Multivariate Gaussian Generative Models and Chamfer Distance for Point Cloud Object Detection

Ryan M. Sander*
6.885 Final Project
Massachusetts Institute of Technology
Cambridge, MA 02139
rmsander@mit.edu

Vikash K. Mansinghka
MIT Probabilistic Computing Project
Massachusetts Institute of Technology
Cambridge, MA 02139
rmsander@mit.edu

Abstract

Object detection is a crucial supervised learning task for visual-based intelligence, particularly for performing feature extraction and developing an efficient and compact semantic understanding of a 3D point cloud scene. In this work, we apply a generative Bayesian inference approach to infer object bounding boxes using Gen, a novel probabilistic programming language. Namely, we combine a multivariate Gaussian generative bounding box model with a Chamfer distance metric to generate a posterior distribution over bounding boxes, which we then sample from the origin to find the best proposed bounding boxes.

1 Introduction

1.1 Motivation

Object detection is a pertinent task in applications ranging from autonomous vehicles to real-time robotic object manipulation. While many state-of-the-art frameworks use neural-based approaches for object detection, this work seeks to apply a generative approach to estimating bounding boxes of objects using point cloud data. Although novel applications have evolved recently to improve the data labeling and curation processes for supervised learning approaches with 3D point clouds [9], curating class-balanced and accurately labeled datasets for customized point cloud learning applications can be significantly resource and time-intensive. Furthermore, for certain three-dimensional object detection tasks, there are limited examples that a framework can be trained on, and as a result, supervised learning approaches such as neural networks may not be able to learn a rich enough data distribution to perform well. Our work seeks to offer an alternative to the supervised learning approach using generative modeling, Bayesian inference, and Chamfer distance, a point set metric.

Specifically, using a multivariate Gaussian generative model for point clouds in tandem with a Chamfer distance proxy for likelihood, we will construct a posterior distribution over parameters, and then sample from this distribution at locations that minimize the Chamfer distance between a point-transformed bounding box and an observed point cloud. Unlike many object detection approaches, this generative approach develops a full posterior distribution over our bounding box parameters, enabling bounding box proposals to be sampled during inference, rather than just inferred directly using Maximum Likelihood estimation. This enables for us to sample from many proposals before selecting one our framework concludes is optimal, and (2) Allows us to quantify how well our bounding box fits our framework using a "goodness of fit".

*GitHub, Website

Final Project, MIT 6.885: Probabilistic Programming and Artificial Intelligence.

1.2 Overview of Object Detection Inference System

Our generative modeling and inference system is composed of three stages:

1. In the first stage, we use importance sampling [4] with a multivariate Gaussian generative model for point cloud bounding boxes. This stage is designed to choose bounding box proposals that are centered around the mean locations of the observed point cloud. We leverage custom proposals that constrain the centered location of the point cloud, but because we want to develop a rich posterior distribution over bounding box dimensions, we do not constrain the dimensions of our sampled bounding box proposals.
2. After sampling many traces, we compute the Chamfer distance between a point-transformed bounding box (a bounding box that has been discretized into the same number of points as the input point cloud) the second stage of our framework. We leverage the Chamfer distance metric, a popular point-set distance metric, as a proxy for bounding box likelihood.
3. Our final stage samples from our distribution over Chamfer distance values conditioned on these distances being zero, and allows us to find bounding boxes that achieve both (1) A centering on the data, and (2) Have dimensions that fit tightly around the point cloud.

