Autonomous Science Platforms and Question-Asking Machines

Kevin H. Knuth  
Departments of Physics and Informatics  
University at Albany (SUNY)  
Albany, NY, USA  
Email: knuth@albany.edu

Julian L. Center  
 Autonomous Exploration, Inc.  
Andover, MA, USA  
Email: jcenter@ieee.org

Abstract—As we become increasingly reliant on remote science platforms, the ability to autonomously and intelligently perform data collection becomes critical. In this paper we view these platforms as question-asking machines and introduce a paradigm based on the scientific method, which couples the processes of inference and inquiry to form a model-based learning cycle. Unlike modern autonomous instrumentation, the system is not programmed to collect data directly, but instead, is programmed to learn based on a set of models. Computationally, this learning cycle is implemented in software consisting of a Bayesian probability-based inference engine coupled to an entropy-based inquiry engine. Operationally, a given experiment is viewed as a question, whose relevance is computed using the inquiry calculus, which is a natural order-theoretic generalization of information theory. In simple cases, the relevance is proportional to the entropy. This data is then analyzed by the inference engine, which updates the state of knowledge of the instrument. This new state of knowledge is then used as a basis for future inquiry as the system continues to learn. This paper will introduce the learning methodology, describe its implementation in software, and demonstrate the process with a robotic explorer that autonomously and intelligently performs data collection to solve a search-and-characterize problem.

I. INTRODUCTION

Remote science platforms are becoming increasingly important in fields where data collection in hostile or remote environments is necessary. Such areas of study include, but are not limited to, exploration of the depths of Earth’s oceans, monitoring Earth’s climate from orbit, and studying off-world environments such as the surface of Mars. Autonomy in such systems is critical as it enables remote instrumentation to operate without constant expert supervision and intervention. The application of intelligent autonomous data collection need not be limited to scientific applications, but is also applicable to problems such as surveillance and monitoring. Nor does application need to be limited to remote platforms. For instance, one could consider applications such as the exploration of massive databases. Regardless of the specific problem, it is clear that the need for intelligent autonomy will increase significantly as we expand our reach.

Autonomous intelligent data collection consists of asking questions of a system and it is in this sense that one can view remote science platforms as question-asking machines. Such a question, which we will refer to generically as a query, could consist of simply sampling from a sensor, designing and performing an experiment, performing a database query, or even posing a question to a human. The challenge consists of selecting the most relevant query to perform, and avoiding irrelevant queries. By selecting highly relevant queries, one naturally ensures that the data is of the highest relevance. Compared to current remote science platforms, this approach has the benefit of reducing the total amount of data needed to make high quality inferences, which can further reduce the amount of required storage or the necessary transmission bandwidth in remote operations. This has the potential to reduce significantly the amount data recorded in a particular mission thus avoiding situations like the data firehouse and the massive database littered with useless information.

Since the inception of information theory [24], it has been clear that the Shannon entropy can be used to quantify the number of binary questions necessary to characterize a signal. More generally, entropy has been used to quantify uncertainty, which we demonstrate is related to question-asking. Many researchers have worked on the problem of designing automated machines as well as automating the processes of experimental design. Relevant to the present approach are the concepts of cybernetics [27] and experimental design [5], [17], [19], which have been pursued in various forms by a good number of researchers. Of particular note is the earlier work on active data selection approach of MacKay [20], maximum entropy sampling and Bayesian experimental design by Sebastiani and Wynn [22], [23], cybernetics by Fry [7], and Bayesian adaptive exploration by Loredo [18].

The present approach is based on the firm foundation of the scientific method, which couples the processes of hypothesis generation, inquiry and inference. Such a foundation is model-based in that the space of potential hypotheses is described by one or more models each comprised of a set of model parameters, which must be estimated. The specification of the set of potential hypotheses and the set of potential queries defines the learning problem.

Specifically, this paper describes a learning system based on two computational engines: an inference engine and an inquiry engine. The inference engine implements the learning process via Bayesian posterior sampling; whereas the inquiry engine quantifies the possible queries by computing their relevance to the estimation problem defined by the model. While much
of our ongoing research is focused on developing the inquiry calculus, which enables one to directly compute the relevance of a question, here we will discuss intelligent inquiry in terms of Bayesian adaptive exploration [18].

The paper is organized as follows. Section II describes the mathematical foundation underlying expected information gain and Bayesian adaptive exploration. Section III describes our implementation of the inference and inquiry engines and the interface with the robotic platform. Section IV describes the results of an experiment that demonstrates automated intelligent data collection.

