What is shape? Characterizing particle morphology using genetic algorithms and deep generative models

Simon Guldager Andersen - Stud. ID: kpr279
(Dated: March 8, 2023)

This brief write-up is a summary of the scientific article "What is shape? Characterizing particle morphology using genetic algorithms and deep generative models" by R. Buarque de Macedo et al. published in Granular Matter 25, 2 (2023) [1]. My emphasis is chiefly on genetic algorithms and how the authors have used them to generate a variety of different particle shapes.

I. INTRODUCTION

Suppose we are in need of a granular material having certain physical properties. How should we design the grain shape in order to realize these properties? This is the fundamental question that the authors have tried to answer. To do this, they have started out by generating 10,000 images of unique particle shapes using a genetic algorithm. These images have then been used to train a variational autoencoder (VAE) to create an approximate basis space of possible grain shapes.

The bulk properties of a material composed of grains with a given shape can be simulated, and with an approximate forward mapping from the shape space to bulk properties at hand, the shape space can be searched for regions of optimal bulk properties as e.g. yield stress or packing fraction, albeit a lot of work is needed to thoroughly explore this mapping.

II. REVIEW

The main focus of this review is how they have used genetic algorithms to generate different particle shapes. In the following, we will sketch the principles of genetic algorithms followed by how the authors specifically implemented one. Finally, the principles of VAEs - and how they have been used to create a space of possible grain shapes - is covered very briefly.

A. Overview of genetic algorithms

A genetic algorithm (GA) is a stochastic global optimization scheme inspired by natural selection. It is very powerful for combinatorial optimization (e.g. the travelling salesman), but works well for continuous problems, too. It is generally quite efficient at finding a near-global optima, but might be inefficient at finding the exact one; by combining it with a local search strategy like e.g. a line search, its convergence rate and robustness can be improved considerably. Quite a lot of hyperparameters are involved, and careful tuning is necessary to achieve good performance. [2]

The basic implementation is as follows: A population of N solutions is initialized randomly or by using prior knowledge. Each iteration consists of a selection, crossover and mutation stage, which are repeated until the termination criterion is reached. These stages, along with their specific implementation in the article, will be illuminated in the following.

B. Using a GA for particle generation

In order to obtain a reasonable approximation to the space of possible grain shapes, the VAE should be trained on images of many different particle shapes. To generate such shapes, the authors have exploited that roundness, convexity and aspect ratio are known to have big influence on the shape of 2D particles, although they do not uniquely specify shape. They are defined as

$$R = \frac{\sum_{i=1}^{N} r_i}{r_{max}}, \quad C = \frac{a}{a_{hull}}, \quad A = \frac{\lambda_1}{\lambda_2}, \qquad (1)$$

where r_i is the radius of the circle than can 'fit into' the i'th corner (as estimated by the curvature of the corner point), and r_{max} is the radius of the biggest inscribed circle of the particle. Low roundness corresponds to sharp corners and vice versa. a is the particle area, while a_{hull} is the area of the convex envelope of the particle, as seen in fig. 3, and so the convexity is a measure of the degree to which a particle is perfectly convex. Finally, the aspect ratio is the ratio between the width and the length of the particle. R, C, A all take values in $[0, 1]$.

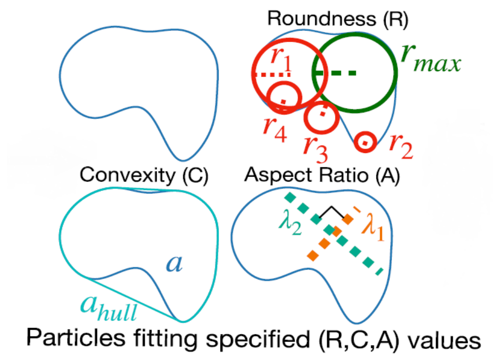

FIG. 1: Illustration (from [1]) of the quantities in (1)

The authors used a GA to generate particle shapes according to different (R, C, A) values, ultimately obtaining 10,000 unique images of shape. The cost function used to evaluate the quality of each solution has the form

$$cost = (C - C_{target})^2 + (R - R_{taget})^2 + (A - A_{target})^2$$
$$+ 100\mathrm{SC} + 100\mathrm{SI} + 100\mathrm{BN},$$

which is a least squares function forcing (R, C, A) as close to the specified values as possible, in addition to 3 terms penalizing unphysical grain shapes, namely too sharp corners (SC), self-intersection (SI), and bottlenecks (BN), i.e. when points on opposite side of the particle are too close. To generate a particle with $(R_{target}, C_{target}, A_{target})$, their implementation goes as follows:

Initialize 50 identical particles as 8 points uniformly distributed around the edge of an ellipse with a specified area and aspect ratio A_{target}

In each iteration:

Crossover: Generate k new solutions, each one by choosing a solution pair with probability $p_c = 0.2$, and randomly swapping points

Mutate each particle point (r, θ) with probability $p_m = 0.5$ by adding Gaussian noise to each of its components

Select the $\frac{N+k}{2}$ best solutions by choosing pairs at random, each time selecting the best solution. Duplicate the winners until the population size is 50

Terminate when the cost function of a solution is less than ϵ

C. Creating a space of possible shapes using VAEs

The GA was used to generate 10,000 unique images of particle shapes, some of which can be seen in fig. 3 in the appendix, and these are then used to train a VAE. Let us very briefly outline the principles of this deep generate model, while referring to the schematic of the architecture in fig. 2.

