Ouroboros: A human-robot hybrid system for consuming tail cases

Daniel H. GROLLMAN Abhijit MAJUMDAR Halit Bener SUAY
Plus One Robotics, Inc. USA

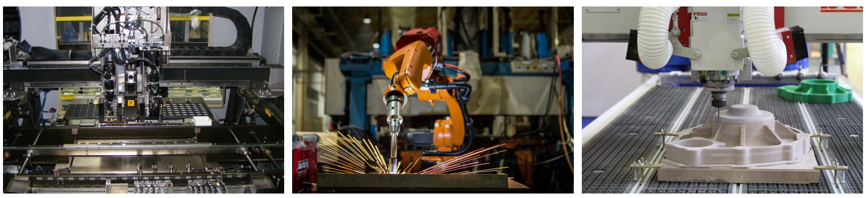
Abstract. We present a symbiotic hybrid intelligence system for automated pick and place in a warehouse logistics setting. This domain is characterized by the need for systems to deal with long-tailed distributions of scenarios, consisting of a wide variety of relatively rare edge cases. Handling of these cases has great impact on the reliability, robustness, and commercial viability of these systems. Rather than attempt to learn a sufficiently generalized method to handle any potential unknown situation, our system focuses learning on the most common scenarios and leverages human flexibility to generate responses to the rarer ones. As more tail cases and their human-generated responses are seen, they are learned, moved into the body, and new tail cases take their place.

Keywords. Human-Robot Systems. Incremental Learning. Heavy tailed distribution.

Introduction

For a long time, successful automation has depended on the existence of highly controlled and precisely repeatable scenarios. Automated machines could accurately execute hard-coded programs to perform a variety of tasks such as those in Figure 1, assuming that their was no uncertainty in their state space. Any variation in the environment had to be explicitly controlled for, or else the machine might go through the motions, without actually performing the specified work.

With the advent of more advanced control paradigms and novel sensors, this situation improved. Automated machines transformed into reactive robots, gaining the capability to sense and respond to changes in their environment. Tolerances have been relaxed, with systems able to handle greater, potentially uncon-



(a) Pick and Place (b) Welding (c) Machining

Figure 1. Examples of automated machines that require no uncertainty in their state space

(a) Household vacuuming (b) Hospital logistics (c) Aerial surveying

Figure 2. Examples of robots that can sense and respond to changes in their environments

trolled (but bounded) variations in their inputs, while still performing the desired work to a high degree of quality, as in Figure 2.

Machine learning (ML) has been yet another nail in the coffin of the old repetitive paradigm. Rather than requiring users to manually specify the acceptable variance, ML techniques enable the system to both learn the relevant features of and extract a model for its operating parameters from examples of the situations it will be faced with. However, ML-based approaches are not without their limitations and caveats. They are still GIGO (Garbage In - Garbage Out) systems, requiring much care and attention as to what they are trained on, and can suffer from issues of poor generalization (particularly extrapolation beyond their training data), inscrutability, and unexpected behaviors [1].

In this paper we show how humans (who are, generally, understandable, good at extrapolation, and behave as expected [at least in our domain]) can counteract these failings of ML and create a hybrid system to continually adapt to new and rare scenarios as they arrive.

1. Background

Our work applies to a subclass of warehouse automation problems. For our problem setting we use a serial manipulator for robotic pick and place applications and deep learning for perception and machine vision [2].

Robotic pick and place applications are commonly seen in many industrial settings such as assembly, CNC machining, and material handling (Fig.1). Recent developments in sensing, machine vision, motion planning and optimal control have enabled a diverse set of pick and place applications such as manipulation of deformable objects [3], and academic competitions such as the Amazon Picking Challenge [4]. For our problem domain, we define the pick and place problem as moving packaged commercial items (e.g. most items you can see on shelves at a grocery or a hardware store) and parcels (e.g. plastic and paper envelopes, boxes and bubble-wrapped envelopes) with an industrial robot from a “pick” pose to a “place” pose. Designing an object-agnostic ”pick” pose generator requires solving several problems such as object detection, classification, and pick-pose candidate evaluation; there are various approaches for each set of problems [5] however for the purposes of this paper we treat the object-agnostic “pick” pose generator as a monolithic component that leverages modern deep learning methods.

