Nonparametric Distribution Regression Applied to Sensor Modeling

Abhijeet Tallavajhula[1], Barnabás Póczos[2], Alonzo Kelly[1]

***Abstract*— Sensor models, which specify the distribution of sensor observations, are a widely used and integral part of robotics algorithms. Observation distributions are commonly approximated by parametric models, which are limited in their expressiveness, and may require careful design to suit an application. In this paper, we propose nonparametric distribution regression as a procedure to model sensors. It is a data-driven procedure to predict distributions that makes few assumptions. We apply the procedure to model raw distributions from real sensors, and also demonstrate its utility to a mobile robot state estimation task. We show that nonparametric distribution regression adapts to characteristics in the training data, leading to realistic predictions. The same procedure competes favorably with baseline parametric models across applications. The results also help develop intuition for different sensor modeling situations. Our procedure is useful when distributions are inherently noisy, and sufficient data is available.**

I. Introduction

The sensors we work with in robotics are often noisy, and operate in varied environments. Sensor models, which model the distribution of observations given state, help deal with these challenges, and are ubiquitous in robotics. To note two examples, they are a component of state estimation algorithms [1], used in updating state based on sensor readings. Observation distributions are also the workhorse of simulation [2], used to generate samples. There are two problems to be considered. The first is the problem of modeling distributions, given samples. There is a long history of parametric modeling, which approximates the observation distributions with a parametric one. While these methods may be sufficient when the approximations are good, they are limited in their expressiveness. Dealing with complex distributions involves incorporating more parameters, and increasingly elaborate models. For example, [3] presents results of detailed parametric modeling for sonar sensors. The second problem is to predict observation distributions. Data, such as logged states and observations, will be available only at certain states, from which we may want to predict a distribution at an unseen state. With the parametric models, the approach is to allow the parameters to depend on state. Observation models for range sensors, for example, vary in complexity from the simple raycast model in [1], to a Bayesian network model developed in [4]. Sensors can behave noisily in commonplace situations, such as a compass on a mobile robot whose readings are affected by motor current. Instead of a search for the right parametric family for each new situation encountered, a principled and simple nonparametric approach to modeling and predicting distributions from data can be of wide benefit to robotics.

In this paper, we take a different approach to this classic problem. There has been recent work in machine learning which extends the standard case of real-to-real regression to distribution-to-real regression [5], distribution-to-distribution regression [6], and other variants. Importantly, these procedures are nonparametric. The case that is relevant to sensor modeling is real-to-distribution regression, or regressing from state to observation distributions. We will henceforth refer to real-to-distribution regression as simply distribution regression. Nonparametric distribution regression is conceptually simple, consisting of two steps that can be abstracted as density estimation, followed by regression. Nonparametric density estimation allows us to deal with complex distributions. Nonparametric regression allows us to deal with complex functions of state.

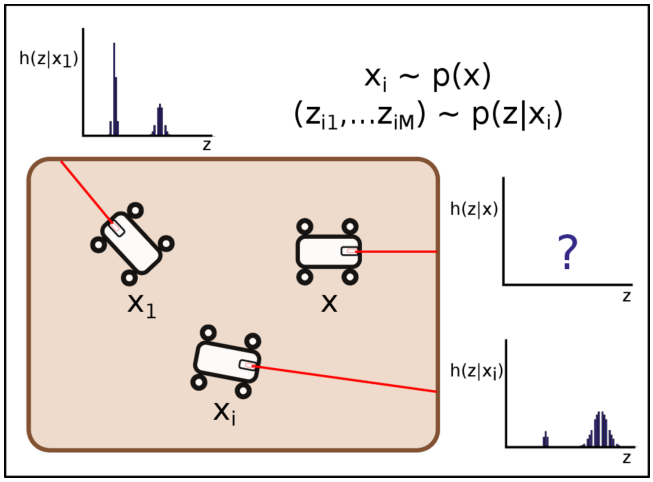


Fig. 1: The problem of sensor modeling. Given data in the form of states and observations at those states, we want to model the observation distribution at unseen states.