A diagram of our system can be found below:

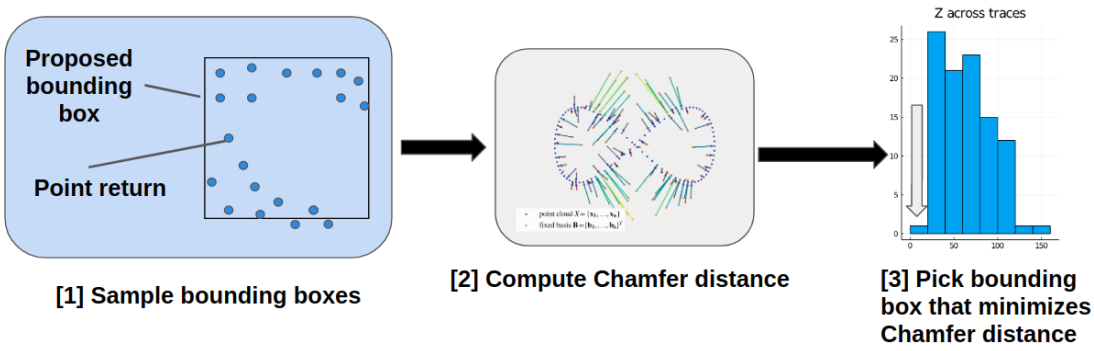

Figure 1: Our generative modeling and inference framework for object detection.

Using this multi-stage unsupervised generative model and inference framework, we are able to learn posterior distributions over bounding boxes that have a variety of different fits over our observed point cloud. We discuss the specifics of each stage of this system in detail using synthetically-generated point clouds.

2 Related Work

2.1 Supervised Neural-Based Approaches

Neural and energy-based approaches have yielded substantial success with supervised object detection tasks on three-dimensional point clouds. Our approach aims to build off of these successful frameworks, while also extending them to the unsupervised, generative Bayesian learning domain.

PointRCNN [7] is a two-stage neural network framework that takes a bottom-up approach to object detection. In the first stage, the point cloud is segmented, and many object bounding box proposals are made. In the second stage, these bounding box proposals are refined. This approach leverages some of the same generative aspects introduced in our object detection framework, namely bounding box proposal generation and refinement. However, unlike our approach, this framework requires supervised learning via semantic segmentation of the point cloud during the proposal generation phase.

Other approaches combine both neural and energy-based methods. In [2], the authors pose the object detection problem as an energy minimization problem over a learned weighted combination of several key metrics, such as point cloud density, free space, a height prior, and a height contrast (note that the last two are for a proposed point cloud bounding box). These weights are learned via a supervised Support Vector Machine algorithm. After these bounding box proposals are generated using this energy minimization framework, these proposals are scored using a Convolutional Neural Network. Like our approach, this paper leverages the use of metrics as proxies for evaluating the "goodness of fit" of bounding boxes for observed point clouds - in our approach, we focus on the use of Chamfer distance. However, the optimal weighting of the metric features leveraged in this approach needs to be learned via supervised learning, whereas our hyperparameters for minimum and maximum variance can be generalized when point clouds are normalized.

2.2 Generative Approaches for 2D Object Detection

Approaches for two-dimensional object detection using generative modeling also provide us with insight for how we can design our framework and improve unsupervised object detection. In [5], the authors develop a GAN-based object detection framework designed for few-shot learning by learning not only a discriminative model, but a generative model as well. The generative component of this approach, however, is achieved using a deep autoencoder, wheareas ours is a bottom-up data-driven proposal multivariate Gaussian generative model.

With this literature framing, we are ready to discuss our generative framework.

3 Learning a Generative Point Cloud Model

Our goal with this paper is to learn a generative model for a point cloud, which is a scene consisting of three-dimensional point returns. Each data point contains x, y, and z coordinates. While most approaches today strive for learning discriminative models using techniques such as supervised neural networks, we aim to learn a generative model using Bayesian inference and probabilistic programming that we can sample from for generating intelligent bounding box proposals.

To learn a distribution over suitable bounding boxes, we first need a representation for the likelihood that a given bounding box generated a dataset. Unlike time series forecasting generative approaches, however, each point in a point cloud sampled from a generative bounding box model has little correspondence with each point in an observed point cloud; thus, computing the log likelihood of this model is not trivial. Rather than computing an exact, closed-form expression of this log likelihood, we instead use a combination of a generative multivariate Gaussian model in tandem with the Chamfer distance metric between our point sets.

