Progress Report

Bardia Mojra

February 5, 2021

Robotic Vision Lab

The University of Texas at Arlington

1 To Do

- Implement pose estimation: implement basic object detection and feature extraction using YCB dataset.
- Implement uncertainty.
- Implement DOPE with added dropout before each layer to estimate variational Bayesian inference.
- Read Data Association and Localization of Classified Objects in Visual SLAM [1]: in progress.
- Work on Pose Estimation Uncertainty paper.
- Implement PoseCNN, DOPE, and BayesOD.
- Look into domain randomization and adaptation techniques.
- Search for recent pose estimation survey papers: Found this paper, [2], on pose estimation.
- As I read more papers, go back and update Pose Estimation Survey paper.
- Finish Tensorflow tutorials, [3]: 4/9 chapters done.
- Learn to implement transfer learning.
- (On pause) Finish Docker tutorials, [4]: 4/18 chapters done.

2 Literature Reviews

2.1 Obtaining Well Calibrated Probabilities Using Bayesian Binning

- code: `https://github.com/pakdaman/calibration/`
- paper: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4410090/`
- citation: [5]

2.1.1 Introduction

In this paper, the authors propose a novel calibration method for probabilistic predictive models. Bayesian Binning into Qualities (BBQ), is a non-parametric and post-processing calibration method. Their proposed method is a binary classifier calibration method is based on the histogram-binning calibration method [6]. It is important to note this method could be extended to multi-class classification tasks [7].

2.1.2 Problem Statement

In machine learning, classification problems are often solved by deploying a a predictive model trained on some given data set but often underperform and make miscalibrated predictions ("classifier score"). Statistically speaking, for a calibrated prediction of i.e. 40%, there will be an occurrence of 40% for a give test set that is 1) large enough with respect to solution space size, 2) test data set is randomly selected (for more insight read on *Central Limit Theorem*.

Figure 1 is a reliability curve [8] [9] that is used as an example of a predictive model with poorly estimated probabilities.

2.1.3 Related Work

Mainly, calibration is done two ways 1) *ab initio*, by modifying objective function which increases computational cost. 2) It can be done as a post-processing procedure. Post processing can be categorized into parametric and non-parametric. Platt's method is an example of parametric calibration, [10]. Non-parametric methods include histogram- binning [6], Platt scaling [10], and isotonic regression [7].

2.1.4 Method

BBQ is an extension of simple histogram binning method [6], with added capability to consider different binning models (different number of bins) and their combinations under a Bayesian framework [11]. The generated Bayesian score provides further insight into network structure and it is also used when combining binning models.

$$\mathit{Score}(M) \;=\; P(M)\ .\ P(D|M)$$

The marginal likelihood, $P(D|M)$, has a closed form solution under the following 3 conditions, [11]:

1. All samples are under i.i.d. assumption and the class distribution $P(Z|B = b)$, which is class distribution for bin b, with a binomial distribution with parameter θ_b.
2. Bin distributions are independent.
3. The prior distribution over binning model parameters θ's are modeled using a Beta distribution.

Marginal likelihood in closed form, [11]:

$$P(D|M) = \prod_{b=1}^{B} \frac{\Gamma(\frac{N'}{B})}{\Gamma(N_b + \frac{N'}{B})} \frac{\Gamma(m_b + \alpha_b)}{\Gamma(\alpha_b)} \frac{\Gamma(n_b + \beta_b)}{\Gamma(\beta_b)}$$

Where:

- $\Gamma(n) = (n-1)!$
- N_b: The total number of training instances in the b'th bin.
- n_b: The total instances of class *zero* among all training instances N_b.
- m_b: The total instances of class *one* among all training instances N_b.
- $P(M)$: The prior distribution of binning model M, uniform distribution for initial condition.

The above equation is used for model averaging by BBQ, they point out that mentioned Bayesian scores could be used for model selection. Per [12], model averaging is superior to model selection methods.

