CS-GY-9223

Synthetic Data

Synthetic Data

- There's so much real data out there, why should we care about synthetic data?
- What are the uses of synthetic data?
- In this class, we will do an initial exploration of work in this area.

Synthetic Generation of High-Dimensional Datasets

Georgia Albuquerque, Thomas Löwe, and Marcus Magnor, *Member, IEEE*

Abstract—Generation of synthetic datasets is a common practice in many research areas. Such data is often generated to meet specific needs or certain conditions that may not be easily found in the original, real data. The nature of the data varies according to the application area and includes text, graphs, social or weather data, among many others. The common process to create such synthetic datasets is to implement small scripts or programs, restricted to small problems or to a specific application. In this paper we propose a framework designed to generate high dimensional datasets. Users can interactively create and navigate through multi dimensional datasets using a suitable graphical user-interface. The data creation is driven by statistical distributions based on a few user-defined parameters. First, a grounding dataset is created according to given inputs, and then structures and trends are included in selected dimensions and orthogonal projection planes. Furthermore, our framework supports the creation of complex non-orthogonal trends and classified datasets. It can successfully be used to create synthetic datasets simulating important trends as multidimensional clusters, correlations and outliers.

Index Terms—Synthetic data generation, multivariate data, high-dimensional data, interaction.

1 Introduction

Generating synthetic datasets is a common practice in many research areas. In situations where real data may be difficult to obtain, due to budget, time or privacy concerns, generating synthetic data poses a viable alternative. It is commonly employed as a substitute for real data and can be used to provide a controlled testing environment that meets specific conditions. This can be especially useful for purposes of verification, simulation or proof of concept.

There are numerous algorithms and tools that can be used to create synthetic data. However, most of the work done so far was developed for specific applications. For example, synthetic data generation to test Information Discovery and Analysis Systems [9, 8] or software testing [10].

The common solution to generate synthetic data is to use well-known statistical tools or develop small problem-oriented applications, which can be a time-consuming task. Addressing this shortcoming, we propose a framework for synthetic data generation that allows creating multivariate datasets based on visual specifications. Our approach was originally developed to create test cases for automated visual exploration algorithms [20, 17], but it can be directly used for a variety of applications that need multivariate datasets, such as learning and data mining algorithms. Such visual exploration algorithms are commonly used to search for hidden structures in high-dimensional datasets that can be observed in lower-dimensional projections of the data (two or three dimensions). A typical synthetically generated dataset to test these algorithms has a large number of dimensions. The values in most dimensions are randomly generated and multidimensional structures are then defined in the remaining dimensions. Some examples of such structures are correlations or clusters between selected dimensions. A good dataset does not necessarily exhibit only features that can be found by any visual analysis tool, but it may contain complex structures that are difficult to find using most algorithms, e.g., non-orthogonal structures or non-linear correlation between dimensions. Using synthetic datasets, it can be easily verified if the algorithm succeeded in finding all defined structures.

- *Georgia Albuquerque is with the Computer Graphics Lab, TU Braunschweig, Germany, E-mail: georgia@cg.cs.tu-bs.de.*
- *Thomas Löwe is with the Computer Graphics Lab, TU Braunschweig, Germany, E-mail: loewe@cg.cs.tu-bs.de.*
- *Marcus Magnor is with the Computer Graphics Lab, TU Braunschweig, Germany, E-mail: magnor@cg.cs.tu-bs.de.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

We propose a powerful approach to generate different patterns in the data, including multidimensional clusters, correlations and other trends. The points of the dataset are created by sampling statistical distributions in a multidimensional space: First, the user gives basic information about the desired dataset, e.g. number of dimensions and samples. Second, the user defines a set of structures in the multidimensional space that are represented by *probability density functions*. Finally, all points are generated by sampling these functions. Our framework allows users to create classified and unclassified high-dimensional datasets and to insert complex structures present in real data using painting tools, e.g. correlations between the attributes and clusters with different forms and dimensions. Additionally, the algorithm is able to create structures based on existing real data.

The rest of the paper is structured as follows: Section 2 discusses related work, Section 3 presents a short overview of our framework, Sections 4 and 5 describe our generation algorithm. Sections 6 and 7 show how the user may interact with the framework and some dataset examples, respectively. Finally, Section 8 concludes the paper, outlining open issues and future work.

2 Related Work

Generating synthetic data is an important tool that is used in a variety of areas, including software testing, machine learning, and privacy protection.

In software testing, synthetically generated inputs can be used to test complex program features and to find system faults. The difficult and expensive process of generating test data that meets specific fault-based testing criteria has traditionally been done by hand. In 1991, DeMillio and Offut claim to have made the first attempt to automatically generate such test data [5], introducing the approximation technique of *constraint-based testing*. More recent research suggests the use of genetic algorithms to generate adequate test data [10, 12]. To this day, generating test data remains an active research topic in this field.

Another application area is the field of machine learning, where synthetic data can be used to balance training data. For example, when teaching an artificial intelligence to distinguish between a number of different patterns, the amount of samples for each pattern should be as equal as possible. In this field, it is a common problem that over-representing a certain pattern may introduce artificial bias towards that pattern. In order to prevent this problem, training datasets may be equalized by generating additional synthetic training data [7].

Synthetic data is also commonly used in scenarios where collecting real data is not an option, due to budget, time or privacy concerns. These datasets are designed to obscure sensitive information, while still maintaining statistical properties of the original data [14]. This may be the case when testing the reliability of data-mining tools that have to operate on complex multivariate data. For the simulation

G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.

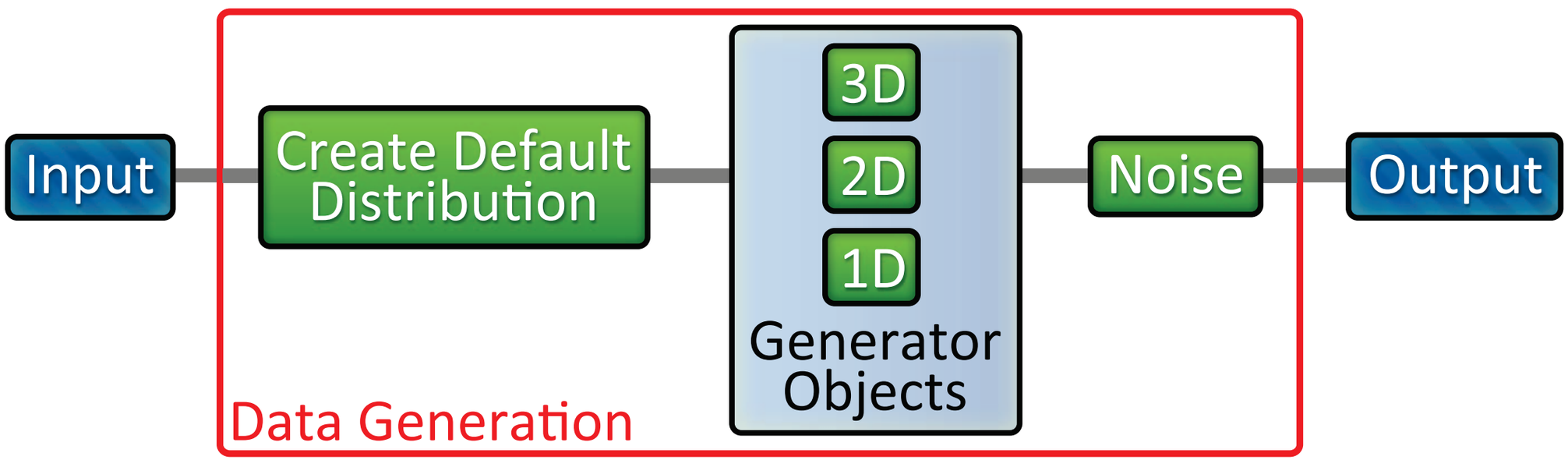
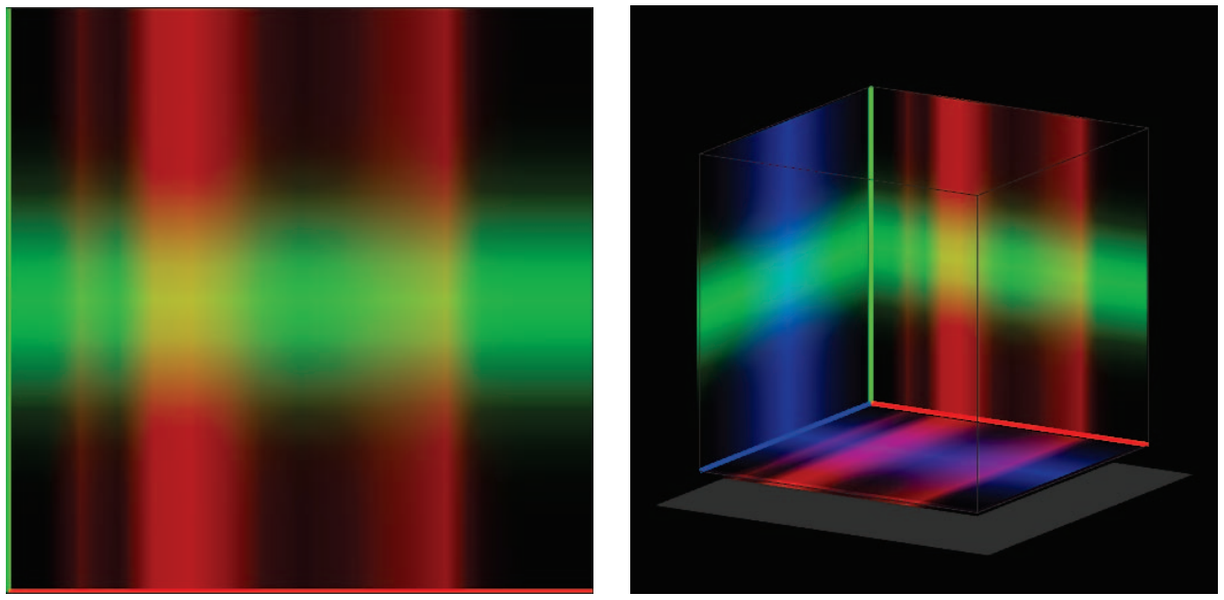


Fig. 1. Framework to generate high-dimensional datasets

G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.

One-Dimensional Probability Density Functions

$$\rho(x_1,\ldots,x_n) = \prod_{i=1}^{n} \rho_i(x_i).$$

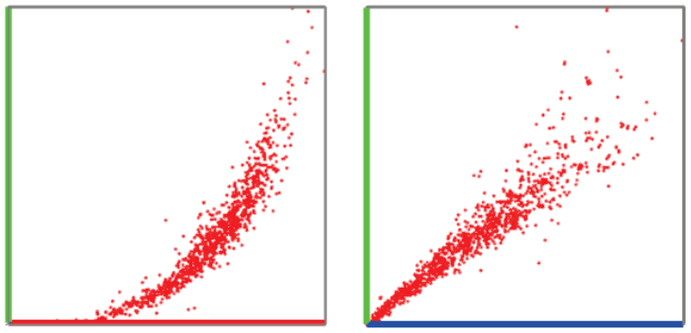


(a) $n = 2$ (b) $n = 3$

Fig. 2. Generator schematics, showing one-dimensional PDFs ρ_i (marked red, green and blue) along the axes of the probability density space in (a) two and (b) three dimensions. This view illustrates how a set of one-dimensional functions span a multi-dimensional PDF, with colour intensities representing the respective probability distributions.

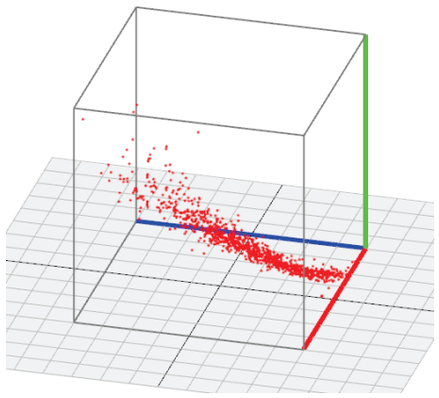
G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.

Two-Dimensional Probability Density Functions



(a) Axis-aligned projection (green and red dimensions)

(b) Axis-aligned projection (green and blue dimensions)



(c) Axis-independent projection

Fig. 3. A dataset consisting of 1000 samples. It was generated using two different two-dimensional PDFs defined between (a) the green and red, and (b) the green and blue dimensions, respectively. (c) is an axis-independent view that shows the three-dimensional structure that has been created by combining the two PDFs.