II. QUESTIONS AND EXPECTED INFORMATION GAIN

We have been approaching the problem of experimental design from a very different direction [9], [13], [15]. Having derived the algebra of questions [8], we continue to work to generalize this algebra to an inquiry calculus that enables one to compute the relevance of a question. Those interested in this ongoing foundational work are encouraged to consult the following papers [8]–[12], [14], as well as the paper by Richard T. Cox, which inspired this approach [2]. We recommend the text by Davey and Priestley [3] as an accessible introduction to order theory, and the text by Sivia and Skilling [25] as an accessible introduction to Bayesian inferential methods.

The approach we have utilized so far can also be viewed from the perspective of Bayesian adaptive exploration [18]. Consider a proposed experiment \( E \), which corresponds to taking a measurement at position \((x_e, y_e)\). We do not know for certain what we will measure. The best we can do is to make a prediction based on the current posterior probability of the model \( M \) possibly obtained from previously recorded data \( D \). To make such a prediction, we write the probability of a measurement \( D_e \) by marginalizing the joint probability of \( D_e \) and \( M \)

\[
p(D_e|D, (x_e, y_e)) = \int dM \ p(D_e, M|D, (x_e, y_e)). \tag{1}
\]

Using the product rule, we can write

\[
p(D_e|D, (x_e, y_e)) = \int dM \ p(D_e|M, D, (x_e, y_e)) \ p(M|D, (x_e, y_e)). \tag{2}
\]

The first term in the integral can be simplified by observing that the data \( D \) are not necessary if we know the precise values of the model parameters \( M \)

\[
p(D_e|D, (x_e, y_e)) = \int dM \ p(D_e|M, (x_e, y_e)) \ p(M|(x_e, y_e)). \tag{3}
\]

At some point we must make a decision as to where to place our sensor. To do this, we must introduce a utility function \( U(\text{outcome}, \text{action}) \) and select the sensor location that maximizes its expectation value

\[
(\hat{x}_e, \hat{y}_e) = \int dD_e \ p(D_e|D, (x_e, y_e)) \ U(D_e, (x_e, y_e)) \tag{4}
\]

where the measurement location \((x_e, y_e)\) represents the action and the measurement \( D_e \) is the predicted outcome. To select a measurement that provides the greatest expected gain in information, we use a utility function based on the information provided by the measurement. Using the Shannon information for our utility function we find

\[
U(D_e, (x_e, y_e)) = \int dM \ p(M|D_e, D, (x_e, y_e)) \log p(M|D_e, D, (x_e, y_e)). \tag{5}
\]

By writing the joint entropy for \( M \) and \( D_e \), and writing the integral two ways, one can show [18] that the optimal experiment can be found by maximizing the entropy of the possible measurements

\[
U(\hat{x}_e, \hat{y}_e) = \arg \max_{(x_e, y_e)} - \left( \int dD_e \ p(D_e|D, (x_e, y_e)) \times \int dM \ p(M|D_e, D, (x_e, y_e)) \log p(M|D_e, D, (x_e, y_e)) \right). \tag{6}
\]

Other utility functions could be used that depend on energy, time, etc. These considerations will be important in a fully-functioning automated instrument.

III. DEMONSTRATION

We demonstrate this methodology by programming a robotic arm to locate and characterize a white circle on a black field using a simple light sensor. Note that the robot is not programmed with any search strategies whatsoever. Instead, the learning problem is defined by providing the robot with a mathematical model of a circle as well as a mathematical model of its light sensor response [21]. With the inference and inquiry engines, the system is able to select sensor position locations autonomously and intelligently enabling it to efficiently collect relevant data from which the circle parameters can be estimated.

A. Robotic Implementation

The robotic arm was constructed using the LEGO Mindstorms System. It has its own local computer called the LEGO brick (developed by MIT Media Labs), which controls the basic movements of the robot and performs the actual data acquisition. The robot communicates with a Dell Latitude PC Laptop via a Bluetooth connection. The inference and inquiry software are coded in Matlab and run on the PC. The PC sends sensor placement instructions to the robot via Bluetooth followed by an execute measurement command, and waits for the results from the robot. The robot follows the motor instructions dictated by the PC and records a light intensity using the light sensor at the end of the robotic arm. This information is sent back to the PC for further analysis and instruction by the inference and inquiry engines.

The robot is placed on a black field with dimensions approximately 1.0 m by 0.5 m (120 × 60 LEGO units) where there is also a white circle of unknown dimensions and unknown position (Figure 1). The goal is for the robot to find the both the center coordinates and the radius of the circle to within
Fig. 1. Robotic Arm and Autonomous Sensor Placement Task. The goal of the task is for the robot to characterize the position and radius of the circle using a simple light intensity sensor that can sample only one point at a time. The inference and inquiry software engines run on the PC, which uses Bluetooth to communicate with the data acquisition software on the robot.