The majority of observation models are parametric. For example, [7] proposes a model for small sonar sensors. [8] characterizes a parametric model for ladar devices. Other work fits a parametric model to sensor readings, but estimates parameters at new states using nonparametric methods. [9] considered Gaussian observation models, but estimated parameters using Gaussian processes. Similarly, [10] used Gaussian process regression to predict models for multiple beams of a range sensor. [11] regressed covariances of a Gaussian distribution using nonparametric kernel smoothing. We take a finer-grained approach to sensor modeling, performing even density estimation in a nonparametric manner.

[1]The Robotics Institute, Carnegie Mellon University atallav1@andrew.cmu.edu, alonzo@cmu.edu
[2]Machine Learning Department, Carnegie Mellon University bapoczos@cs.cmu.edu

Procedure 1: P-REG

Input : Query x, Dataset $\mathcal{D} = \{(x_i, z_{ij})\}$
Output : Predicted distribution $\hat{p}(z|x)$

$\hat{\theta}_i \leftarrow \text{ESTIMATEPARAMETERS}(\{z_{ij}\})$
$\mathcal{D}^\theta \leftarrow \{(x_i, \theta_i)\}$
$\hat{\theta} \leftarrow \text{REGRESS}(x, \mathcal{D}^\theta)$
$\hat{p}(z|x) \leftarrow \hat{p}(z|x, \theta)$

Procedure 2: NP-REG

Input : Query x, Dataset $\mathcal{D} = \{(x_i, z_{ij})\}$
Output : Predicted distribution $\hat{p}(z|x)$

$\hat{p}(z|x_i) \leftarrow \text{ESTIMATEDENSITY}(\{z_{ij}\})$
$\mathcal{D}^z \leftarrow \{(x_i, \hat{p}(z|x_i))\}$
$\hat{p}(z|x) \leftarrow \text{REGRESS}(x, \mathcal{D}^z)$

The outline of the paper is as follows. Section II formalizes the procedures of parametric and nonparametric distribution regression. In Section III, we discuss the implementation in detail. Experiments on modeling real sensors and filtering in simulation are in Section IV.

II. FORMULATION

A. Predicting Distributions

Regression is the task of predicting an output from an input. Standard regression considers the case when the input and output are reals. If $x \in \mathbb{R}^p$ and $z \in \mathbb{R}^q$ denote the input and output, respectively, and $\mathcal{D} = \{(x_i, z_i)\}, i = 1\colon N$ is the training data, then standard regression is $\hat{z} = \text{REGRESS}(x, \mathcal{D})$. We will use the hat notation to denote predictions. To be concrete, we will further call x state and z observation. For sensors which are noisy, predicting an observation at a state is not very useful. It is the distribution of the observation, given the state, that is of interest. We denote the true distribution by $p(z|x)$ and the estimate by $\hat{p}(z|x)$. The task of predicting $\hat{p}(z|x)$ given x is distribution regression.

For distribution regression, assume that training data is available in the form $\mathcal{D} = \{(x_i, z_{ij})\}, i = 1\colon N, j = 1\colon M$. States are drawn iid from a state distribution, $x_i \sim p(x)$. The test distribution of states is assumed to be the same as the training distribution, which intuitively means that we will encounter states similar to those trained on. Observations are drawn iid from the observation distribution, $z_{ij} \sim p(z|x_i)$. These assumptions are simply the standard ones made by most learning procedures. It is not necessary that the same number of observations are logged at each state x_i. We take them all to be M for simplicity. Multiple observations at each state are required for good density estimates. This form of $\mathcal{D}$ is easy to collect in a number of cases. Consider a range sensor mounted on a mobile robot, as in Figure 1. The state is the robot pose, and environment map. The robot can drive to different poses and sit still while collecting data. On the other hand, there are cases where this form of data is hard to collect. Consider a contact sensor on a robot hand. If observations depend on how fast the hand approaches an object when making contact, then velocity is part of state. The state x is hard to maintain, since it changes the instant the hand makes contact. Multiple observations may be collected over repeated trials, but it is unlikely that contact will occur at the same state due to control errors.