We aim to solve this unsupervised object detection problem using a three-stage Bayesian approach: (1) Applying importance sampling to a Multivariate Gaussian generative model, (2) Construct an empirical distribution over Chamfer distances between bounding boxes and observed point clouds, and (3) Sample from this empirical distribution near $z = 0$ to generate intelligent bounding box proposals.

3.1 Multivariate Gaussian Generative Model and Importance Sampling

We will use importance sampling to generate initial proposals for bounding boxes on a particular point cloud. This generative probabilistic technique is used to estimate parameters that explain data from one distribution, denoted $p(x)$ or the *nominal distribution*, using a different distribution, denoted $q(x)$ or the *importance distribution*. In our case, we will importance sampling for generating candidate bounding box proposals that are roughly centered around the mean of the observed data. For this, we use a multivariate Gaussian generative model, as well as a custom proposal distribution that constrains the range over three-dimensional center points of the bounding box, but introduces no restrictions over the actual dimensions of the bounding box.

Our multivariate Gaussian generative model generates the mean of the dataset around the center of a proposed bounding box. Though this will not guarantee that the proposed bounding boxes fit the observed point clouds tightly, it will encourage the proposed bounding boxes to be centered around

the mean of the data, as the highest-likelihood proposal bounding boxes will be centered around the mean of the observed data.

Let our generative bounding box mean parameters be given by $\theta = [x_0 \;\; y_0 \;\; z_0]$, where x_0, y_0 and z_0 are the center points of the bounding box, and our standard deviation parameter be $\sigma > 0$. Then, given an observed point cloud dataset composed of N independent and identically-distributed points (an admittedly simplifying assumption):

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(N)}\end{bmatrix}, \; \mathbf{x}^{(i)} \in [0,1]^3, \; \mathbf{x}^{(i)} \perp\!\!\!\perp \mathbf{x}^{(j)} \; \forall \; i, j \in \{1, \; \cdots \; , N\},$$

Then we can express the likelihood of this data as a multivariate Gaussian likelihood:

$$L(\mathbf{X}; \theta) = \prod_{i=1}^{N} \frac{\exp(-\frac{1}{2\sigma^2}(\mathbf{x}^{(i)} - \theta)^T(\mathbf{x}^{(i)} - \theta))}{\sqrt{(2\pi\sigma)^3}}$$

Expressing this as the log-likelihood:

$$l(\mathbf{X}; \theta) = \log L(\mathbf{X}; \theta) = \sum_{i=1}^{N} -\frac{1}{2\sigma^2}(\mathbf{x}^{(i)} - \theta)^T(\mathbf{x}^{(i)} - \theta) - \frac{3}{2}\log 2\pi\sigma \tag{1}$$

To find the maximum-likelihood parameters, we solve for the following:

$$\theta^* = \arg\max_{\theta \in \Theta} L(\mathbf{X}; \theta) = \arg\min_{\theta \in \Theta} l(\mathbf{X}; \theta) \tag{2}$$

These optimal parameters can be solved using first-order conditions on log-likelihood:

$$\frac{\partial l(\mathbf{X}; \theta)}{\partial \theta} = \frac{1}{\sigma^2} \sum_{i=1}^{N} ||(\mathbf{x}^{(i)} - \theta^*|| = 0 \implies \theta^* = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} \tag{3}$$

Therefore, we can use a custom proposal distribution using importance sampling that roughly centers bounding box proposals on the mean values of point clouds. We don't choose to center the bounding box on this point cloud perfectly, since the third-order skew matrix of the observed point cloud could be nonzero. Psuedocode for our generative algorithm can be found below.