4 mm (0.5 LEGO units) by using its light intensity sensor. For this task to be completed in a reasonable amount of time, the robot must sample adaptively and intelligently.

B. Inference and Inquiry Engines

Using a Bayesian posterior sampling algorithm, the inference engine samples 50 circles from the posterior probability (see blue circles in Figure 2). Each of these circles represents a probable solution. To compute the entropy of a given sensor site, one simply queries each circle in the sample for a prediction as to what intensity will be measured at that site. The entropy of this distribution of 50 predicted measurements as a function of sensor position is computed and illustrated graphically as a copper shading of the playing field. Measurement positions with low relevance (low entropy) are dark; whereas highly relevant measurement positions are light. In the figure, the white squares indicate previous light intensity measurements that resulted in a high intensity (probably on the circle), and the black squares represent measurements resulting in a low intensity. The three white squares are most likely on the circle, and this relevance-based technique indicates that it is not worthwhile to take a measurement within that region, as it is expected to be white. This all comes naturally from the simple entropy calculation. There is no problem-specific hard coding other than the model description for the Bayesian inference engine and the sensor characteristics. The next measurement (green square) has been selected to be the position with the highest entropy. This measurement will naturally rule out one half of the blue hypothesized circles at each step leading to an optimal search. In practice, once the robot finds a position that is on the circle, it can estimate its coordinates and radius to the desired accuracy in approximately 25 measurements. This is much more efficient than scanning the whole region with a point sensor.

To characterize a circle, we require a model consisting of three parameters: the center location \((x_o, y_o)\) and radius \(r_o\). We will refer to the set of model parameters as

\[
M = \{(x_o, y_o), r_o\}. \tag{8}
\]

This model, in conjunction with the sensor model (implemented in the likelihood function) defines the learning problem.

The inference engine implements Bayesian learning. It is provided with the model of the circle as well as a prior probability on the circle parameters. The prior probability is uniform in each of the model parameters and incorporates cut-offs defined by the range of the playing field as well as a minimum and maximum radius for the circle. The inference engine is built from a Monte Carlo algorithm called nested sampling [25], which provides the evidence of the model for use in model testing. Inference proceeds by taking the recorded data and evolving a set of 50 circles sampled from the prior probability. At the conclusion, nested sampling provides the evidence for the model as well as a set of samples. Often we require more unique samples than are returned by the inference engine, and this is resolved by applying the algorithm recursively to the remaining confident samples.

Fig. 2. The Playing Field with the Inference and Inquiry Status Overlaid. The coordinate axes are in the robot’s native units. The semi-annulus reflects the range of reach of the robotic arm. White and black squares indicate locations of previously recorded samples. Their white and black coloration reflects relatively light and dark measurement results, respectively. The blue circles represent the set of circles sampled from the posterior probability, and the red circle indicates the mean. The copper-toned shading denotes the entropy values for each of the possible measurement locations. Note that the algorithm finds that measurements in regions where the circle has been ruled out are less relevant. In addition, measurements in regions where the circle probably resides are also less relevant. The green square in the high entropy region of the map indicates the measurement location chosen for the next iteration. Notice that due to the nature of entropy, this measurement stands to rule out one-half of the potential circle positions and sizes. This effectively results in an efficient binary search.
nested sampling algorithm. In these cases, we further diversify the samples by employing Metropolis-Hastings Markov chain Monte Carlo (MCMC).

At each step, the software displays the current state of both the inference and inquiry procedures. These sampled circles, which are displayed as the blue circles in Figures 2 and 3, indicate probable circle locations and dimensions given the previously recorded light intensity levels. These samples are passed directly to the inquiry engine as a representation of the current posterior probability.

In this particular problem, the most relevant measurement location is computed using the entropy. The inquiry engine cycles through a grid of discrete measurement location candidates. For a given location candidate, each of the 50 sampled circles is queried as to what light intensity value is predicted to be observed. If the considered measurement location is positioned within the sampled circle, a measurement value is sampled from a Gaussian distribution with a mean value equal to the measurement intensity expected from a white background and an experimentally estimated standard deviation. If the measurement location lies outside of the sampled circle, the inquiry engine samples from a Gaussian distribution with a dark mean value and standard deviation. In a more advanced version of the algorithm, we employ a model of the spatial sensitivity function of the light sensor [21] and integrate over the surface intensity predicted by the model. The set of predicted measurement results are binned and the entropy is computed. The measurement location with the greatest entropy is selected as the next sensor position.