Probability Distribution Planes

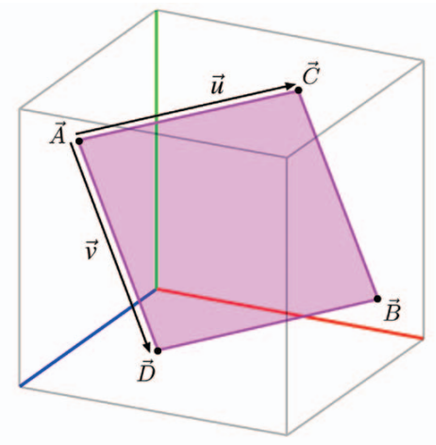
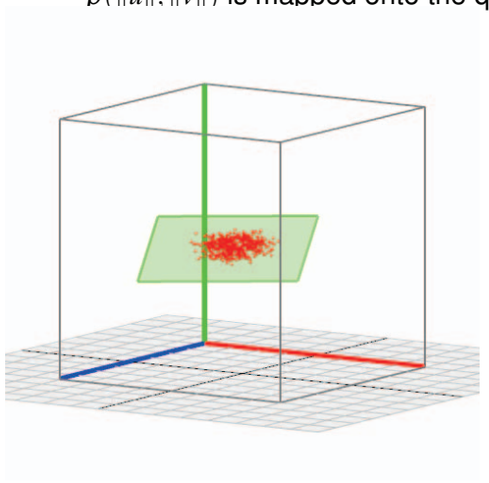
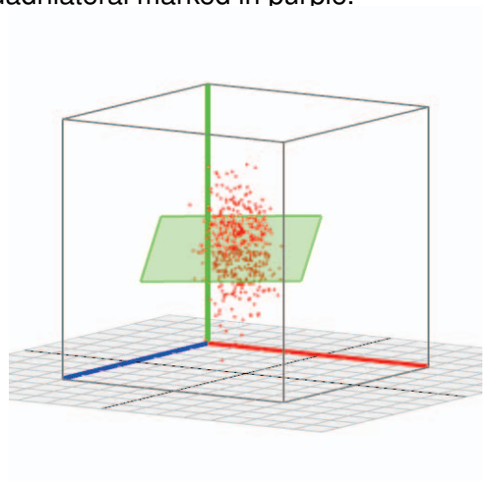


Fig. 4. Probability Distribution Plane (PDP). $\vec{A}$, $\vec{B}$, $\vec{C}$, and $\vec{D}$ define the corner points of the PDP generator object. A two-dimensional PDF $\rho(\|\vec{u}\|, \|\vec{v}\|)$ is mapped onto the quadrilateral marked in purple.



(a) No scattering



(b) Gaussian scattering

Fig. 5. A dataset of 500 points generated using a PDP

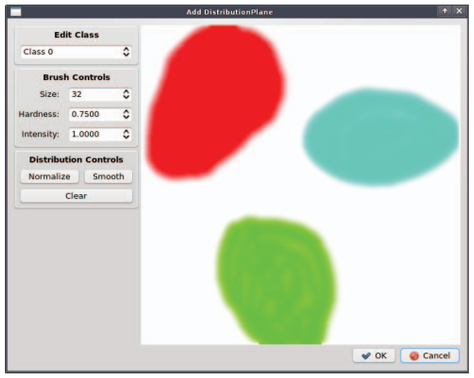
G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.

5 Sampling a Probability Density Function

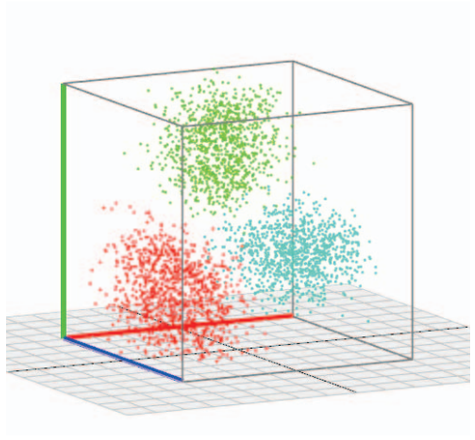
Using our framework, the user may define an arbitrary one- or two-dimensional *probability density function* as input by simply drawing it on the screen. This drawn input is then interpreted as a finite list of values representing a PDF. Directly interpolating these values into a continuous function and using common statistical methods to sample this distribution may be very computationally expensive depending on the given function. For that reason, we developed an alternative algorithm that allows us to quickly sample any function the user may give as input.

SEE PAPER FOR DETAILS

G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.



(a) Dimensions 3 and 5



(b) Dimensions 3, 5 and 2

Fig. 12. Example of a cluster pattern created with our painting tool. (a) Sketching the class clusters. (b) 3-dimensional projection including dimensions 3, 5 and 2.

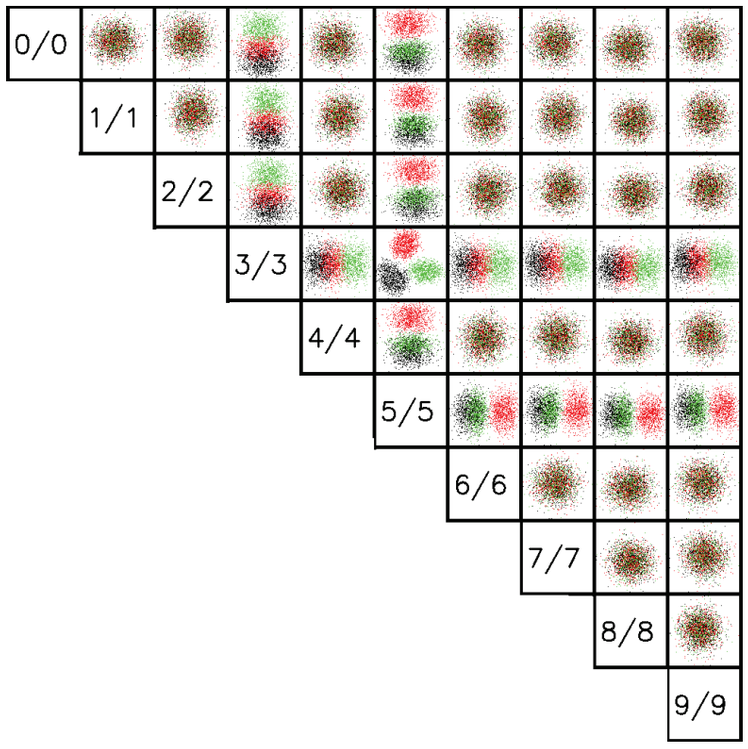
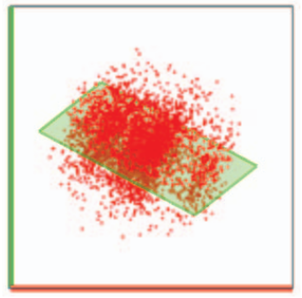
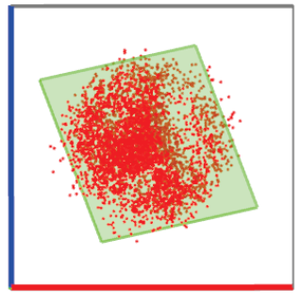


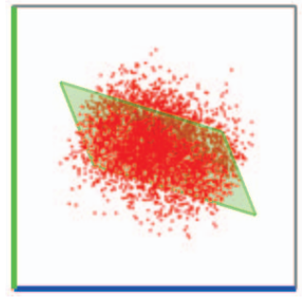
Fig. 13. Generated dataset with 10-dimensions, 1000 samples for each of the three defined classes. Hidden cluster patterns were modeled in two dimensions: 3 and 5. The remaining dimensions were defined by randomly sampling a normal distribution.



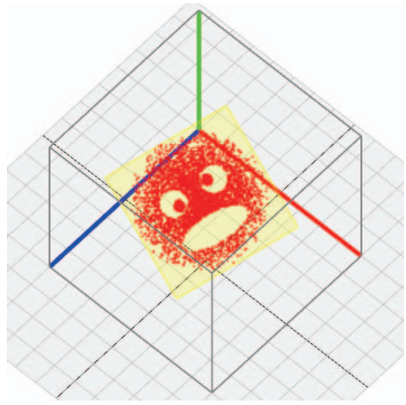
(a) Dimensions 3 and 8



(b) Dimensions 3 and 9



(c) Dimensions 9 and 8



(d) Hidden structure

Fig. 16. Dataset with 10 dimensions with non-orthogonal structures between dimensions 3, 8 and 9 (d). (a), (b) and (c) show axis-aligned, orthographic projections of the three participating dimensions (3, 8 and 9) to visualize the dataset. (d) shows the hidden structured viewed from a non-orthogonal perspective.

G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.

DataSynthesizer: Privacy-Preserving Synthetic Datasets

Haoyue Ping
Drexel University, USA
hp354@drexel.edu

Julia Stoyanovich*
Drexel University, USA
stoyanovich@drexel.edu

Bill Howe†
University of Washington, USA
billhowe@cs.washington.edu

ABSTRACT

To facilitate collaboration over sensitive data, we present DataSynthesizer, a tool that takes a sensitive dataset as input and generates a structurally and statistically similar synthetic dataset with strong privacy guarantees. The data owners need not release their data, while potential collaborators can begin developing models and methods with some confidence that their results will work similarly on the real dataset. The distinguishing feature of DataSynthesizer is its usability — the data owner does not have to specify any parameters to start generating and sharing data safely and effectively.

DataSynthesizer consists of three high-level modules — DataDescriber, DataGenerator and ModelInspector. The first, DataDescriber, investigates the data types, correlations and distributions of the attributes in the private dataset, and produces a data summary, adding noise to the distributions to preserve privacy. DataGenerator samples from the summary computed by DataDescriber and outputs synthetic data. ModelInspector shows an intuitive description of the data summary that was computed by DataDescriber, allowing the data owner to evaluate the accuracy of the summarization process and adjust any parameters, if desired.

We describe DataSynthesizer and illustrate its use in an urban science context, where sharing sensitive, legally encumbered data between agencies and with outside collaborators is reported as the primary obstacle to data-driven governance.

The code implementing all parts of this work is publicly available at https://github.com/DataResponsibly/DataSynthesizer.

CCS CONCEPTS

• **Security and privacy → Data anonymization and sanitization**; **Privacy protections**; **Usability in security and privacy**;

KEYWORDS

Data Sharing; Synthetic Data; Differential Privacy

*This work was supported in part by NSF Grants No. 1464327 and 1539856, and BSF Grant No. 2014391.
†This work was supported by the University of Washington Information School, Microsoft, the Gordon and Betty Moore Foundation (Award #2013-10-29) and the Alfred P. Sloan Foundation (Award #3835) through the Data Science Environments program.

SSDBM '17, Chicago, IL, USA

DOI: http://dx.doi.org/10.1145/3085504.3091117

ACM Reference format:
Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *Proceedings of SSDBM '17, Chicago, IL, USA, June 27-29, 2017,* 5 pages.
DOI: http://dx.doi.org/10.1145/3085504.3091117

1 INTRODUCTION

Collaborative projects in the social and health sciences increasingly require sharing sensitive, privacy-encumbered data. Social scientists, government agencies, health workers, and non-profits are eager to collaborate with data scientists, but formal data sharing agreements are too slow and expensive to create in ad hoc situations — our colleagues report that 18 months is a typical timeframe to establish such agreements! As a result, many promising collaborations can fail before they even begin. Data scientists require access to the data before they can understand the problem or even determine whether they can help. But data owners cannot share data without significant legal protections in place. Beyond legal concerns, there is a general reluctance to share sensitive data with non-experts before they have "proven themselves," since they do not understand the context in which the data was collected and may be distracted by spurious results.

To bootstrap these collaborations without incurring the cost of formal data sharing agreements, we saw a need to generate datasets that are *structurally and statistically similar* to the real data but that are 1) obviously synthetic to put the data owners at ease, and 2) offer strong privacy guarantees to prevent adversaries from extracting any sensitive information. These two requirements are not redundant: strong privacy guarantees are not always sufficient to convince data owners to release data, and even seemingly random datasets may not prevent subtle privacy attacks. With this approach, data scientists can begin to develop models and methods with synthetic data, but maintain some degree of confidence that their work will remain relevant when applied to the real data once proper data sharing agreements are in place.

We propose a tool named DataSynthesizer to address this problem. Assume that the private dataset contains one table with m attributes and n tuples, and that the values in each attribute are homogeneous, that is, they are all of the same data type. We are interested in producing a synthetic dataset such that summary statistics of all numerical, categorical, string, and datetime attributes are similar to the private dataset. What statistics we preserve depends on the data type, as we will discuss in Section 3.

DataSynthesizer infers the domain of each attribute and derives a description of the distribution of attribute values in the private dataset. This information is saved in a dataset description file, to which we refer as data summary. Then DataSynthesizer is able to generate synthetic datasets of arbitrary size by sampling from the probabilistic model in the dataset description file.