Generating a viable pick pose from 2D and 3D perception data has attracted a remarkable amount of effort in the research community recently [6,7,8]. Due to the high variability of items we aim to handle for our problem setting, we assume that it is not possible to have a fully defined set of items at the time of training a pick pose generator but rather the set of items is always variable. Recent work [9,10] has shown promise in using deep learning to generate machine learning models that can predict object boundaries and classes after being trained with annotated datasets. Approaches that are shown to work with promising accuracy employ this standard process to generate predictive models: 1. Collect data representative of the scenarios the system will have to handle 2. Annotate the data to establish ground truth 3. Train a prediction model using the annotated data 4. Deploy the model and monitor its performance in a test/real-world setting.

This standard process assumes that the data collected for training represents the data that will be encountered in testing. Issues can arise then when this assumption is violated such as when distributions of the test and training datasets differ (covariate shift [11]), or if the training samples do not accurately capture the entire distribution, which can occur if the true distribution is heavy-tailed [12]. Additional steps are necessary to address these scenarios.

A naïve approach is to continue the data collection, annotation, and training cycle that was started pre-deployment, but doing so can come with a large overhead cost. An alternate tactic is to use the running system to generate its own data for continued training, for example through Reinforcement Learning [13].

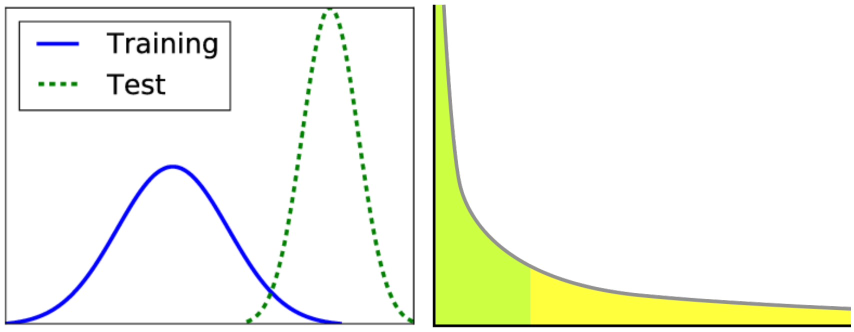
In contrast with previous work, our domain imposes physical constraints where every deployed system instance can have a different physical layout, which limits what can be done in simulation or via data augmentation for continued training[14]. Thus, our approach differs from previous work in that we employ deployed systems in the continued training process itself.

To address the unique set of needs of our problem domain, we present a human-in-the-loop machine learning framework Ouroboros, that approaches the problems of covariate shift and heavy tails by bootstrapping the standard process of training machine learning models using data from test cases.

2. Distributional View

At the root of our problem are the intertwined problems of data drift and long-tailed distributions, illustrated in Figure 3. Data drift (or covariate shift) refers to the scenario where the distribution of inputs presented to the system shifts over time, resulting in a different distribution being used at test time (or in production) than was used for training [15]. As an example, consider the scenario where a vision system is trained only on outdoor images taken at 9am, but then deployed and used throughout the whole day. As the sun and shadows move, the input data will 'drift' away from that which the model learned on.

Tail cases, on the other hand, are scenarios that are drawn from the same distribution as the training data, but are so different and rare as to either not appear in the training data sample, or be ignored by the training system [12]. Due to their rarity, a learned system (even a well regularized and generalized one), may behave unexpectedly when encountering a tail case. We note that as deployed production systems run longer in continuous domains, the probability of encountering a tail case approaches one.