A common approach is to parametrize the predicted distribution, $\hat{p}(z|x) = \hat{p}(z|x, \theta)$. $\theta_i \in \mathbb{R}^d$ is a vector of parameters which compresses the information in the observations $\{z_{ij}\}$. Given training data $\mathcal{D}$, parametric distribution regression involves two steps. First, parameters are estimated from the observations, resulting in a reduced dataset $\mathcal{D}^\theta = \{(x_i, \hat{\theta}_i)\}, i = 1\colon N$. At a query state x, we predict the distribution by estimating $\hat{\theta}$ via regression with $\mathcal{D}^\theta$, and constructing the distribution. We call this procedure P-REG, and it is summarized in Procedure 1.

We can now introduce nonparametric distribution regression. Given training data $\mathcal{D}$, it involves two familiar steps. First, since we only have samples $\{z_{ij}\}$ from $p(z|x_i)$, we construct nonparametric density estimates $\hat{p}(z|x_i)$[1]. This results in datasets $\mathcal{D}^z = \{x_i, \hat{p}(z|x_i)\}, i = 1\colon N$. At a query state x, we predict the distribution by regressing at each z with $\mathcal{D}^z$. We call this nonparametric regression procedure NP-REG, and it is summarized in Procedure 2.

Note that, for both P-REG and NP-REG, $\hat{p}(z|x)$ has to be computed for each z of interest, although we have suppressed this in the input to the procedures. If we are performing state estimation, $\hat{p}(z|x)$ only at the observation currently being processed is of interest. If we are performing simulation, we need $\hat{p}(z|x)$ over the domain of z in order to draw samples.

There are a number of methods to evaluate observation models (see [12] for a discussion), such as the likelihood of observations. We use the formal concept of risk from learning theory. The loss l of prediction is specified in terms of a metric d between the observation distribution and the prediction

$$l(p(z|x), \hat{p}(z|x)) = d(p(z|x), \hat{p}(z|x)) \tag{1}$$

The expected value of the loss under the state distribution $p(x)$ is the risk R of the prediction procedure. The empirical risk calculated from samples is $\hat{R}$. With the shorthand $l(p, \hat{p}) = l(p(z|x), \hat{p}(z|x))$, $l(p_i, \hat{p}_i) = l(p(z|x_i), \hat{p}(z|x_i))$, we have

$$R = \mathbb{E}[l(p, \hat{p})] \tag{2}$$

$$\hat{R} = \frac{1}{N} \sum_{i=1}^{N} l(p_i, \hat{p}_i) \tag{3}$$

The component algorithms in Procedures 1 and 2 (ESTIMATEDENSITY etc.) will have free parameters. During training, these free parameters will be selected to minimize empirical risk. Sensor observations often take values from a discrete set, $z \in \{z_1\colon z_B\}$. In such cases, we integrate the distribution to obtain the observation probabilities, and construct a histogram $h(x) \in \mathbb{R}^B$. The histogram has B

[1]We use the terms density and distribution interchangeably.