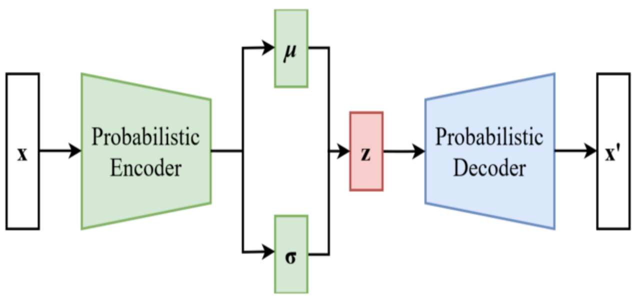

FIG. 2: Schematic of the architecture of a VAE. [4]

It is an unsupervised machine learning architecture used for dimensionality reduction. Through multiple convolutional layers, the encoder maps an input x to the d-dimenional vectors μ and σ, which typically has much fewer dimensions than the input. A 'latent vector', as it is called, z, is then calculated as $z \sim \mathcal{N}(\mu, \sigma^2)$, after which it is fed to the decoder part of the network, which consists of multiple convolutions, and whose job it is to reconstruct the input as closely as possible.

To construct the space of grain shapes, the idea is this: We assume that the 10,000 particle shape images are distributed fairly homogeneously throughout the space of possible grain shapes, in the sense that there are no important regions of shape space with no images.

These are then used to train the VAE, after which the encoder is able to generate two unique d-dimensional vectors μ, σ for a given particle shape, and the decoder can generate a unique particle shape given any latent vector z. In this sense, the latent space of z-values constitute a space of possible grain shapes.

Since each particle shape corresponds to a given Gaussian $\mathcal{N}(\mu, \sigma^2)$ (by construction), the distribution of possible grain shapes consists of many Gaussians, whose relative distance is determined by their grain shape similarity.

The authors found that $d = 20$ dimensions was enough to accurately reconstruct the input, i.e. that 20 parameters are needed to uniquely specify 2D particle shape, and that for each dimension z_i, only $z_i \in [-4, 4]$ corresponds to connected and therefore physically acceptable shapes. Using just 10 subdivisions per dimension still gives rise to 10^{20} unique particle shapes.

III. CONCLUSION

In this write-up, we have reviewed the method by which the authors have constructed a space of possible grain shapes using genetic algorithms to generate traning data for a VAE. After training, the decoder can generate a unique particle shape given any 20 dimensional latent vector z, and the 20 dimensional latent space of possible z's constitutes an approximate basis for all (physically acceptable) grain shapes.

This shape space can then be searched for regions of optimal bulk properties. Since the space of possible shapes is vast and the simulation of bulk properties computationally expensive, it is not an easy task to search the entire shape space, but one could start searching smaller subspaces and hope to get lucky. It would be very useful to obtain an understanding of the 20 latent space dimensions in terms of how they each affect shape, although it is by no means given that each dimension has a simple physical interpretation.

The prospects of having constructed a space of possible shapes are exciting: Suppose for instance that we feed a large number of images of different biological cell types into the VAE. If different cell types happen to aggregate in different regions of the shape space, a classification scheme connecting cell shape and functionality/type could be made, potentially making it possible to identify e.g. cancer cells just from their shape.

IV. APPENDIX

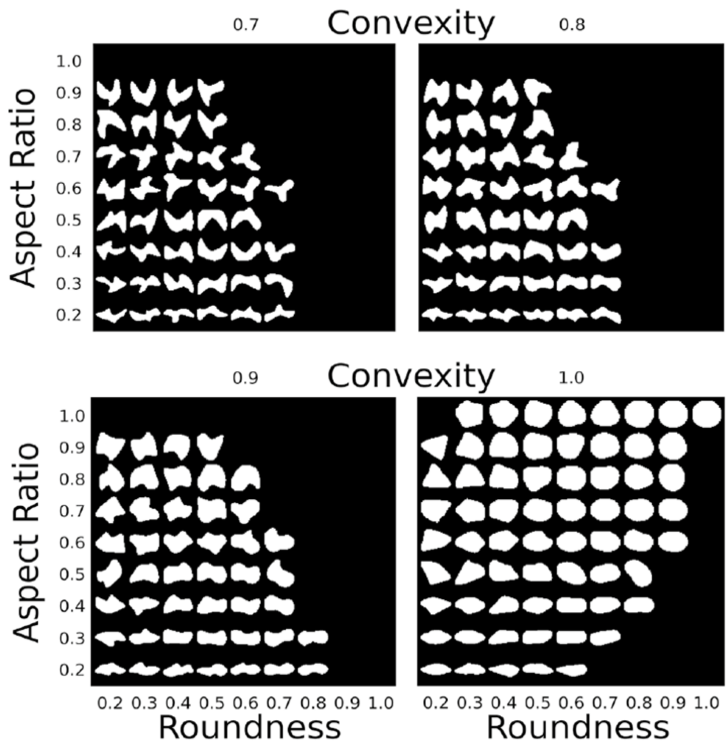

FIG. 3: Figure taken from [1]. Examples of grain shapes with specified (R, C, A) values generated by a genetic algorithm. Each box corresponds to a given convexity, with roundness on the horizontal axis and aspect ratio on the vertical axis. If a shape is missing for a given (R, C, A), they genetic algorithm did not converge.

V. REFERENCES

[1] Macedo, R.B.d., Monfared, S., Karapiperis, K. et al. *What is shape? Characterizing particle morphology with genetic algorithms and deep generative models*, Granular Matter 25, 2 (2023). https://doi.org/10.1007/s10035-022-01282-y

[2] Yang, Xin-She (2010) *Nature-Inspired Metaheuristic Algorithms*, 2nd edition, Luniver Press, p 1-10, 41-46

[3] Doersch, C.(2016) *Tutorial on variational autoencoders*
arXiv preprint: `https://arxiv.org/abs/1606.05908`

[4] Image retrieved from
`https://en.wikipedia.org/wiki/Variational_autoencoder#/media/File:Reparameterized_Variational_Autoencoder.png` on March 7, 2023