(a) Data drift (b) Long-tailed distribution

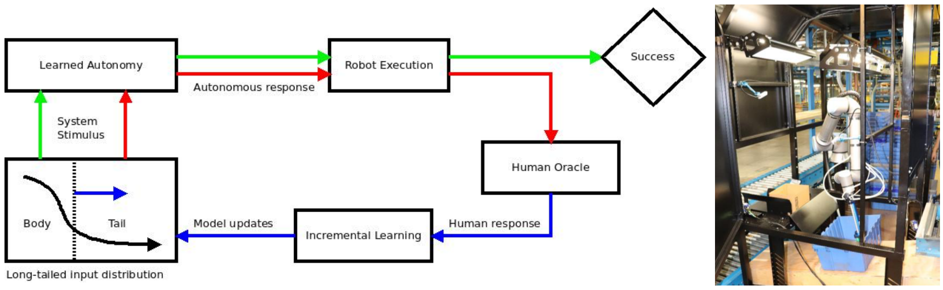
Figure 3. The two problems we seek to address. Graphs show the probability (vertical) of particular scenarios (horizontal) being encountered during operation. In data drift, the distribution changes between training and deployment, and the system may be asked to respond to scenarios it has never seen or anticipated. Similarly, in long-tailed distributions, there are many rare cases that are either not seen, or extremely outnumbered, during training.

These two problems can compound each other. Namely, as a system runs longer, more and more tail cases occur, in effect shifting the distribution of inputs, and eventually leading to covariate shift and degrading system performance. To combat this behavior, many systems incorporate ideas from Out-Of-Distribution detection [16] to detect when input data falls outside of the training distribution, and either reject the input and resort to a default policy, or trigger model updates

3. Ouroboros

Our approach, illustrated in Figure 4a, focuses on handling tail cases, and we view data drift as an issue of tail accumulation (In the extreme, a total shift of input distribution looks like a flood of tail cases). Our autonomous system handles the bulk (body) of scenarios and we use local self-feedback from the robot to detect when our system is not performing well (out of distribution). Tail cases are handled by a human-in-the-loop oracle who provides immediate corrective feedback, which doubles as the ground-truth annotation required for learning. As tail cases repeat, iterative learning incorporates them into the body of expertise and the distribution of able-to-be-handled-autonomously scenarios, so the human's input is no longer needed for them. Instead they will focus on new, even rarer, tail cases as they arise.

This shift in distribution is illustrated in Figure 4a (lower left). In the extreme, our approach can start with an untrained autonomous system that thinks everything is a tail case and requires the human to respond to every input. As time progresses, the most common scenarios will be learned, while the rarer ones will still require the human to be involved. Eventually, we believe the system will settle into a steady-state, where the human's input is required only sporadically, at a rate determined by the true underlying distribution of inputs.



(a) Ouroboros

(b) Test System

Figure 4. Left: An illustration of our system. Samples from the long-tailed input distribution are first processed by the learned autonomy, generating an autonomous response which is executed by the robot. In the case of success, the process stops, but if a failure is encountered the stimulus and autonomous response are passed to a human oracle. The human provides an appropriate response, which is used to make incremental updates to the learned model. Samples from the learned 'body' of the distribution tend to follow the success pathway (green), while those from the unlearned 'tail' tend to follow the failure path (red). Continued learning results in the shift of the dividing line between body and tail (blue). Right: Our test system, consisting of a 6DOF arm and a RGBD camera. The task is to pick items, one at a time, from the blue tote and place them into the cardboard box as quickly as possible.

3.1. Case Study

We implemented our system on the production system shown in Figure 4b. Starting with a traditionally trained vision model the system ran for 15 weeks, during which 23K out of 36k inputs were deemed to be tail cases (64%, speaking to the heaviness of the tail of the input distribution) and were sent for human handling. Analysis of the model performance on the tail case data is in Table 1, and shows that the tail data is sufficiently different from the original training data to have degraded performance. After re-training using the new data, our updated model achieves similar performance on both the old and new data, illustrating that the tail has become part of the body.