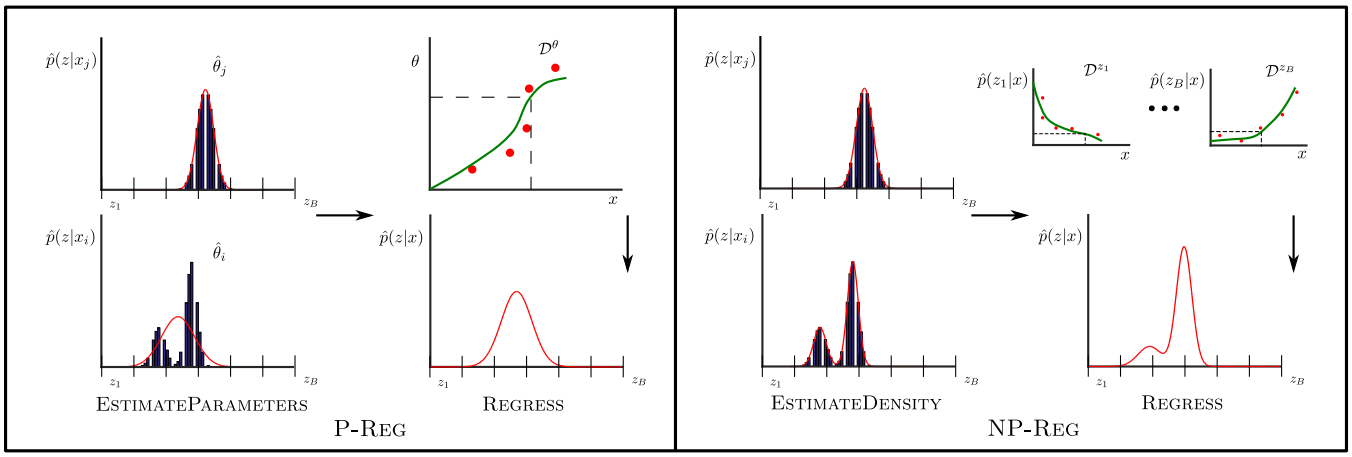


Fig. 2: Illustration of the two regression procedures. In P-REG, parameters θ_i are estimated for the training data. At a new state x, θ is estimated via regression. In NP-REG, distributions are estimated nonparametrically. At a new state x, each $\hat{p}(z_b|x)$ is predicted independently via regression.

bins, and stores probability values in each bin, $h^b(x) = P(z_b|x), b = 1\colon B$. The loss and risk definitions are modified to take histograms as arguments instead of distributions. The procedures are summarized in Figure 2.

B. Theoretical Guarantee

A detailed theoretical analysis of nonparametric distribution regression is carried out in [6]. Under assumptions on the smoothness of the observation distributions $p(z|x)$ and the state distribution $p(x)$, the procedure is shown to be consistent. Convergence rates on the risk are also derived.

C. Sources of Noise

The main advantage of NP-REG over P-REG is the increased expressiveness. Parametric assumptions are not made during modeling, and the theory justifies regression at the level of distributions. This gain comes at a price, as nonparametric procedures generally require more training data. Given high-frequency sensors and the ability to autonomously collect data, this is not a constraint.

NP-REG is obviously useful when $p(z|x)$ is inherently noisy. We use 'noisy' in an informal sense, to mean that the observation distribution is hard to approximate parametrically. We point out another case. Let us expand the state to $(x\ y)$. The part y of state is unmodeled. If the information y were available, it may be possible to build a good parametric model $\hat{p}(z|x, y)$ of the true distribution $p(z|x, y)$. But our model is $\hat{p}(z|x)$, and the effect of the umodeled states on this distribution is increased apparent noise.

Finally, there are situations in which sensor modeling is fundamentally hard. This is because both the parametric and nonparametric procedures rely on a REGRESS step. Regression is hard when there is little smoothness of outputs in the inputs. Consider a range sensor again, operating outdoors, pointed at a patch of grass (Figure 3). The state x is just the pose of the sensor. $p(z|x)$ may be hard to model parametrically at any state, because of effects such as pass-throughs (e.g. when lidar randomly senses the range to foreground or background surfaces). Nonparametric methods

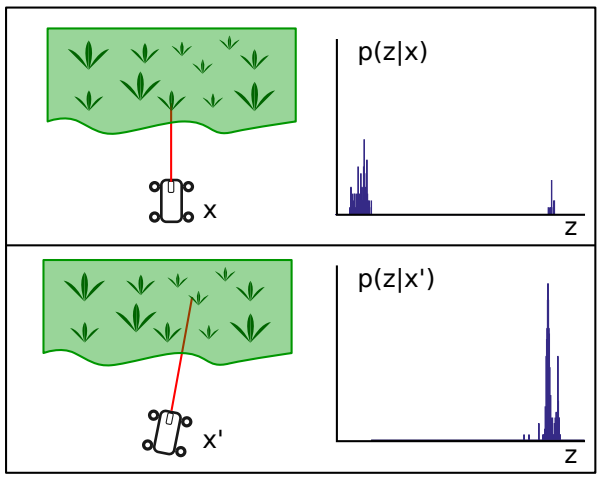


Fig. 3: Illustration of when sensor modeling is fundamentally difficult. Observation distributions vary widely for small changes in pose. $p(z|x)$ is not only noisy but also non-smooth in state.

can model distributions at any two states x and x'. However, between x and x', the distributions themselves may vary in a non-smooth manner, because blades of grass are spatially discontinuous.