Haoyue Ping, Julia Stoyanovich, Bill Howe: DataSynthesizer: Privacy-Preserving Synthetic Datasets. SSDBM 2017: 42:1-42:5

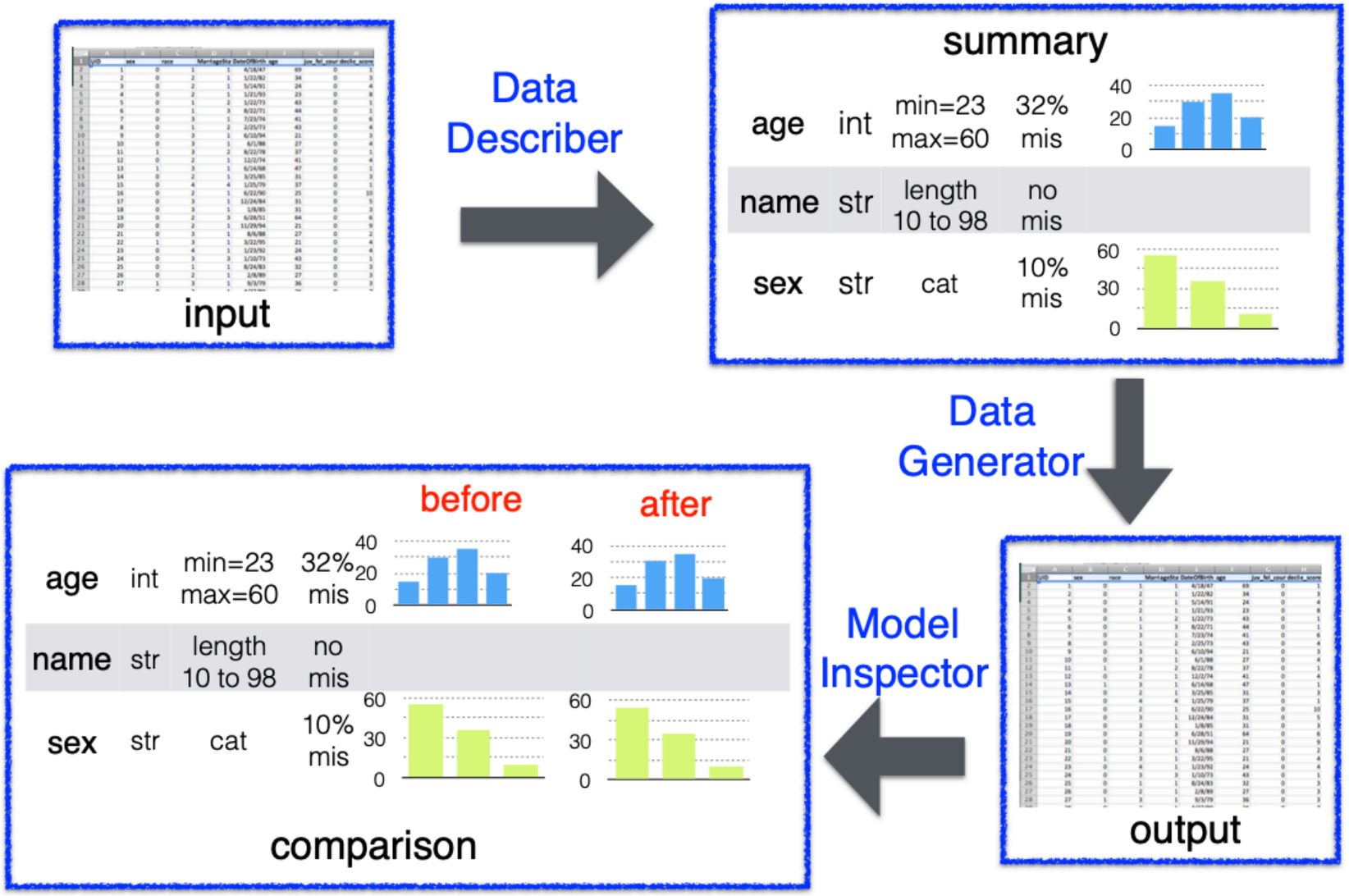


Figure 1: The DataSynthesizer system architecture.

Haoyue Ping, Julia Stoyanovich, Bill Howe: DataSynthesizer: Privacy-Preserving Synthetic Datasets. SSDBM 2017: 42:1-42:5

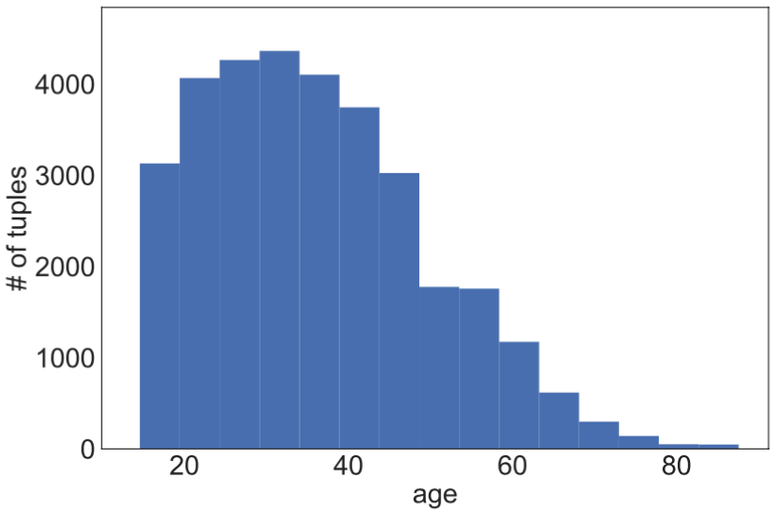


Figure 2: Histogram on *age*: Adult Income [6].

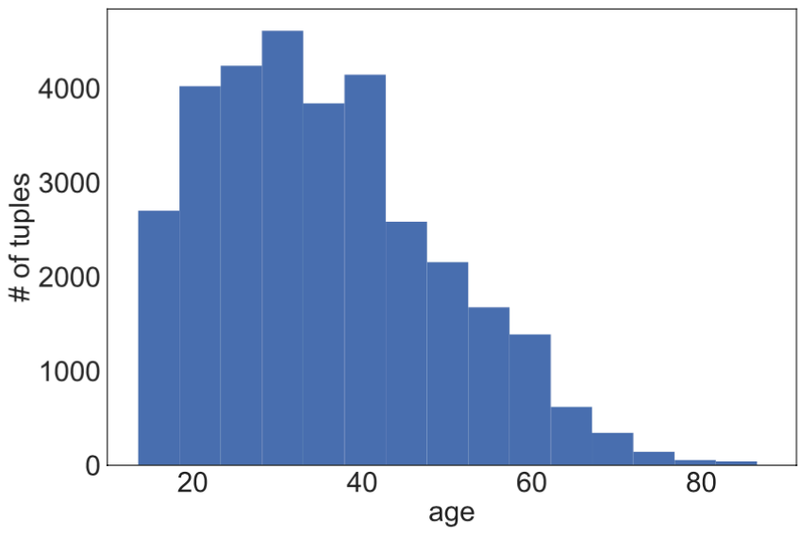


Figure 3: Histogram on *age*: synthetic.

Figure 4: Pair-wise correlations: Adult Income [6].

Figure 5: Pair-wise correlations: synthetic.

https://github.com/DataResponsibly/DataSynthesizer

Haoyue Ping, Julia Stoyanovich, Bill Howe: DataSynthesizer: Privacy-Preserving Synthetic Datasets. SSDBM 2017: 42:1-42:5

https://github.com/capitalone/synthetic-data

Towards Ground Truth Explainability on Tabular Data

Brian Barr [1] **Ke Xu** [2] **Claudio Silva** [2 3] **Enrico Bertini** [2] **Robert Reilly** [4] **C. Bayan Bruss** [5] **Jason D. Wittenbach** [6]

Abstract

In data science, there is a long history of using synthetic data for method development, feature selection and feature engineering. Our current interest in synthetic data comes from recent work in explainability. Today's datasets are typically larger and more complex - requiring less interpretable models. In the setting of *post hoc* explainability, there is no ground truth for explanations. Inspired by recent work in explaining image classifiers that does provide ground truth, we propose a similar solution for tabular data. Using copulas, a concise specification of the desired statistical properties of a dataset, users can build intuition around explainability using controlled data sets and experimentation. The current capabilities are demonstrated on three use cases: one dimensional logistic regression, impact of correlation from informative features, impact of correlation from redundant variables.

1. Introduction

The combination of large public datasets and novel machine learning architectures has provided state of the art predictive power in many diverse fields, such as computer vision and machine translation. These models are largely regarded as opaque *black-box* models. With their prevalence and increasing adoption, an active field of research is eXplainable Artificial Intelligence (XAI), which has sought to provide explanations for their predictions.

One avenue of research is on building interpretability into the model architecture (Kim et al., 2015; Canini et al., 2016; Lee et al., 2019). We focus on the area *post hoc* explanations – which occurs after model training. Currently there is no one size fits all solution. The method of explanation depends on model type (Chen et al., 2018; Krakovna & Doshi-Velez, 2016; Grath et al., 2018), desired granularity (Ribeiro et al., 2016; Ibrahim et al., 2019; Dhurandhar et al., 2018; Bhatt et al., 2020a; van der Linden et al., 2019), and audience (Wachter et al., 2017; Bhatt et al., 2020b).

In support of the methods, there are a growing number of packages that seek to provide an umbrella of methods such as AIX360 (Arya et al., 2019), ELI5 (TeamHG-Memex, 2019), and Alibi (Klaise et al., 2020).

Early methods were focused on explaining image classifiers. The use of sensitivities of the output class based on the input image pixels, provides a visual and immediately interpretable explanation for the classifier's prediction. For tabular data, the intuitive visual nature of sensitivities is not a natural metaphor. Additionally, where as computer vision typically relies on correlations between pixels as features, for tabular data that can be detrimental (Aas et al., 2019).

An ongoing challenge in XAI is the lack of ground truth. To add to the landscape of XAI, and move towards ground truth explainability for tabular data, we provide a flexible synthetic data generation method allowing generation of arbitrarily complex data. The current implementation is focused on the task of binary classification.

The paper is structured as follows: previous work is discussed in Section 2, Section 3 presents the method used to construct the synthetic tabular data, and Section 4 shows some results from three use cases: one dimensional logistic regression, impact of correlation from informative features, and impact of correlation from redundant variables.

2. Previous work

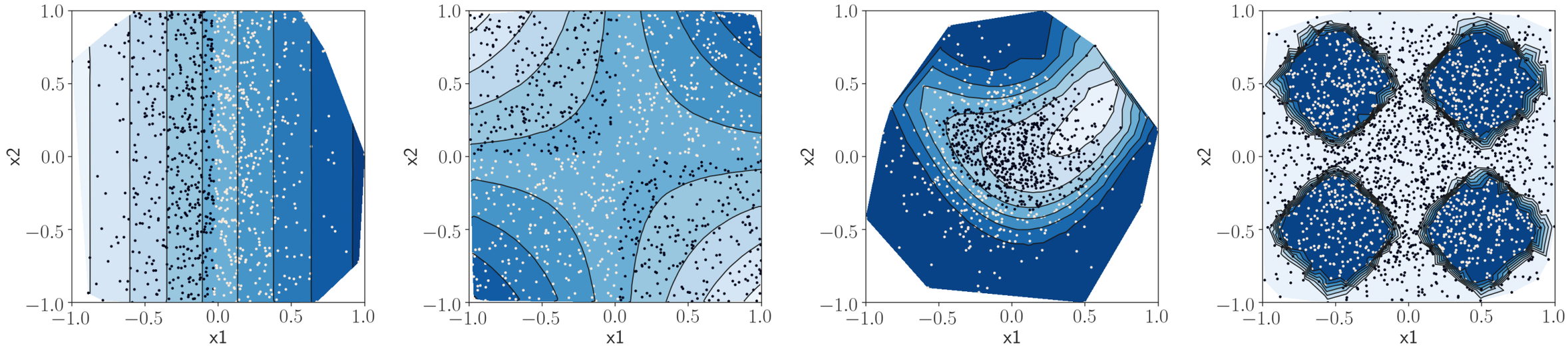
Early use cases of synthetic data focused on the tasks of feature and model selection (Guyon, 2003). This method is available in the scikit-learn (Pedregosa et al., 2011) module $make\_classification$. An alternative method of generating tabular data for classification is to generate clusters and apply class labels to them.

Another approach is to model joint probability $P(\boldsymbol{X}, y)$

[1]Center for Machine Learning, Capital One, New York, NY [2]Tandon School of Engineering, New York University, NY [3]Center for Data Science, New York University, NY [4]Card Machine Learning and Technology, Capital One, Plano, TX [5]Center for Machine Learning, Capital One, Maclean, VA [6]Center for Machine Learning, Capital One, Cambridge, MA. Correspondence to: Brian Barr <brian.barr@capitalone.com>, Claudio Silva <csilva@nyu.edu>.

