Shape optimization in laminar flow

Stephan Eismann
Stanford University
seismann@cs.stanford.edu

Stefan Bartzsch
TU Munich
stefan.bartzsch@tum.de

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Abstract

Computational design optimization in fluid dynamics usually requires to solve non-linear partial differential equations numerically. In this work, we explore a Bayesian optimization approach to minimize an object's drag coefficient in laminar flow based on predicting drag directly from the object shape. Jointly training an architecture combining a variational autoencoder mapping shapes to latent representations and Gaussian process regression allows us to generate improved shapes in the two dimensional case we consider.

1 Introduction

The optimization of hydro- and aerodynamic properties for vehicles, turbines and engines is a common and challenging engineering problem. The design process usually involves both numeric simulations as well as experimental measurements. Due to the large complexity of the design space success often relies on experience and skill of the engineer [1].

In this work we explore whether Bayesian optimization can facilitate the design process. As an exemplary and general problem we discuss the resistance of an object in a fluid (e.g. air or water). What shape should a vesicle have to reduce energy dissipation by drag to a minimum? The resistance of an object in fluid dynamics is characterized by the drag coefficient c_d [2],

$$c_d = \frac{2F_d}{\rho v^2 A}. \tag{1}$$

In the reference frame of the object, F_d is the component of the force caused by the fluid in flow direction which represents drag. ρ is the density of the fluid, v the flow velocity and A the frontal area of the object in flow direction. c_d is not a constant but depends on velocity, viscosity and other parameters which are summarized in the Reynolds number Re, a dimensionless quantity. Together c_d and Re which are mostly empirically determined govern the drag of an object.

The Navier-Stokes equations provide a theoretical description of the flow of a fluid around an object [2],

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} + \rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\nabla p + \eta \nabla^2 \boldsymbol{v}, \tag{2}$$

where p denotes the pressure and η the viscosity coefficient. Equation 2 describes a second order non-linear partial differential equation and is except for a few special cases only numerically solvable. Here we consider the laminar (i.e. non-turbulent) flow of an incompressible fluid that obeys

$$\nabla \cdot \boldsymbol{v} = 0, \tag{3}$$

in the limit of low Reynolds numbers Re. The drag forces on the object have two components: (i) friction drag which is caused by the viscosity of the fluid and (ii) pressure drag which is caused by the pressure distribution around the object's surface.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Finally, it should be noted that Equation 1 strictly speaking only holds true for a limited range of Re. In the domain of laminar flow at low Re, the drag of an object is proportional to the velocity v (Stokes law) [2]. Drag only becomes proportional to the square of velocity when turbulences occur in the wake field of the object at larger velocities. Nevertheless, c_d provides a quantitative account of the drag in laminar flow if v is kept constant throughout all simulations.

Previous works have demonstrated the successful application of kriging or Gaussian process (GP) regression for the optimization of parametric designs in aerodynamic applications [3–6]. Here, we present a purely data-based approach. The combination of variational autoencoder (VAE) and Gaussian process allows us to learn and optimize shape-characteristics in a non-parametric form directly from images.

2 Data and model

2.1 Laminar flow in 2D

We consider the case of laminar flow around an object in two dimensions as it allows us to generate training and test data at relatively low computational cost.

Figure 1 illustrates the simulation setup and computation. We generate random shapes by picking radii from a uniform distribution at different polar angles. Subsequently we use a set of Fourier descriptors to obtain a smooth shape which facilitates the geometry meshing (Fig.1A). The discretized space allows us to solve the Navier-Stokes equations (Eq.2) with a standard finite element solver (QuickerSim, MATLAB). The left wall is defined as fluid inlet with constant flow profile directed in positive x-direction. Furthermore we apply slip boundary conditions to the upper and lower wall (i.e. $v_y = 0$) and consider slip at the object boundaries. Figures 1B,C show the resulting fluid velocity fields for a range of shapes. In analogy to the three dimensional case we define the frontal area A as the maximum object extension in y-direction as projected onto the x-axis. All simulations are performed with a scalar viscosity $\nu = 0.02$. Generating a dataset of 5000 shapes requires about 16h on a single machine (Intel i7-3520M with 2.90 GHz, 16GB memory).