Algorithm 1 Multivariate Gaussian Generative Model

$x_0, y_0, z_0 \leftarrow$ **Mean PC Proposal**(observations)
$\mu = [x_0 \;\; y_0 \;\; z_0]$
$\sigma \sim U(\sigma_{\text{min}}, \sigma_{\text{max}})$
$L \sim U(0, 1)$
$W \sim U(0, 1)$
$H \sim U(0, 1)$
for i in 1:N **do**
  $x[i] \sim \mathbb{N}(\mu, \sigma\mathbf{I_3})$
end for
return L, W, H

Our custom proposal aims to increase the sample efficiency of our proposal distributions by only proposing bounding boxes with their mean centered around the mean of the observed point cloud:

Algorithm 2 Mean PC Proposal

Require: Input dataset: $\mathbf{X} = \begin{bmatrix}\mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(N)}\end{bmatrix}$, Input standard deviation: $\sigma \leftarrow 0.02$
$\mu \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{x^{(i)}}$
$x_0, y_0, z_0 \leftarrow \mathbb{N}(\mu, \sigma\mathbf{I})$
return x_0, y_0, z_0

Where our $\sigma = 0.02$ is a hyperparameter that we found performed well, both qualitatively and quantitatively, in our experiments with synthetically-generated point clouds.

3.2 Chamfer Distance as a Proxy for Likelihood

Now that we have an approach for generating bounding box proposals that are loosely centered on the mean of the point cloud, we can focus on scoring the proposed bounding boxes using a Chamfer distance metric as a proxy for likelihood. With many traces, it becomes possible to find and score candidate bounding box proposals using this distance metric, and construct a distribution-like arrangement of proposals according to the proposed bounding box's overall goodness of fit.

Chamfer distance is a metric that can be used the minimum square euclidean distance between two point sets, such as point clouds. Given two independently and identically-distributed datasets $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(N_X)}\end{bmatrix}, \ \mathbf{x}^{(i)} \in [0,1]^3, \ \mathbf{x}^{(i)} \perp\!\!\!\perp \mathbf{x}^{(j)} \ \forall \ i,j \in \{1, \cdots, N_X\}$$

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}^{(1)} & \mathbf{y}^{(2)} & \cdots & \mathbf{y}^{(N_Y)}\end{bmatrix}, \ \mathbf{y}^{(i)} \in [0,1]^3, \ \mathbf{y}^{(i)} \perp\!\!\!\perp \mathbf{y}^{(j)} \ \forall \ i,j \in \{1, \cdots, N_Y\}$$

Then we can express the Chamfer distance between these two point sets as:

$$CD(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{N_X} \min_{j \in \{1,\cdots,N_Y\}} ||\mathbf{x^{(i)}} - \mathbf{y^{(j)}}||^2 + \sum_{j=1}^{N_Y} \min_{i \in \{1,\cdots,N_X\}} ||\mathbf{y^{(j)}} = \mathbf{x^{(i)}}||^2 \tag{4}$$

Chamfer distance can be visualized using the graphic below:

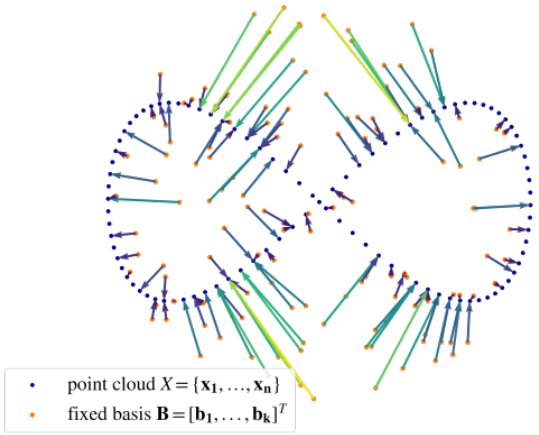

Figure 2: A graphical interpretation of how Chamfer distance is computed between two point sets. Note from the equation above and the graphic that this distance metric is symmetric. Source: [6].