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).



(a) (b) (c) (d)

Figure 1. Sample datasets - (a) linear, (b) nonlinear, (c) rosenbrock function, and (d) rastrigin function. The background has contours of probability with a scatter plot of the sampled points, colored by class. The datasets in (a) and (b) will be used later for experiments in Section 4.

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).

Synthetic data generation

We separate the input feature vectors, $\boldsymbol{X}$, into three categories - informative, redundant, and nuisance features.

$$\boldsymbol{X} = (\boldsymbol{X}_I | \boldsymbol{X}_r | \boldsymbol{X}_n) \tag{1}$$

Informative features, $\boldsymbol{X}_I$, used to determine the binary labels, are specified by a copula. A multivariate distribution can be separated into a set of marginal distributions and the correlation structure between them. Copula theory is the mathematical framework of the separation of the correlation

2020 ICML Workshop on
Human Interpretability in
Machine Learning (WHI 2020).

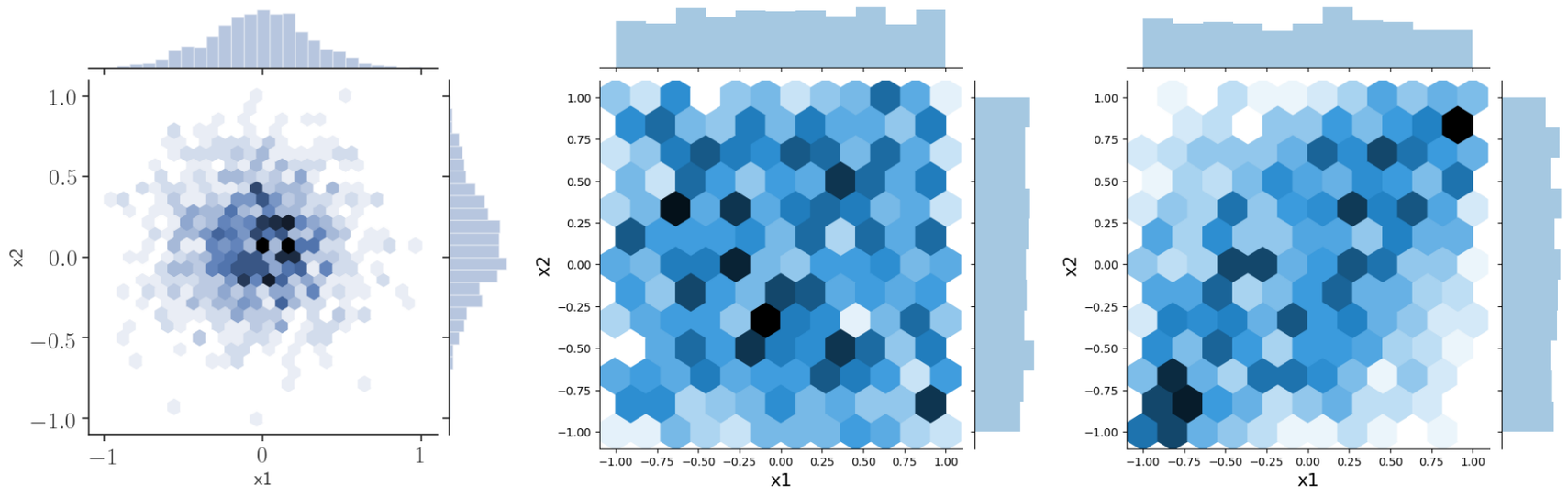
Synthetic data generation

We separate the input feature vectors, $\boldsymbol{X}$, into three categories - informative, redundant, and nuisance features.

$$\boldsymbol{X} = (\boldsymbol{X}_I | \boldsymbol{X}_r | \boldsymbol{X}_n) \tag{1}$$

Redundant features, $\boldsymbol{X}_r$, are a random linear combination of informative features. Nuisance features, $\boldsymbol{X}_n$ are random uncorrelated vectors drawn from the interval [-1, 1] which are useful as benchmarks to set the lower bound on explainability. Being purely random and not contributing to the specification of the labels, any feature found to have lower importance than a nuisance feature should be subjected to further scrutiny and possibly removed from the model.

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).



(a) (b) (c)

Figure 2. Joint probability plot for features x_1 and x_2 (a) with Gaussian marginals (b) with uniform marginals with no correlation and (c) uniform marginals with positive correlation.

For more details on copulas, see: https://twiecki.io/blog/2018/05/03/copulas/

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).

Synthesizing Tabular Data using Conditional GAN

by

Lei Xu

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In data science, the ability to model the distribution of rows in tabular data and generate realistic synthetic data enables various important applications including data compression, data disclosure, and privacy-preserving machine learning. However, because tabular data usually contains a mix of discrete and continuous columns, building such a model is a non-trivial task. Continuous columns may have multiple modes, while discrete columns are sometimes imbalanced, making modeling difficult.

To address this problem, I took two major steps.

(1) I designed `SDGym`, a thorough benchmark, to compare existing models, identify different properties of tabular data and analyze how these properties challenge different models. Our experimental results show that statistical models, such as Bayesian networks, that are constrained to a fixed family of available distributions cannot model tabular data effectively, especially when both continuous and discrete columns are included. Recently proposed deep generative models are capable of modeling more sophisticated distributions, but cannot outperform Bayesian network models in practice, because the network structure and learning procedure are not optimized for tabular data which may contain non-Gaussian continuous columns and imbalanced discrete columns.

(2) To address these problems, I designed `CTGAN`, which uses a conditional generative adversarial network to address the challenges in modeling tabular data. Because `CTGAN` uses reversible data transformations and is trained by re-sampling the data, it can address common challenges in synthetic data generation. I evaluated `CTGAN` on the benchmark and showed that it consistently and significantly outperforms existing statistical and deep learning models.

Thesis Supervisor: Kalyan Veeramachaneni
Title: Principal Research Scientist
Laboratory for Information and Decision Systems

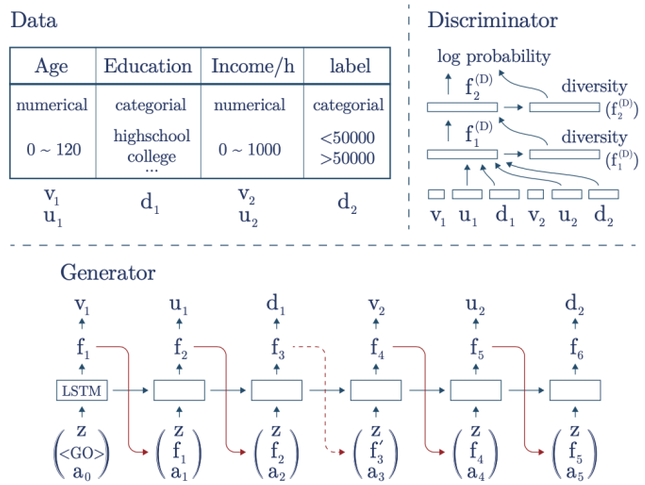


Figure 1: Example of using TGAN to generate a simple census table. The toy example has 2 continuous variables and 2 discrete variables. Our model generates these 4 variables one by one following their original order in the table. Each sample is generated in 6 steps. Each numerical variable is generated in 2 steps while each categorical variable is generated in 1 step. The discriminator concatenates all features together and uses Multi-Layer Perceptron (MLP) to distinguish real and fake data.

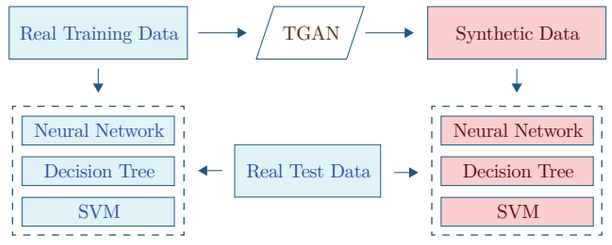


Figure 2: The process of training and evaluating TGAN. The real training data, including the *labels*, is used to learn a GAN and generate synthetic data. Several machine learning models (We choose 5 methods.) are learned using the real training data and the synthetic data. The learned models' accuracy is tested on the real test data that was set aside.

https://arxiv.org/pdf/1811.11264.pdf

https://dai.lids.mit.edu/wp-content/uploads/2020/02/Lei_SMThesis_neo.pdf

Computer Science > Machine Learning

[Submitted on 25 Sep 2019]

Synthetic Data for Deep Learning

Sergey I. Nikolenko

Synthetic data is an increasingly popular tool for training deep learning models, especially in computer vision but also in other areas. In this work, we attempt to provide a comprehensive survey of the various directions in the development and application of synthetic data. First, we discuss synthetic datasets for basic computer vision problems, both low-level (e.g., optical flow estimation) and high-level (e.g., semantic segmentation), synthetic environments and datasets for outdoor and urban scenes (autonomous driving), indoor scenes (indoor navigation), aerial navigation, simulation environments for robotics, applications of synthetic data outside computer vision (in neural programming, bioinformatics, NLP, and more); we also survey the work on improving synthetic data development and alternative ways to produce it such as GANs. Second, we discuss in detail the synthetic-to-real domain adaptation problem that inevitably arises in applications of synthetic data, including synthetic-to-real refinement with GAN-based models and domain adaptation at the feature/model level without explicit data transformations. Third, we turn to privacy-related applications of synthetic data and review the work on generating synthetic datasets with differential privacy guarantees. We conclude by highlighting the most promising directions for further work in synthetic data studies.

Comments:	156 pages, 24 figures, 719 references
Subjects:	**Machine Learning (cs.LG)**; Cryptography and Security (cs.CR); Computer Vision and Pattern Recognition (cs.CV)
Cite as:	arXiv:1909.11512 **[cs.LG]**
	(or arXiv:1909.11512v1 **[cs.LG]** for this version)

Submission history

From: Sergey Nikolenko [view email]
[v1] Wed, 25 Sep 2019 14:20:57 UTC (28,811 KB)

https://arxiv.org/abs/1909.11512

Contents

Name	Year	Ref	Size / comments
			Low-level computer vision
Tsukuba Stereo	2012	[450]	1800 high-res stereo image pairs
MPI-Sintel	2012	[79]	Optical flow from an animated movie
Middlebury 2014	2014	[525]	33 high-res stereo datasets
Flying Chairs	2015	[152]	22K frame pairs with ground truth flow
Flying Chairs 3D	2015	[400]	22K stereo frames
Monkaa	2015	[400]	8591 stereo frames
Driving	2015	[400]	4392 stereo frames
UnrealStereo	2016	[700]	Data generation software
Underwater	2018	[431]	Underwater synthetic stereo pairs generator
			Datasets of basic objects
YCB	2015	[467]	77 objects in 5 categories
ShapeNet	2015	[93]	>3M models, 3135 categories, rich annotations
ShapeNetCore	2017	[672]	51K manually verified models from 55 categories
UnrealCV	2017	[467]	Plugin for UE4 to generate synthetic data
VANDAL	2017	[87]	4.1M depth images, >9K objects in 319 categories
SceneNet	2015	[232]	Automated indoor synthetic data generator
SceneNet RGB-D	2017	[401]	5M RGB-D images from 16K 3D trajectories
DepthSynth	2017	[455]	Framework for realistic simulation of depth sensors
PartNet	2018	[415]	26671 models, 573535 annotated part instances
Falling Things	2018	[594]	61.5K images of YCB objects in virtual envs
ADORESet	2019	[42]	Hybrid dataset for object recognition testing
			Datasets of synthetic people
ViHASi	2008	[475]	Silhouette-based action recognition
Agoraset	2014	[127]	Crowd scenes generator
LCrowdV	2016	[110]	1M videos, 20M frames with crowds
PHAV	2017	[132]	40K videos for action recognition (35 categories)
SURREAL	2017	[609]	145 subjects, 2.6K sequences, 6.5M frames
SyRI	2018	[32]	Virtual humans in UE4 with realistic lighting
GCC	2019	[625]	15K images with 7.6M subjects

Table 1: An overview of synthetic datasets discussed in Section 2.

