\theta(\mathbf{z}|\mathbf{c})$ and finally generate the resulting image using $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})$.

Example 2: Generating an image of a sky

A more complex task would be to generate images of a sky. For example, we might want to control c_1 the types of clouds (cumulonimbus, altocumulus, etc) which is a categorical variable, the cloud density c_2 as a continuous variable where 0 represents a clear sky and 1 a cloudy sky and finally a day/night binary variable c_3 controlling the brightness of the sky. These three variables form the set of control variables $\mathbf{c}$. We have to manually assign the values of c_1, c_2 and c_3 for a few images before training the model.

The latent space $\mathbf{z}$ will take care of other aspects of the image, for example the actual location of the clouds and the exact size and number of clouds. When we decide to generate an image of a sky, we pick the cloud type, the cloud density and the time of the day by fixing c_1, c_2 and c_3 then randomly generates other aspects of the images by sampling from $p_\theta(\mathbf{z}|\mathbf{c})$ and finally generate the resulting image using $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})$.