Using this Chamfer distance, we can score our proposed bounding boxes according to the difference between the point-transformed bounding box (a bounding box represented as a sub-sampled point cloud along its edges) and the observed point cloud. Since we want to account for varying levels of fit for these bounding boxes, we ensure to sample many traces in the first stage, allowing us to find candidate bounding box proposals that fit our data well, but also other proposals that do not. Our algorithm for computing these Chamfer distances z can be found below:

Algorithm 3 Chamfer Distance Computation

Require: (1) Input dataset: $\mathbf{X} = \begin{bmatrix}\mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(N)}\end{bmatrix}$

Require: (2) Set of proposal bounding boxes $\{(x_0, y_0, z_0, L, W, H)\}_{t=1}^{T}$

$z = [0 \ \ 0 \ \ \cdots \ \ 0]$

for c in 1:C **do**

$\mathbf{B} \leftarrow$ **bbox_to_points**$((x_0, y_0, z_0, L, W, H)^{(c)})$

$[c] \leftarrow$ **Chamfer Distance**$(\mathbf{B}, \mathbf{X})$

end for

return z

Using these Chamfer distances as proxies for likelihood for our proposals, we are now ready to find optimal traces for our generative bounding box model.

3.3 Conditional Sampling for Proposal Refinement

With our Chamfer values computed for the proposed point clouds, we can now sample from our empirical distribution over Chamfer values to determine optimal bounding boxes over a wide range of "tightness-of-fit" values. To achieve this, we sort our Chamfer scores from least to greatest values, and then return the top five traces, which contain their associated bounding box dimensions and center points.

4 Implementation

To implement our generative models and importance sampling algorithm with a custom, data-driven proposal efficiently, we used Gen [3], a novel probabilistic programming language integrated into Julia [1]. Our implementation of our generative model, importance sampling algorithm, and constructing our posterior distribution over our bounding boxes was accomplished through the Gen API.

5 Experiment and Evaluation on Synthetically-Generated Point Clouds

5.1 Synthetically-Generated Point Clouds

to evaluate our generative model and inference framework, we apply our approach to synthetically-generated point clouds. These point clouds are synthetically generated by sampling from a bounding box of random size and dimension, and adding random Gaussian noise with standard deviation $\sigma = 0.02$. An example of a point-transformed bounding box, and its corresponding noisily-generated synthetic point cloud, can be seen below.

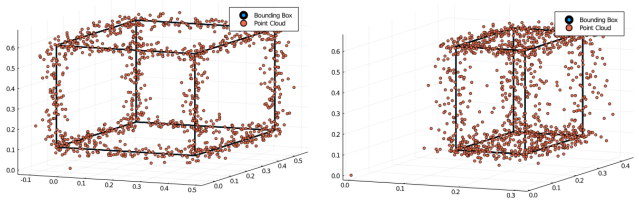

Figure 3: To generate our synthetic point clouds, we sampled from different segments corresponding to point-transformed bounding boxes of random offsets and dimension, and added random Gaussian noise to the sampled point values.

To evaluate our framework, we generated 10 synthetic point clouds, and for each of these, generated 1000 traces of proposed bounding boxes using importance sampling and our mean point cloud data-driven proposal. Each of our synthetically-generated point clouds had 481 points.

6 Results

Using the two approaches above, we found the following average performance amongst the top five traces for each of the experiments:

Experiment	Average Chamfer distance)	Fraction of Points Inside Bounding Box
Synthetic Data	496	0.713

Table 1: Experimental results for our unsupervised object detection model when evaluated on 10 synthetically-generated point clouds. For each of our synthetically-generated point clouds, we sampled 1000 proposed bounding boxes and chose the five bounding boxes with the smallest Chamfer distance.