Contents

Name	Year	Ref	Engine	Size / comments
		Outdoor urban environments, driving		
TORCS	2014	[651]	Custom	Game-based simulation engine
Virtual KITTI	2016	[184]	Unity	5 environments, 50 videos
GTAVision	2016	[292]	GTA V	GTA plugin, 200K images
SYNTHIA	2016	[499]	Unity	213K images
GTAV	2016	[494]	GTA V	25K images
VIPER	2017	[493]	GTA V	254K images
CARLA	2017	[153]	UE	Simulator
VIES	2018	[515]	Unity3D	61K images, 5 environments
ParallelEye	2018	[585]	Esri	Procedural gen, import from OSM
VIVID	2018	[341]	UE	Urban sim with emphasis on people
DeepDrive	2018	[469]	UE	Driving sim + 8.2h of videos
PreSIL	2019	[269]	GTA V	50K images with LIDAR point clouds
AADS	2019	[356]	Custom	3D models of cars on real backgrounds
WoodScape	2019	[674]	Custom	360° panoramas with fisheye cameras
ProcSy	2019	[317]	Esri	Procedural generation with varying conditions
		Robotic simulators and aerial navigation		
Gazebo	2004	[324]	Custom	Industry standard robotic sim
MuJoCo	2012	[589]	Custom	Common physics engine for robotics
AirSim	2017	[539]	UE	Sensor readings, hardware-in-the-loop
CAD^2RL	2017	[513]	Custom	Indoor flying sim
X-Plane	2019	[555]	X-Plane	8K landings, 114 runways
Air Learning	2019	[332]	—	Platform for flying sims
VRGym	2019	[658]	UE	VR for human-in-the-loop training
ORRB	2019	[112]	Unity	Accurate sim used to train real robots
		Indoor environments		
ICL-NUIM	2014	[233]	Custom	RGB-D with noise models, 2 scenes
SUNCG	2016	[557]	Custom	45K floors, 3D models
MINOS	2017	[521]	SUNCG	Indoor sim based on SUNCG
AI2-THOR	2017	[325]	Unity3D	Indoor sim with actionable objects
House3D	2018	[647]	SUNCG	Indoor sim based on SUNCG
Habitat	2019	[393]	Custom	Indoor sim platform and library

Table 2: An overview of synthetic datasets and virtual environments discussed in Section 3.

Project Ideas / Discussions

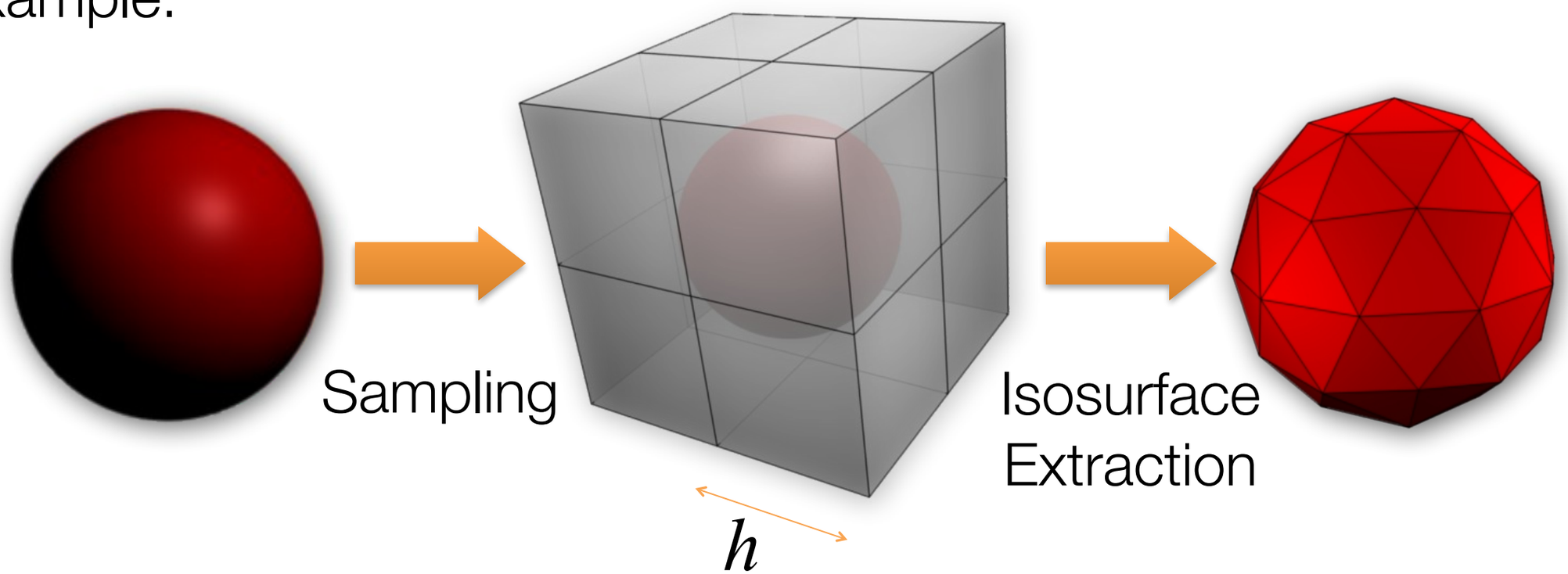
Potential project:

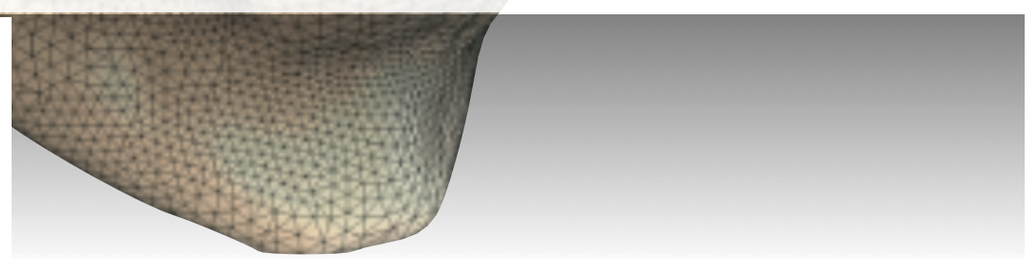
Studying the properties and correctness of ML implementations through the use of synthetic data and the “method of manufactured solutions”?

Isosurface Extraction

Let $f : \mathbb{R}^3 \to \mathbb{R}$ be a smooth function. The isosurface S_k associated with an isovalue k is defined as $S_k = \{x \in \mathbb{R}^3 | f^{-1}(k) = x\}$.

3D Example:





NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Expected Order of Accuracy

- Given an isosurface f = k and assuming linear interpolation
  - Algebraic distance:

$$\left\| f^h(\vec{x}) - f(\vec{x}) \right\| = O(h^2)$$

  - Normal (cross product of edges):

$$\left\| \nabla f^h(\vec{x}) - \nabla f(\vec{x}) \right\| = O(h)$$

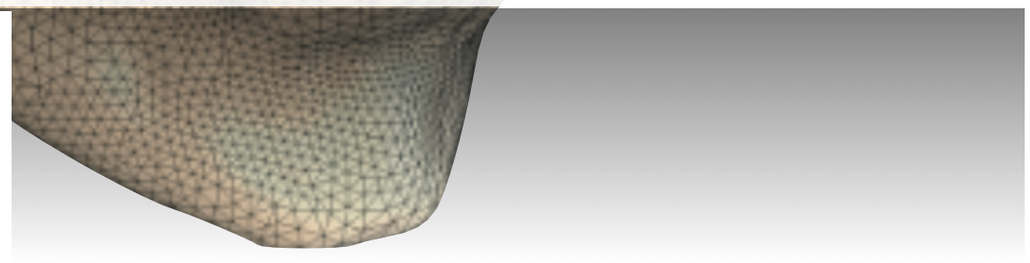
  - Gaussian curvature (angle deficit method):

$$\left\| \kappa(f^h) - \kappa(f) \right\| = O(1)$$

  - Surface area:

$$\left\| area(f^h) - area(f) \right\| = ?$$

Verifiable Visualization for Isosurface Extraction, T. Etiene, C. Scheidegger, L. G. Nonato, R. M. Kirby, and C. Silva. IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE Visualization 2009).

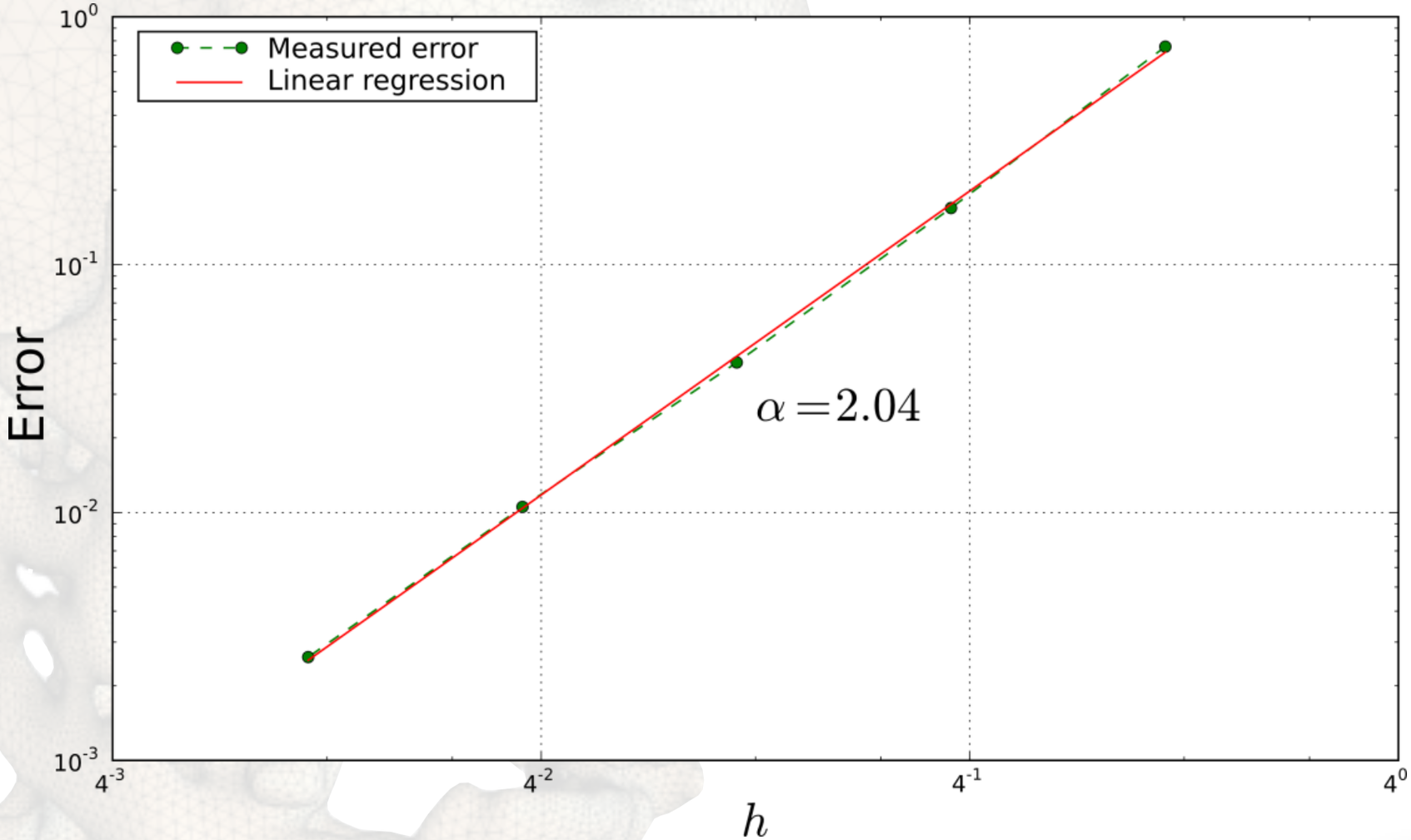


Observed Order of Accuracy

- In practice:

$$E_h = \left\| u^h(x) - u(x) \right\| = ch^{\alpha}$$

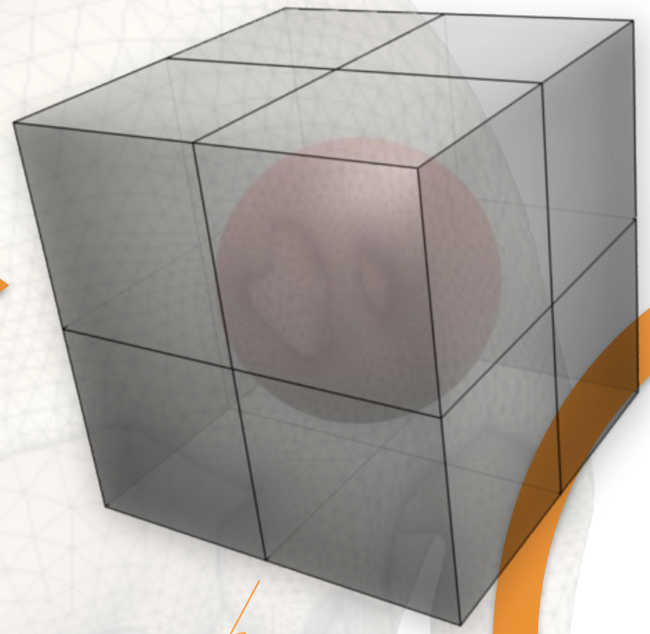
$$\log(E_h) = \log(c) + \alpha \log(h)$$



Method of Manufactured Solution

Manufactured solution.