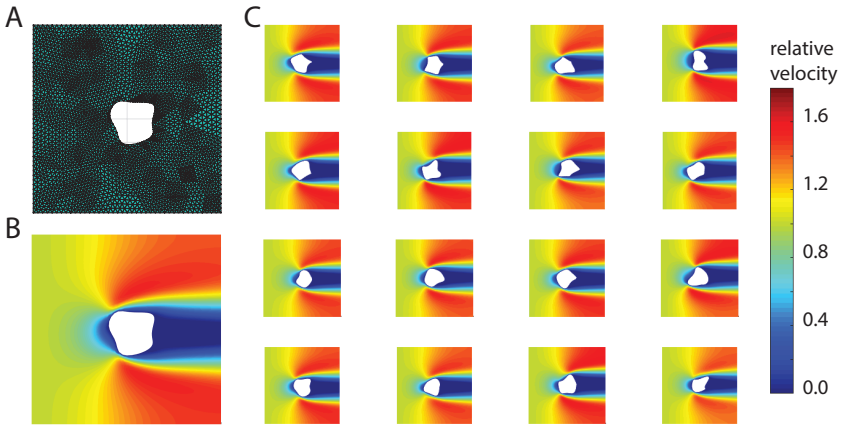

Figure 1: **Generation of training data** (**A**) Meshed geometry around a randomly generated shape. (**B**) Color-coded fluid velocity around the shape calculated as a solution to the Navier-Stokes equations. The left wall defines an inlet for a flow in positive x-direction. Red and green indicate regions of high and low flow velocity, respectively. (**C**) Color-coded velocity fields for a range of randomly generated shapes.

2.2 Guided shape encoding and Bayesian optimization

Figure 2A shows a schematic of the model architecture. The architecture is similar to the approach described in Gómez-Bombarelli et al. [7] for the design of drug-like molecules and light-emitting diodes.

Object shapes are encoded as binary images of size 112×84 and mapped to and from a continuous latent space $\mathbf{z}$ of dimensionality d by two neural networks forming the encoder and decoder pair of a variational autoencoder (VAE) [8]. The encoder consists of two convolutional and two dense layers. The decoder is designed similarly with two dense followed by two deconvolutional layers.

We map the latent feature vectors $\mathbf{z}$ to drag coefficient y through a sparse-spectrum Gaussian process (GP) regression model [9]. We choose the sparse approach to avoid the computational complexity of a full GP. Next to the most probable drag coefficient, the Gaussian process provides us also with an uncertainty estimate when predicting the drag of unobserved points in the latent space. The probabilistic measure allows us to calculate expected improvement (EI) [10] for each new point with respect to the smallest drag coefficient observed so far and optimize towards larger EI values using gradient ascent. Finally, we decode latent feature vectors associated with large EI values to obtain corresponding object shapes.

In the approach of Gómez-Bombarelli et al. [7] VAE and GP are trained separately and sequentially. Here we introduce an additional neural network termed 'drag network' (DN) mapping from latent space $\mathbf{z}$ to drag coefficient y and whose parametric form allows us to enforce label-guided encoding in a straightforward manner. We define the joint loss function $\mathrm{loss}_{\mathrm{JOINT}}$ as the sum of VAE loss (reconstruction plus KL divergence) and the mean-squared regression error of the DN:

$$\mathrm{loss}_{\mathrm{JOINT}} = \mathrm{loss}_{\mathrm{VAE}} + \mathrm{loss}_{\mathrm{DN}} \quad . \tag{4}$$

The $\mathrm{loss}_{\mathrm{DN}}$ term encourages the encoder to find a representation that is amenable for regression. However, DN does not provide a measure of uncertainty for its predictions, hence it is not suitable for shape optimization. We therefore additionally train the GP regression model on the latent space, keeping the encoder fixed.

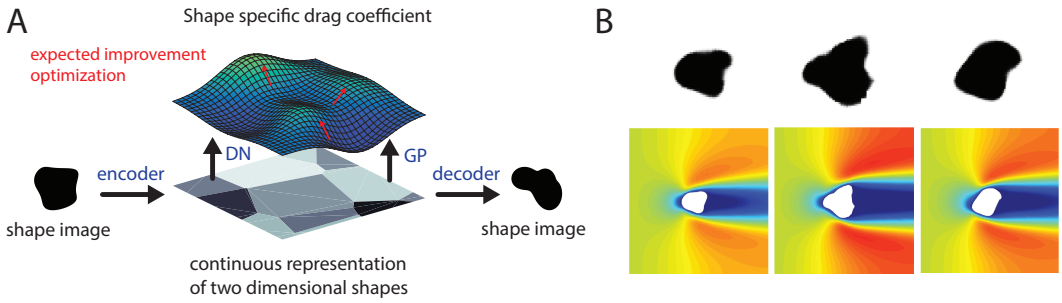

Figure 2: **Model architecture and drag optimized shapes** (**A**) Images of object shapes are mapped to and from a continuous latent space by an encoder-decoder pair of neural networks. An additional neural network (DN) maps the latent space feature vectors to drag coefficient. Combining the networks' loss functions yields a simple way to label-guided encoding. To obtain an uncertainty measure for the predictions we also train a sparse Gaussian process (GP) and generate new shapes by optimizing expected improvement (EI) through gradient ascent. (**B**) Three shapes corresponding to local EI maxima and the corresponding velocity-field solutions. The generated shapes are slightly fuzzy as is characteristic for VAEs. All shapes improve upon the best example in the training set with respect to drag coefficient.

3 Results

3.1 Comparing joint and separate training

We consider a dataset of 5200 shapes with corresponding drag coefficients and randomly assign 80% and 20% of the data to training and test set, respectively. Table 1 compares two different training modalities for the models: (i) VAE and drag network are trained jointly (Equation 4), (ii) VAE and drag network are trained independently in a sequential manner. All models were trained using the same set of hyperparameters except for latent dimensionality and the number of units per dense layer which is doubled for columns $(2, 4, 6, 8)$ compared to $(1, 3, 5, 7)$.