2.1.5 BBQ

BBQ framework defines calibrated prediction as:

$$P(z = 1|y) = \sum_{i=1}^{T} \frac{Score(M_i)}{\sum_{j=1}^{T} Score(M_j)} \cdot P(z = 1|y, M_i)$$

Where:

- T : total number of binning models considered
- $P(z=1|y, M_i)$: probability estimate using model M_i for uncalibrated classifier output y.

2.1.6 Calibration Measures

- ECE: Expected Calibration Error is calculated over the bins.
- MCE: Maximum Calibration Error is calculated among the bins.

$$ECE = \sum_{i=1}^{K} P(i).|o_i - e_i| \quad , \quad MCE = \max_{i=1}^{K}(|o_i - e_i|),$$

Where:

- o_i: true fraction of positive instances in the ith bin.
- e_i: mean of the post-calibrated probabilities in the ith bin.
- $P(i)$: empirical probability (fraction) of all instances in the ith bin.

2.1.7 Empirical Results

- Acc: accuracy
- AUC: area under the ROC curve (receiver operator characteristic curve).
- RMSE
- ECE
- MCE

3 Progress

The following items are listed in the order of priority:

- Object Pose Estimation with Uncertainty: After talking to Chris, it seems like using Tensorflow Graphics module may not be the simplest route. He recommended using OpenCV and NumPy for import-export and processing, respectively. I have some familiarity with both modules and it seems straight forward.

  I recently learned a better way to keep track of my environment requirements. Python package manager (Pip) and Anaconda have built-in tools that if used properly could solve a large portion of my issues. There are features where once the environment is setup and operation, you can list and log all installed package with complete version number. This way I won't have to deal with Docker, for now at least.

- YCB Dataset [13]: I converted the provided download script from Python 2 to Python 3 and added it to the repository. I will use YCB as my main realistic dataset. I will have to some sort of semantic segmentation to isolate each object. For now, I can use pixel-wise Hough voting but later I will change to the method used by [14].

- Pose Estimation in Simulation: Chris recommended Nvidia Isaac SDK [15] and seems fairly realistic and is developed by Nvidia to tackle the *sim-to-reality* problem. Given my limited processing power and skillset, it seems to be a more feasible approach in comparison to using Unreal Engine.

- Implement features from PoseCNN, DOPE, and BayesOD.

- Look into domain randomization and adaptation techniques.

- Search for recent pose estimation survey papers: Found this paper, [2], on pose estimation.

4 Plans

The following items are listed in the order of priority:

- Use Nvidia Isaac SDK for pose estimation training.

- Keep working on Bayesian Pose Estimation paper.

- ARIAC: For now, I will focus on implementing pose estimation and BayesOD implementations.

- Continue on UE4 tutorials.
- Pose Estimation Survey Paper Feedback: On hold, I am working on Bayesian Pose Estimation.
- Project Alpe with Nolan: On pause for right now.
- UR5e: Finish ROS Industrial tutorials.

5 2021 Goals and Target Journals/Conferences

- Submit a paper on pose estimation with uncertainty to ICIRS.
- Get comfortable with TensorFlow and related Python modules.
- Keep writing.

References

[1] A. Iqbal and N. R. Gans, "Data association and localization of classified objects in visual slam," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 113–130, 2020.

[2] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," *Artificial Intelligence Review*, pp. 1–58, 2020.

[3] B. Planche and E. Andres, "Hands-on computer vision with tensorflow 2," 2019.

[4] G. Schenker, *Learn Docker – Fundamentals of Docker 19.x: Build, test, ship, and run containers with Docker and Kubernetes, 2nd Edition.* Packt Publishing, 2020.

[5] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[6] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *Icml*, vol. 1, pp. 609–616, Citeseer, 2001.

[7] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

[8] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, no. 1-2, pp. 12–22, 1983.

[9] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632, 2005.

[10] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[11] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[12] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, pp. 382–401, 1999.

[13] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*, pp. 510–517, IEEE, 2015.

[14] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] Nvidia, "Nvidia isaac sdk — nvidia developer." `https://developer.nvidia.com/Isaac-sdk`, 2021. (Accessed on 02/05/2021).