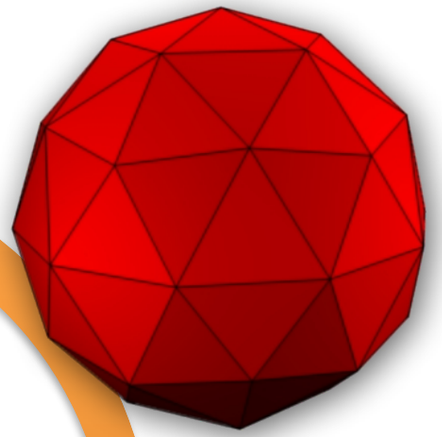
Sampling



h_i

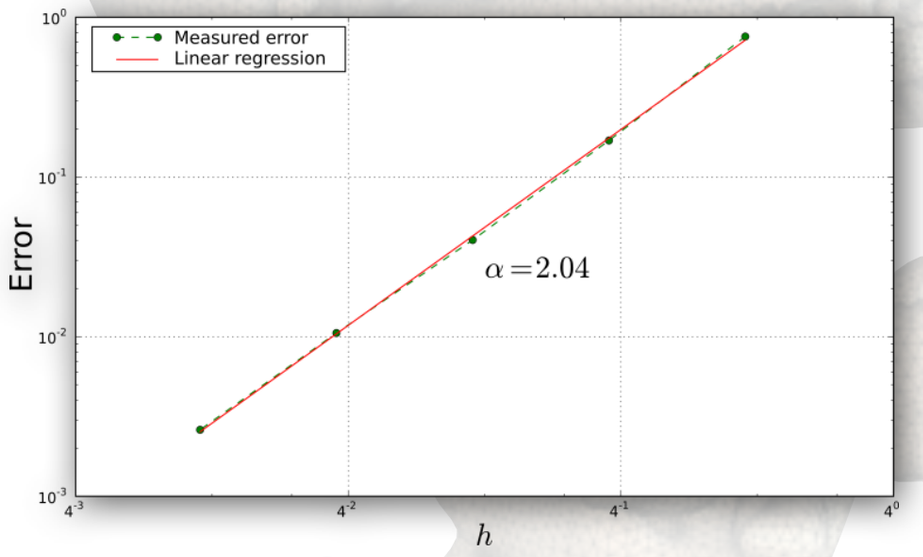
Grid refinement

Isosurfacing



Error computation

$$E_{h_i} = \| u(\quad) - u(\quad) \|_L$$



(h_i, E_{h_i})

Convergence

NEW YORK UNIVERSITY

NYU·poly

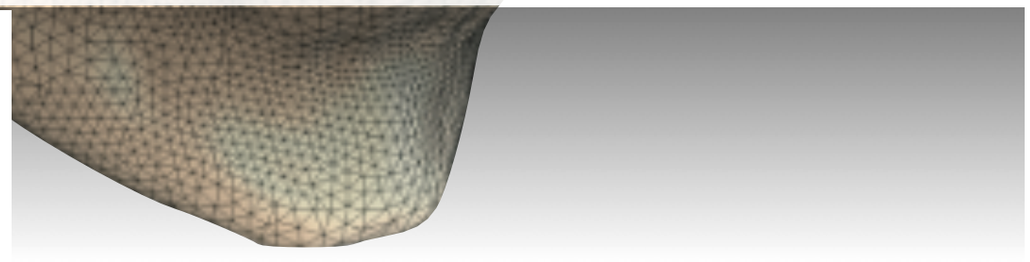
POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Implementations under verification

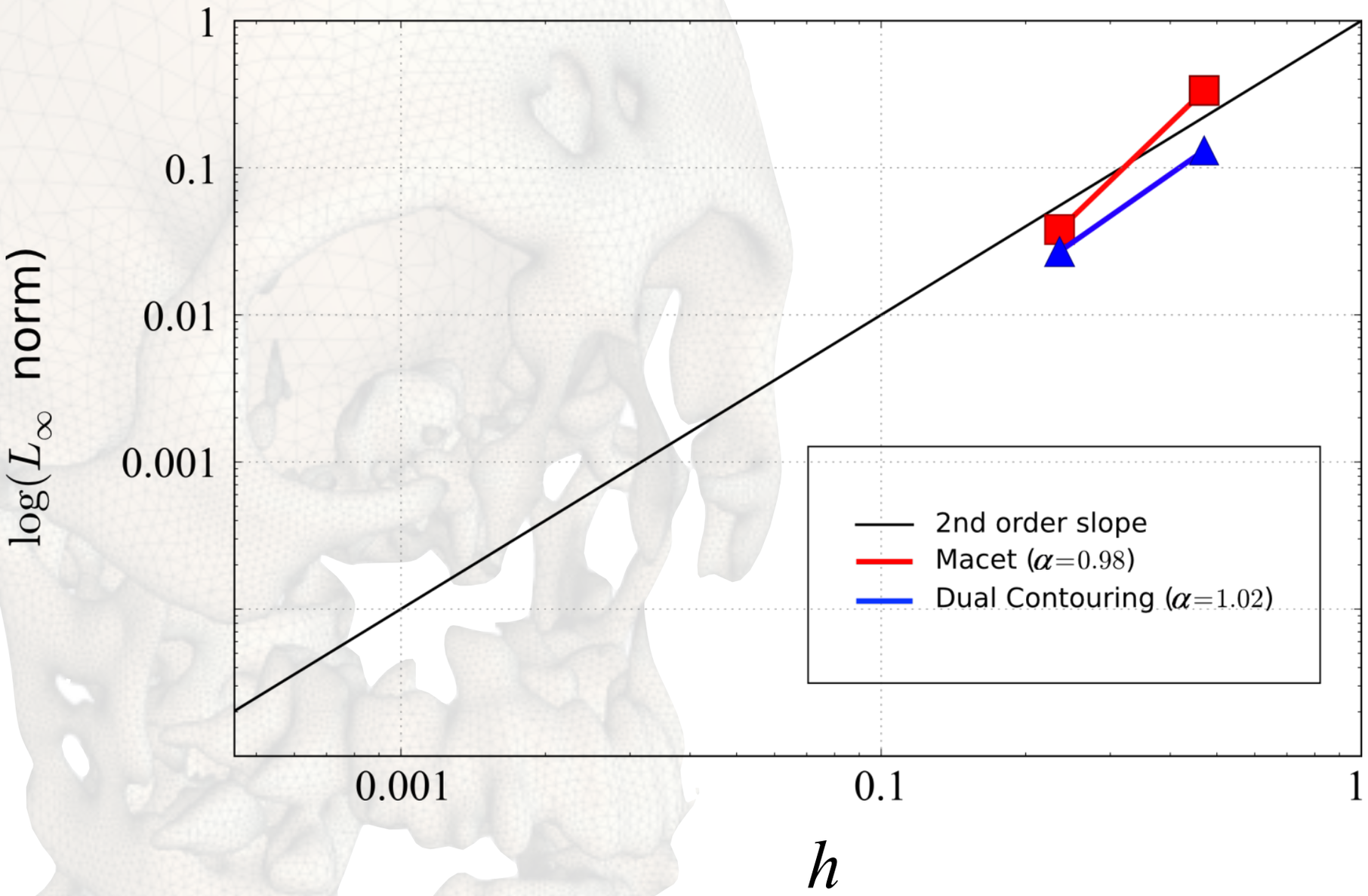
- VTK Marching Cubes [W. Lorensen and H. Cline, 1987]
- Macet [C. A. Dietrich et. al., 2008]
- Dual Contouring [T. Ju et. al., 2002]
- SnapMC [S. Raman and R. Wenger, 2008]
- Afront [J. Schreiner et. al., 2006]
- Dellso [T. Dey and J. Levine, 2007]

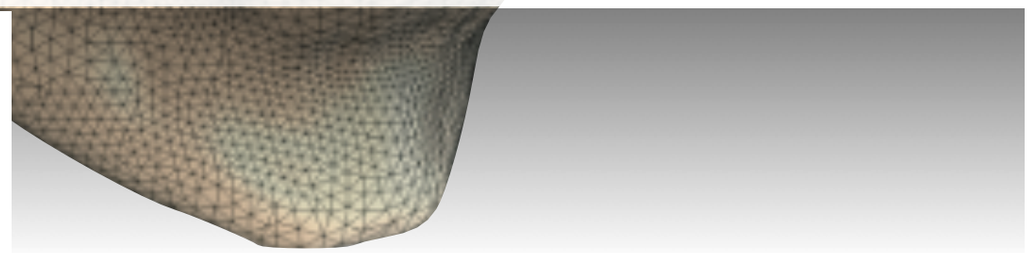
- Manufactured Solution:

$$f(x, y, z) = x^2 + y^2 + z^2 - 1 = 0$$



Observed order of accuracy Algebraic distance



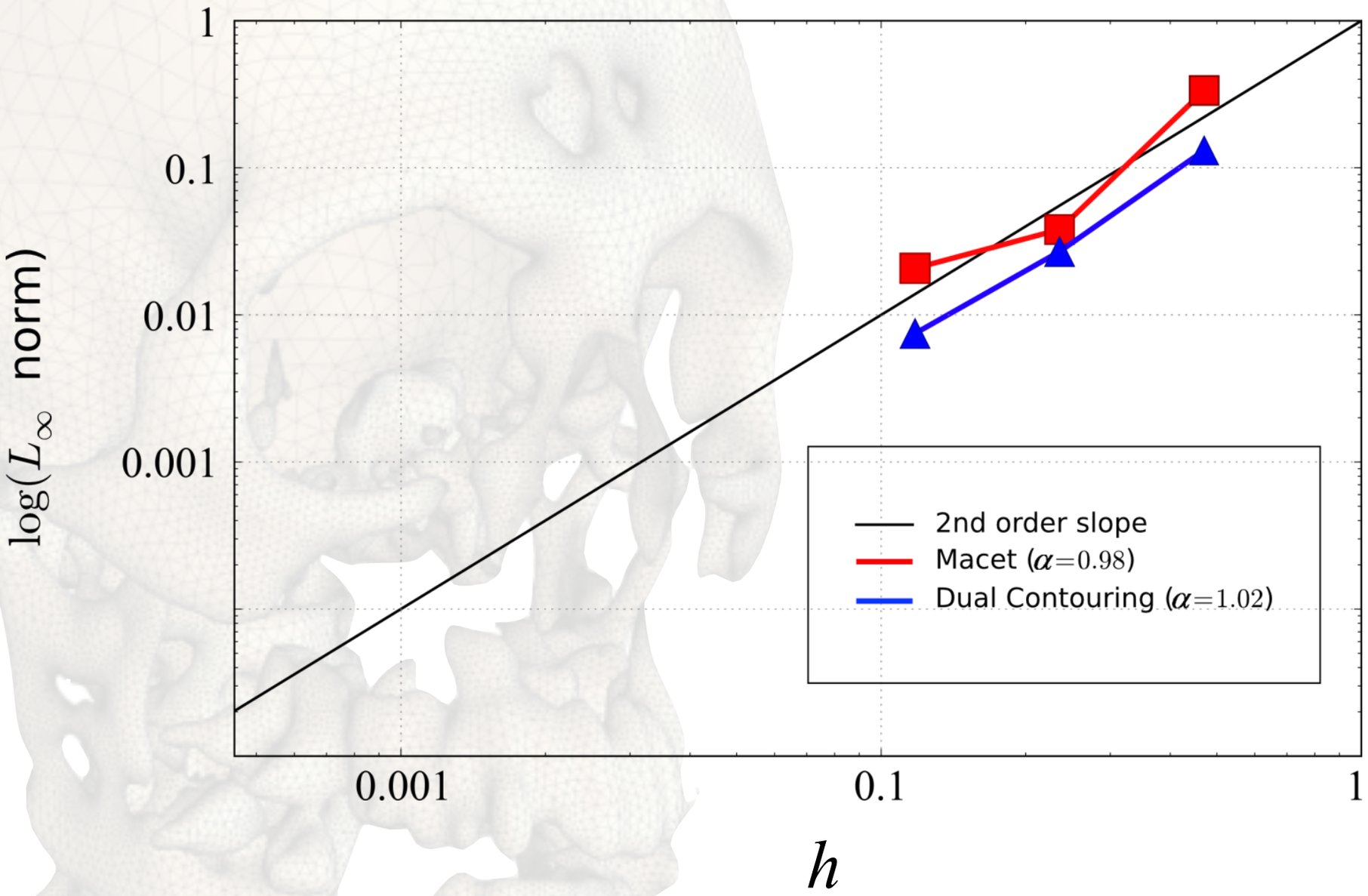


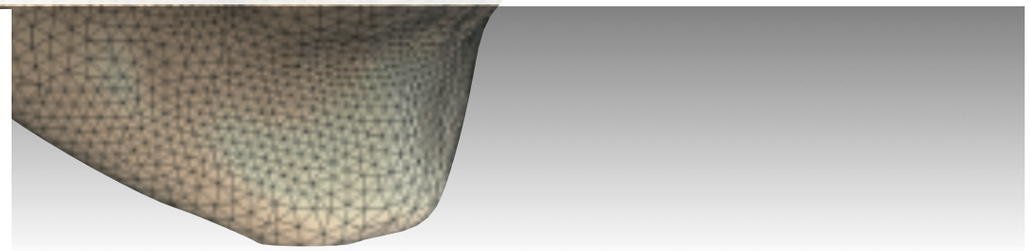
NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Observed order of accuracy Algebraic distance