Our results for our top five proposed bounding boxes can be seen below:

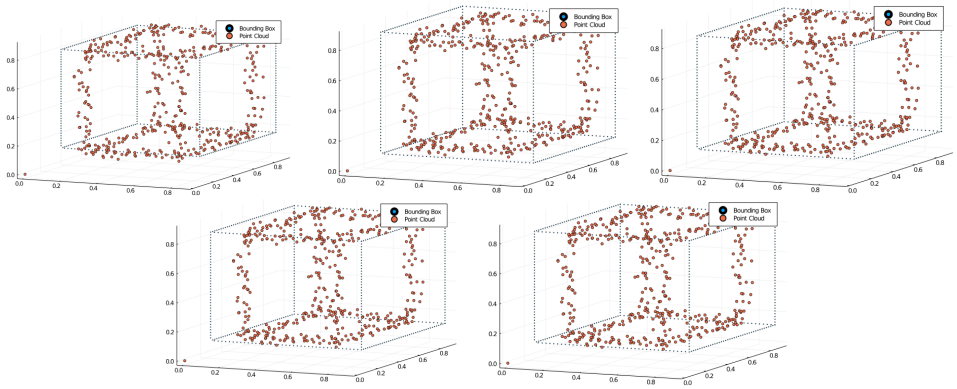

Figure 4: Our top five proposed bounding boxes for this given synthetically-generated point cloud. Notice how the top bounding boxes are all relatively similar to one another.

7 Future Work, and Conclusion

To further extend this model, we envision several avenues of future work. The first of these avenues is to account for higher-order statistical moments of our observed dataset inside of our proposal distribution. Although our proposal distribution accounted for the first-order moments of the observed point cloud distribution, enabling our sampled bounding boxes to be centered on our synthetic data, our input proposal distribution doesn't account for higher-order moments. Accounting for these higher-order moments could help improve sampling efficiency for generating proposed bounding boxes of near-optimal fit. The other avenue we see for extending this model is to leverage it for anchor weights initialization in neural-based object detection approaches, such as YOLO3D [8]. We believe this may help improve learning for 3D object detection approaches.

This work demonstrates the potential for unsupervised Bayesian object detection methods using generative models, data-driven proposals, and Chamfer distance as a proxy metric for the likelihood of an object bounding box. On our synthetic dataset, we found that on average we were able to sample bounding boxes that fit 71% of the synthetically-generated point cloud returns inside of them. This model certainly still has room to grow, and we believe that the integration of more advanced generative models, sampling approaches, and other computer vision metrics such as edge loss will help this generative model and inference framework continue to improve. This integration is made especially possible through the Gen API, which enables for a multitude of permutations between sampling algorithms and generative models [3]. We believe that this framework can serve as a proof of concept for future probabilistic programming-based unsupervised object detection techniques.

8 Acknowledgements

Thank you especially to Dr. Vikash Mansinghka and Alex Lew for all their helpful insights and advice this semester in 6.885, as well as for providing critical and meaningful instruction for how to use Gen.

References

[1] Jeff Bezanson et al. "Julia: A fresh approach to numerical computing". In: *SIAM review* 59.1 (2017), pp. 65–98. URL: https://doi.org/10.1137/141000671.

[2] Xiaozhi Chen et al. "3d object proposals for accurate object class detection". In: *Advances in Neural Information Processing Systems*. 2015, pp. 424–432.

[3] Marco F. Cusumano-Towner et al. "Gen: A General-purpose Probabilistic Programming System with Programmable Inference". In: *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI 2019. Phoenix, AZ, USA: ACM, 2019, pp. 221–236. ISBN: 978-1-4503-6712-7. DOI: 10.1145/3314221.3314642. URL: http://doi.acm.org/10.1145/3314221.3314642.

[4] Herman Kahn. "Use of different Monte Carlo sampling techniques". In: (1955).

[5] Lanlan Liu et al. *Generative Modeling for Small-Data Object Detection*. 2019. arXiv: 1910.07169 [cs.CV].

[6] Sergey Prokudin, Christoph Lassner, and Javier Romero. "Efficient learning on point clouds with basis point sets". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[7] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "Pointrcnn: 3d object proposal generation and detection from point cloud". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 770–779.

[8] Martin Simon et al. "Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[9] Walter Zimmer, Akshay Rangesh, and Mohan M. Trivedi. "3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams". In: *CoRR* abs/1905.00525 (2019). arXiv: 1905.00525. URL: http://arxiv.org/abs/1905.00525.