On average the separately trained variational autoencoders perform better in reconstructing shapes from the test and training set as expected. Reconstruction loss here is measured as the mean squared

deviation per pixel. The same holds true for the predictive performance of the drag network on the training set. However, the situation is reversed in the test situation in which the jointly trained models do much better both in terms of Gaussian process and drag network prediction.

	Joint training				**Separate training**			
Latent dimensions	10	10	20	20	10	10	20	20
GP: MSE test	0.64	0.77	0.64	**0.52**	0.76	0.76	1.00	0.74
DN: MSE test	**0.54**	0.61	0.56	**0.54**	0.92	0.87	1.00	0.87
DN: MSE train	1.00	0.72	0.94	0.91	0.54	**0.48**	0.58	0.47
Reconstr. test	0.60	1.00	0.40	0.38	0.50	0.46	0.95	**0.28**
Reconstr. train	0.67	0.80	0.43	0.35	0.56	0.51	1.00	**0.30**

Table 1: **Guided encoding enhances predictive performance for test shapes**
The table compares test and training errors for models of varying latent dimensionality and joint and separate training of VAE and DN. Errors are relative to the largest error among all models. The number of units per dense layer is doubled for experiments in columns $[2, 4, 6, 8]$ compared to $[1, 3, 5, 7]$. For these models the DN performed better on the test set than the sparse spectral GP which we attribute to the relative small number of inducing points (500) and a lack of their optimization with respect to marginal likelihood. Best results are shown in bold.

3.2 Optimizing for drag-reduced shapes

After training an additional Gaussian process model (1000 inducing spectral points, 20 latent dimensions, joint training) on the dataset mentioned above we generate new object shapes by optimizing expected improvement with respect to the smallest drag coefficient in our training data. To do so we randomly select 2000 starting points from the latent distribution of the training set and perform parallel gradient ascent. The latent points corresponding to the largest local maxima are then decoded to new object shapes. Overall training requires about 7h (TensorFlow implementation on a NVIDIA TITAN Xp) with the expected improvement optimization performed locally (Intel i7-3520M with 2.90 GHz, 16GB memory) requiring an additional 1h.

Fig. 2B shows the shapes and computed velocity fields for the three best out of a total of 25 decoded latent points after one round of Bayesian optimization. The decoded shapes are slightly blurry as is characteristic for VAE-generated images. We perform Sobel edge detection to extract a set of boundary points for the shapes and can subsequently calculate the corresponding drag coefficient. All three shapes slightly outperform the best training data point with improvements in the range of $4 - 8\%$.

4 Discussion

Computational design in fluid dynamics is costly due to the requirement of numerically solving non-linear differential equations. Here we present a data-driven Bayesian approach to design optimization. We are able to generate object shapes with an improved drag coefficient compared to the training dataset in the case of two dimensional, laminar flow. In addition, we demonstrate the benefit of label-guided encoding for shape-based drag prediction. Natural extensions of this work include the application of the presented approach to the case of three dimensions and turbulent flow.

References

[1] A. Kumar and J. N. Hefner. Future challenges and opportunities in aerodynamics. *The Aeronautical Journal*, 104(1038):365–373, August 2000. ISSN 0001-9240, 2059-6464. doi: 10.1017/S0001924000064009.

[2] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*. Elsevier, October 2013. ISBN 9781483140506.

[3] Timothy W. Simpson, Timothy M. Mauery, John J. Korte, and Farrokh Mistree. Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization. *AIAA Journal*, 39(12): 2233–2241, 2001. ISSN 0001-1452. doi: 10.2514/2.1234.

[4] Jay D. Martin and Timothy W. Simpson. Use of Kriging Models to Approximate Deterministic Computer Models. *AIAA Journal*, 43(4):853–863, 2005. ISSN 0001-1452. doi: 10.2514/1.8650.

[5] Shinkyu Jeong, Mitsuhiro Murayama, and Kazuomi Yamamoto. Efficient Optimization Design Method Using Kriging Model. *Journal of Aircraft*, 42(2):413–420, 2005. ISSN 0021-8669. doi: 10.2514/1.6386.

[6] J. C. Jouhaud, P. Sagaut, M. Montagnac, and J. Laurenceau. A surrogate-model based multidisciplinary shape optimization method with application to a 2d subsonic airfoil. *Computers & Fluids*, 36(3):520–529, March 2007. ISSN 0045-7930. doi: 10.1016/j.compfluid.2006.04.001.

[7] Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *arXiv:1610.02415 [physics]*, October 2016. arXiv: 1610.02415.

[8] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013. arXiv: 1312.6114.

[9] Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse Spectrum Gaussian Process Regression. *J. Mach. Learn. Res.*, 11:1865–1881, August 2010. ISSN 1532-4435.

[10] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *J. of Global Optimization*, 13(4):455–492, December 1998. ISSN 0925-5001. doi: 10.1023/A:1008306431147.