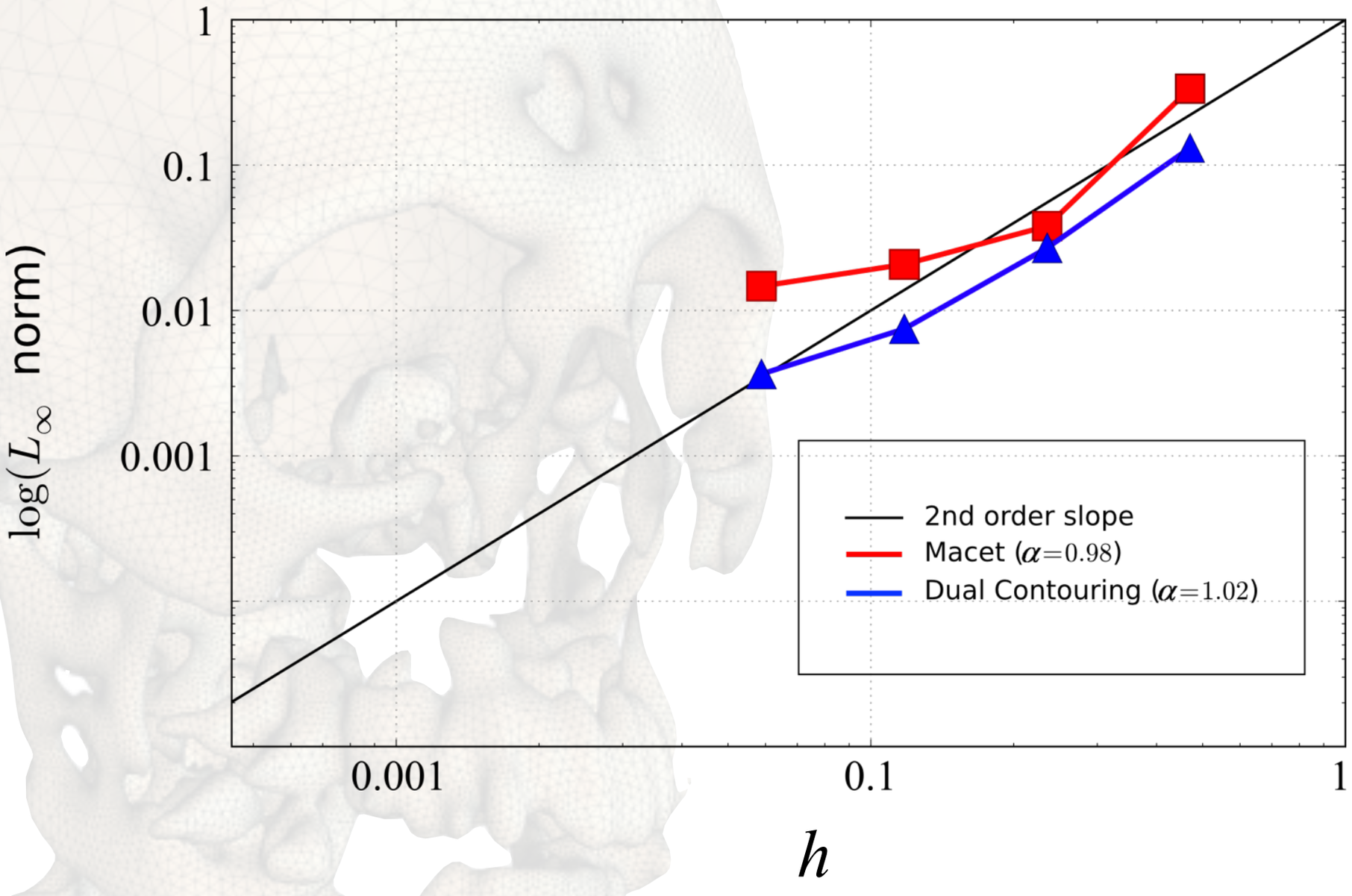


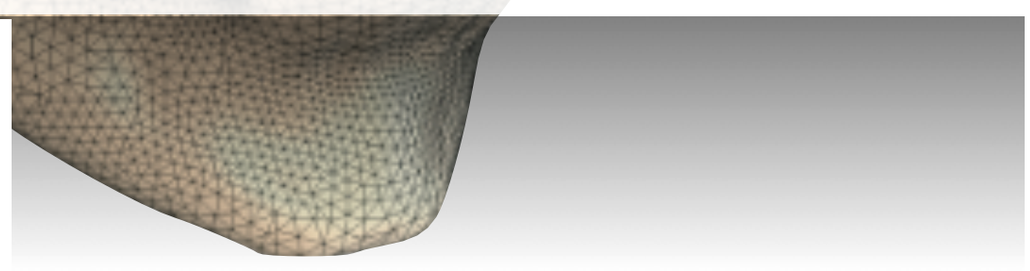
NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Observed order of accuracy Algebraic distance



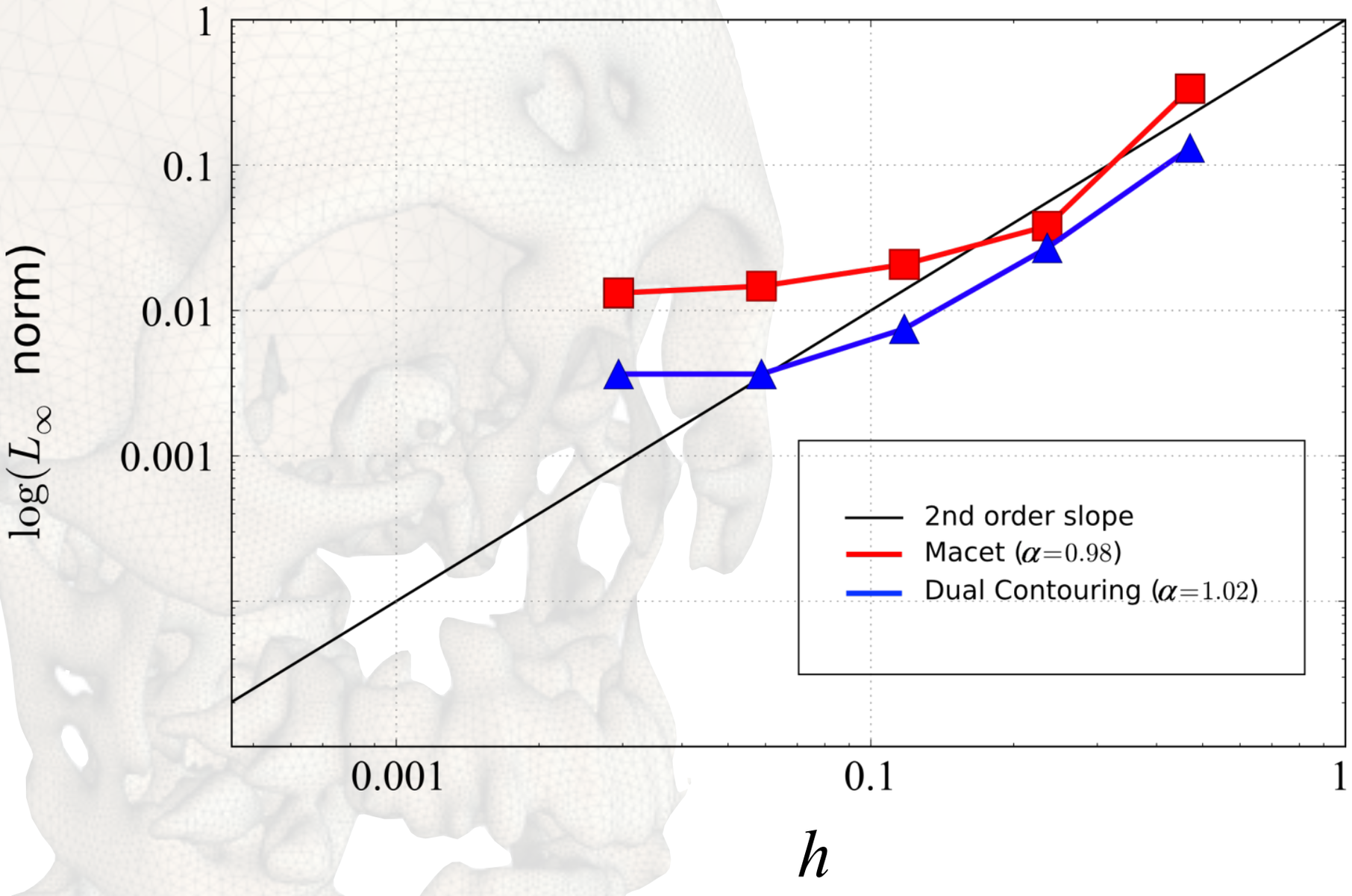


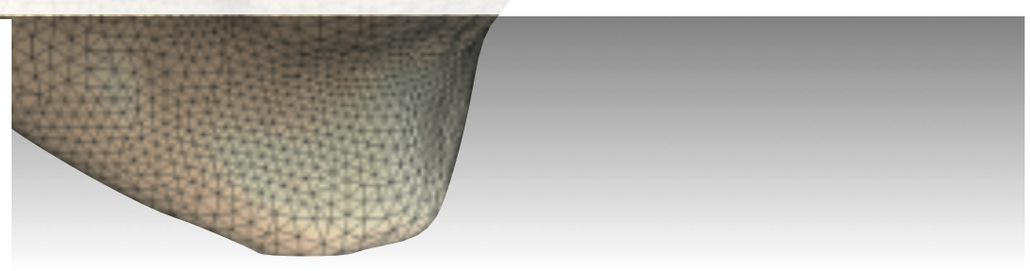
NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Observed order of accuracy Algebraic distance