ModelID	Body Performance	Tail Performance
Original	0.9523	0.5793
Retrained	0.9454	0.8153

Table 1. Average Precision scores before and after tail incorporation

4. Conclusion and Vision of the Future

We have presented a symbiotic hybrid intelligence system for automated pick and place in a warehouse logistics setting. Such tasks are the epitome of the dirty, dull, and dangerous jobs that we want robots to take over from humans, but there are growing concerns of increased automation inducing an employment shortage, despite the current inability of employers to staff all of their shifts. Additionally, it is our belief that any attempt at a pure AI solution to this problem using current technologies will fail, due to the ever-shifting nature of the items handled and the infinite variety of possible situations the system has to handle. Instead, by leveraging the skills of both humans (adaptability) and robots (repeatability) we can enable robots and humans to work together, each doing what they do best.

Authors

Daniel H. Grollman, Ph.D - Principal Engineer focusing on Machine Learning applied to Human-Robot Interaction for logistics and companion robots

Abhijit Majumdar - Senior Developer in Artificial Intelligence focusing on Computer Vision in warehouse environments

Halit Bener Suay, Ph.D - Senior R&D Developer in Artificial Intelligence and Vision focusing on Systems for scalable learning

References

[1] Power A, Burda Y, Edwards H, Babuschkin I, Misra V. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets. CoRR. 2022;abs/2201.02177.

[2] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[3] Lee AX, Lu H, Gupta A, Levine S, Abbeel P. Learning force-based manipulation of deformable objects from multiple demonstrations. In: ICRA. IEEE; 2015. p. 177-84.

[4] Eppner C, Höfer S, Jonschkowski R, Martín-Martín R, Sieverling A, Wall V, et al. Lessons from the Amazon Picking Challenge: Four Aspects of Building Robotic Systems. In: IJCAI. AAAI Press; 2017. p. 4831–4835.

[5] Prince SJD. Computer Vision: Models Learning and Inference. Cambridge University Press; 2012.

[6] Mahler J, Liang J, Niyaz S, Laskey M, Doan R, Liu X, et al.. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. arXiv; 2017.

[7] Mahler J, Matl M, Liu X, Li A, Gealy D, Goldberg K. Dex-Net 3.0: Computing Robust Robot Vacuum Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning. arXiv; 2017.

[8] Mahler J, Matl M, Satish V, Danielczuk M, DeRose B, McKinley S, et al. Learning ambidextrous robot grasping policies. Science Robotics. 2019;4(26).

[9] Redmon J, Divvala SK, Girshick RB, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. CoRR. 2015;abs/1506.02640.

[10] Cheng B, Collins MD, Zhu Y, Liu T, Huang TS, Adam H, et al. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. CoRR. 2019;abs/1911.10194. Available from: `http://arxiv.org/abs/1911.10194`.

[11] Shimodaira H. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference. 2000;90(2):227-44.

[12] Wang YX, Ramanan D, Hebert M. Learning to Model the Tail. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30; 2017. .

[13] Wu Y, Yan W, Kurutach T, Pinto L, Abbeel P. Learning to Manipulate Deformable Objects without Demonstrations. CoRR. 2019;abs/1910.13439.

[14] Ghiasi G, Cui Y, Srinivas A, Qian R, Lin TY, Cubuk ED, et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. arXiv. 2021 Jun.

[15] Ren AZ, Majumdar A. Distributionally Robust Policy Learning via Adversarial Environment Generation. arXiv:210706353 [cs]. 2021 Dec. ArXiv: 2107.06353.

[16] Sun L, Jia X, Dragan A. On complementing end-to-end human behavior predictors with planning. In: Robotics: Science and Systems XVII; 2021. .