In this demonstration, the robot performs an exhaustive search. This enables us to generate an entropy map, which can be seen in Figures 2 and 3 as the shaded copper background. Lighter background values reflect higher entropy indicating highly relevant measurement locations. Darker background values reflect lower entropy indicating less relevant measurement locations. The white and black squares indicate past light and dark measurement results. The green square situated firmly in the high entropy region denotes the next measurement location.

C. Demonstration Results

The robot is programmed to estimate the three circle parameters. It has been pre-programmed with a prior probability that encodes the information that the circle has a radius that is bounded by a minimum of 3 units and maximum value of 65 units. Having not yet collected data, the sampled circles are essentially uniformly distributed according to the prior, and the entropy map is more-or-less uniform indicating that all sensor locations are equally relevant (Figure 3, Top).

In this example, the first selected sensor location happened to be over a dark region of the field, which resulted in a dark measurement. The resulting posterior probability is then biased by this data via the likelihood, indicating that the circle is probably not near that sensor location. The entropy map is naturally favors the more distant unsearched areas of the playing field. In addition, sensor locations near to the sampled point are less relevance than those far away (Figure 3, Center).

Once a white point on the circle has been identified, the posterior probability becomes more focused as indicated by the samples. Note that sensor locations near the point identified to be on the circle are less relevant as they too have a high probably of also being located on the circle (Figure 3, Bottom). The entropy map indicates that the most relevant measurements will serve to rule out approximately one half of the circles in each step resulting in an efficient binary search. This can be more clearly seen during the advanced stages of the experiment (Figure 2). On average we have found the robot to take approximately 25 measurements to get to the desired
accuracy of 0.5 distance units. This corresponds to locating the center of the circle to within an area of 0.25 units\(^2\) out of a total area of almost 4000 units\(^2\).

To compare the robot’s performance to the results expected from a binary search, we must consider the total volume of the parameter space for all three model parameters. The prior probability encoded the radii of the circles to be within 65 units \(− 3 \text{ units} = 62\) units. This gives us a total volume of 62 units \(× 4000\) units\(^3\) \(≈ 250,000\) units\(^3\) in which we locate a volume of 0.125 units\(^3\). The contraction is on the order of \(2 \times 10^6\), which corresponds to 21 binary questions. This illustrates that by finding the solution to the desired accuracy with an average of 25 samples, the robot is nearly optimal.

IV. CONCLUSION

This paper describes a practical methodology for designing autonomous intelligent data collection systems based on an implementation of the scientific method using software engines that perform both inference and inquiry. Such a system is not explicitly programmed with search strategies, but instead its actions are constrained by both the specification of what is to be learned and the potential queries that can be performed. This is handled by specifying the potential models and their defining parameter values, as well as the statistical properties of the response to each potential query.

While our research focuses on remote science platforms, the applications that could benefit from this computational technology are numerous and varied. We have already applied this methodology to the problem of experimental design in an electron paramagnetic resonance (EPR) study [4], which is neither autonomous or remote. While dramatic futuristic applications of this computational technology include exploration of the subsurface oceans of Jupiter’s moons, these methods also promise help people here on Earth by enabling robotic search and rescue missions in hazardous or confined environments, such as in the aftermath of an earthquake or biological or nuclear attack, or simply by enabling robots to intelligently navigate their daily duties in the home or workplace.

While these entropy- or mutual information-based methods have been used increasingly in the last few years to enable remote science platforms (eg. [16], [26]), our ultimate goal is to continue to develop the mathematical foundation underlying these solutions and to compute the appropriate form for the relevance automatically given the problem description. This introduces the potential to develop automated systems that generate code for intelligent automated data collection systems, much like AutoBayes generates code for data analysis [6]. This will result in intelligent systems that need not be programmed how to think in particular situations, but can perform inference and inquiry automatically given only the model or set of models to be learned.

ACKNOWLEDGMENT

This work has been supported in part by the College of Arts and Sciences and the College of Computing and Information of the University at Albany (SUNY), 2008 University at Albany Faculty Research Award, the NASA Applied Information Systems Research Program (NASA NNG06GH17G) and the NASA Applied Information Systems Technology Program (NASA NNX07AD97A). In addition, K.H. Knuth would like to thank Philip Erner, Scott Frasso, Nabin Malakar, and Rotem Guttman for their assistance with various aspects of software and hardware during the development of this work.

REFERENCES