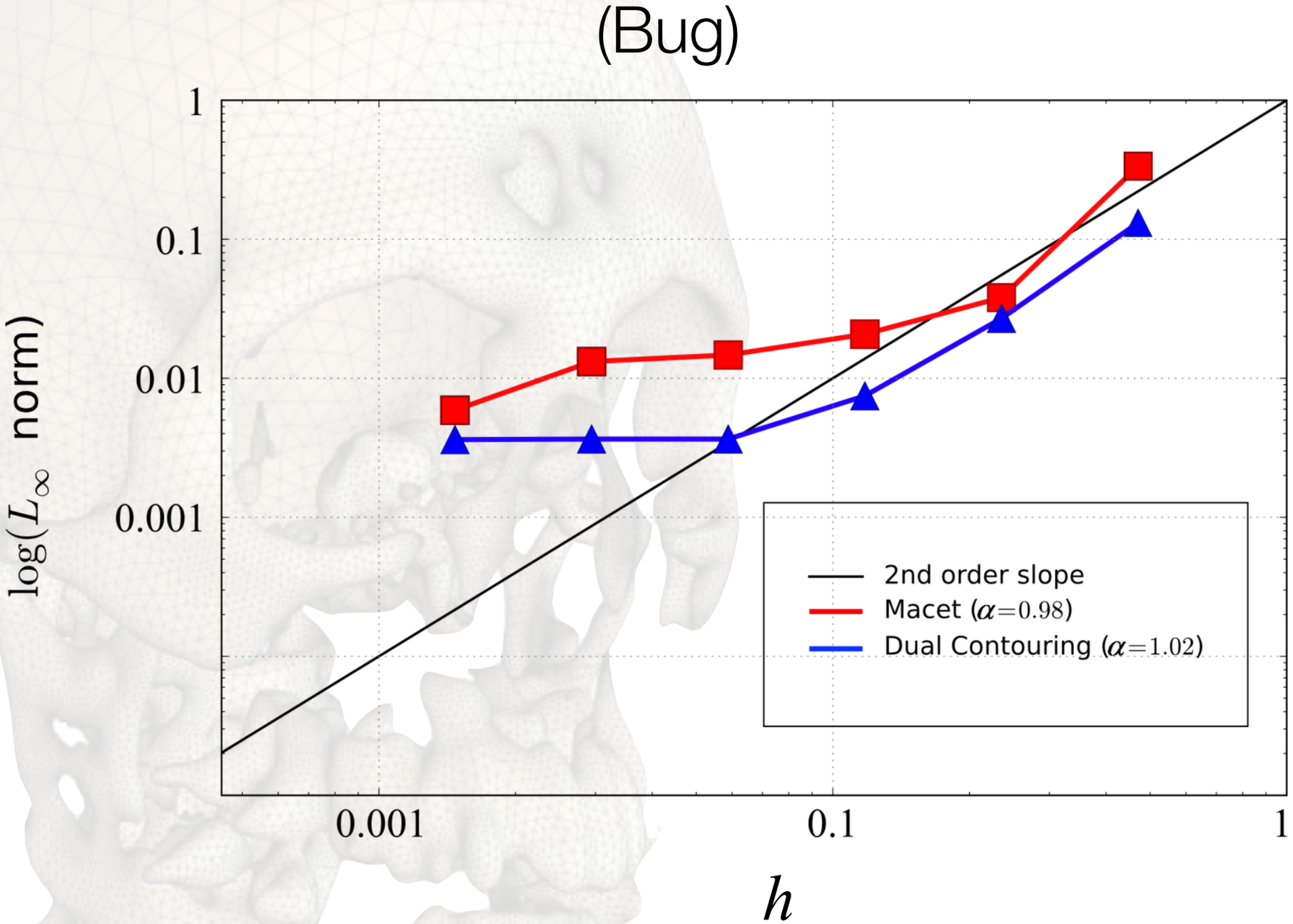


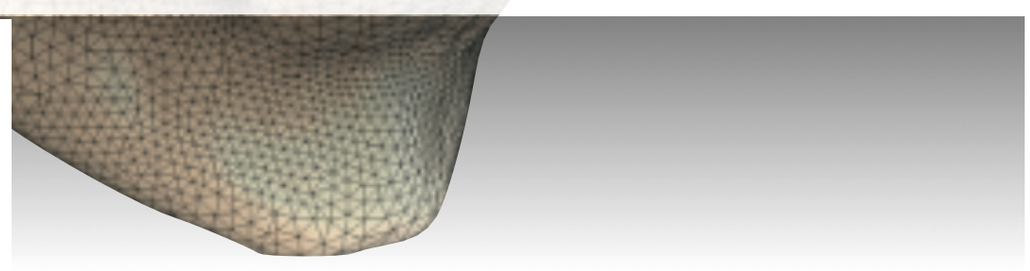
NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Observed order of accuracy Algebraic distance





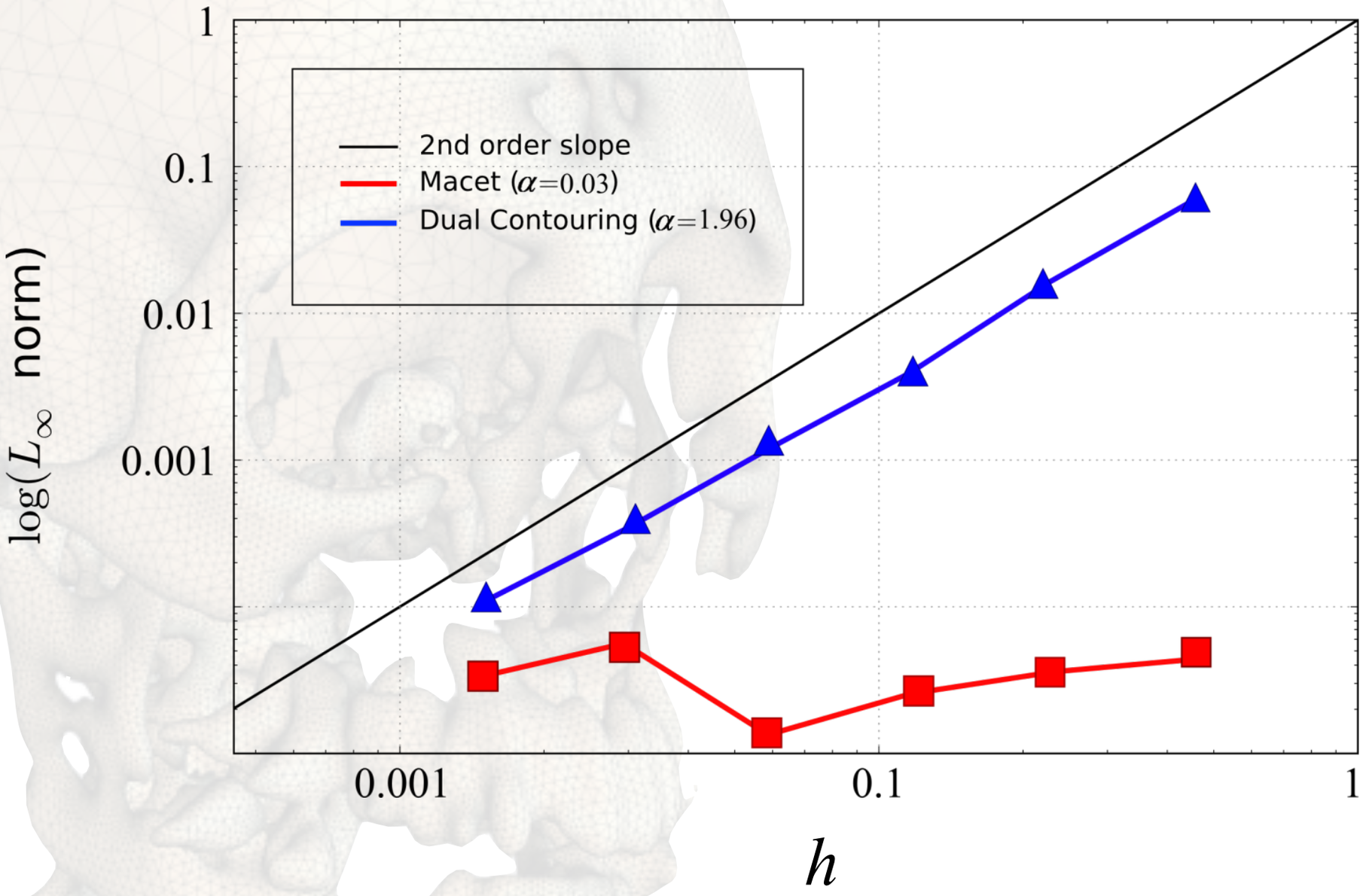
NEW YORK UNIVERSITY

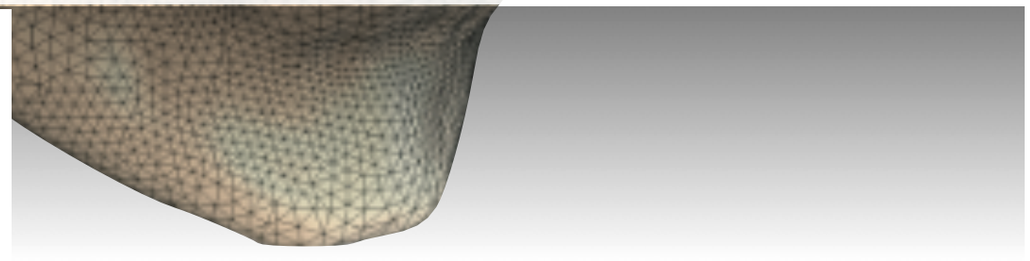
NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Observed order of accuracy Algebraic distance

(Fixed)





NEW YORK UNIVERSITY

NYU·poly

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

Potential project:

Studying the properties and correctness of ML implementations through the use of synthetic data and the “method of manufactured solutions”?

IDEAS?

Project Ideas / Discussions

A New Approach for Pedestrian Density Estimation Using Moving Sensors and Computer Vision

Sensing the city

ERIC K. TOKUDA, University of São Paulo, Brazil
YITZCHAK LOCKERMAN, New York University, USA
GABRIEL B. A. FERREIRA, University of São Paulo, Brazil
ETHAN SORRELGREEN and DAVID BOYLE, Carmera, USA
ROBERTO M. CESAR-JR., University of São Paulo, Brazil
CLAUDIO T. SILVA, New York University, USA

An understanding of person dynamics is indispensable for numerous urban applications, including the design of transportation networks and planning for business development. Pedestrian counting often requires utilizing manual or technical means to count individuals in each location of interest. However, such methods do not scale to the size of a city and a new approach to fill this gap is here proposed. In this project, we used a large dense dataset of images of New York City along with computer vision techniques to construct a spatio-temporal map of relative person density. Due to the limitations of state-of-the-art computer vision methods, such automatic detection of person is inherently subject to errors. We model these errors as a probabilistic process, for which we provide theoretical analysis and thorough numerical simulations. We demonstrate that, within our assumptions, our methodology can supply a reasonable estimate of person densities and provide theoretical bounds for the resulting error.

CCS Concepts: • **Information systems → Data mining**;

Additional Key Words and Phrases: Computer vision, objects detection, urban computing, simulation, agent-based modelling

ACM Reference format:
Eric K. Tokuda, Yitzchak Lockerman, Gabriel B. A. Ferreira, Ethan Sorrelgreen, David Boyle, Roberto M. Cesar-Jr., and Claudio T. Silva. 2020. A New Approach for Pedestrian Density Estimation Using Moving Sensors and Computer Vision. *ACM Trans. Spatial Algorithms Syst.* 6, 4, Article 26 (July 2020), 20 pages.
https://doi.org/10.1145/3397575

Fig. 1. Sample of the images provided by Carmera. More than 10M images have been used for the people density estimation evaluated in this work.

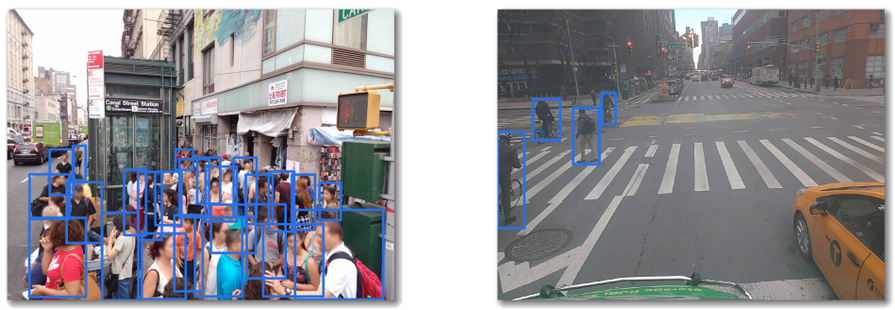


Fig. 9. Person detection by the method in special cases. On the left, person detection in a crowded scene. On the right, pedestrians and non-pedestrians (a cyclist) are equally identified by the method.

E. K. Tokuda and Y. Lockerman contributed equally to the article.
This work was supported in part by: NSF awards CNS-1229185, CCF-1533564, CNS-1544753, CNS-1730396, CNS-1828576; FAPESP (grants #14/24918-0 and #2015/22308-2); CNPq and CAPES; the Moore-Sloan Data Science Environment at NYU, and C2SMART. C. T. Silva is partially supported by the DARPA D3M program.
Authors' addresses: E. K. Tokuda, G. B. A. Ferreira, and R. M. Cesar-Jr., University of São Paulo, Rua do Matao, 1010, São Paulo, 05508-090, Brazil; emails: {tokudaek, gabriel.augusto.ferreira, rmcesar@}usp.br; Y. Lockerman and C. T. Silva, New York University, 2 MetroTech Center, NY, 11201; emails: {ydl214, csilva}@nyu.edu; E. Sorrelgreen, Carmera, 1100 NE Campus Pkwy Suite 200, Washington, 98105; email: ethan@carmera.co; D. Boyle, Carmera, 20 Jay St Suite 312, New York, 11201; email: david@carmera.co.

2374-0353/2020/07-ART26 $15.00
https://doi.org/10.1145/3397575

Visual Road

A Video Data Management Benchmark

Hyperparameters:
scale, resolution,
duration, seed

Configure

Generate

Simulated Dataset

Video
Dataset Generator

Benchmark Queries &
Verifier

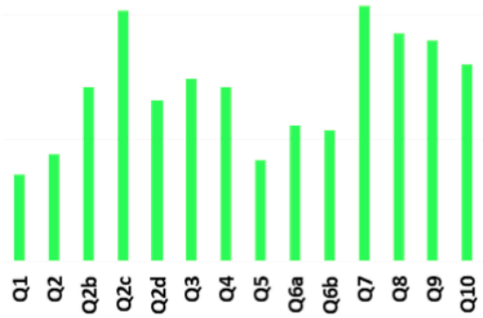
Execute

Verify

Video Data
Management System

UNREAL ENGINE
Commercial Gaming
Engine

CARLA
Autonomous Driving
Simulator



https://db.cs.washington.edu/projects/visualroad/

Potential project:

IDEAS?