III. IMPLEMENTATION

We detail implementation choices in this section. In our examples, the observation z is one-dimensional. We use consistent estimators for ESTIMATEPARAMETERS. For a Gaussian parameterization, $\theta = (\mu\ \sigma)$

$$\hat{\mu}_i = \frac{1}{M}\sum_{j=1}^{M} z_{ij},\ \hat{\sigma}_i = \frac{1}{M-1}\sum_{j=1}^{M}(z_{ij} - \hat{\mu}_i)^2 \qquad (4)$$

We make extensive use of kernel methods [13]. We will denote kernels by K. An example is the Gaussian kernel

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

For ESTIMATEDENSITY, we use kernel density estimation. Let K_z denote the observation kernel, and δ_z the observation kernel bandwidth

$$\hat{p}(z|x_i) = \frac{1}{M}\sum_{j=1}^{M} K_{z,\delta_z}(||z - z_{ij}||) \tag{5a}$$

$$K_{z,\delta_z}(||z - z_{ij}||) = \frac{1}{\delta_z} K_z\left(\frac{||z - z_{ij}||}{\delta_z}\right) \tag{5b}$$

It may be easier to work with probability histograms for sensors. Instead of kernel density estimates integrated to histograms, we directly use histogram estimates when we want to speed up predictions. With $\mathbb{I}$ denoting the indicator function

$$\hat{h}^b(x_i) = \frac{1}{M}\sum_{j=1}^{M} \mathbb{I}(z_{ij} = z_b),\ b = 1\colon B \tag{6}$$

For REGRESS, we use kernel smoothing. Let K_x denote the state kernel, and Δ_x the state kernel bandwidth. $\Delta_x \in \mathbb{R}^{p\times p}$ is a matrix, where p is the state dimension.

$$\hat{p}(z|x) = \frac{\sum_{i=1}^{N} \hat{p}(z|x_i) K_{x,\Delta_x}(||x - x_i||)}{\sum_{i=1}^{N} K_{x,\Delta_x}(||x - x_i||)} \tag{7a}$$

$$K_{x,\Delta_x}(||x - x_i||) = |\Delta_x|^{-\frac{1}{2}} K_x\left(\Delta_x^{-\frac{1}{2}}||x - x_i||\right) \tag{7b}$$

The kernel bandwidths δ_z and Δ_x are the hyperparameters of the procedures. They are selected to minimize the empirical risk $\hat{R}$ on holdout data. The observation kernel bandwidth δ_z is a scalar, but the state kernel bandwidth Δ_x is a matrix. It is common to choose a scaled identity matrix, $\Delta_x = \delta_x I_p$, where I_p is the $p \times p$ identity matrix. But not all dimensions of state have the same scale. To balance computational overhead with expressiveness, we use a diagonal matrix, $\Delta_x = \text{diag}(\delta_{x^1}\colon \delta_{x^p})$. The hyperparameters are selected using a grid search. Although not listed, we found that the choice of kernel was not crucial. Using a different kernel, such as the boxcar kernel, only led to different bandwidth values.

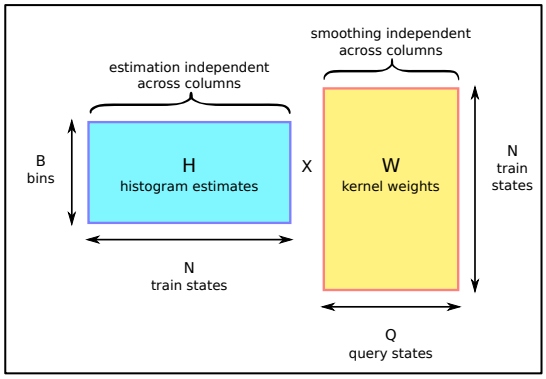


Fig. 4: With histogram estimation and kernel smoothing, prediction is an efficient matrix product.

Other choices exist for the component procedures, such as series estimation [14] for ESTIMATEDENSITY, and Gaussian processes [15] for REGRESS. With our choices, prediction using NP-REG is efficient. Let $H \in \mathbb{R}^{B\times N}$ denote the matrix of estimated histogram values, where $H_{bi} = \hat{h}^b(x_i)$. Suppose there are Q query states, $\{x_1\colon x_Q\}$. Let $W \in \mathbb{R}^{N\times Q}$ denote the matrix of weights derived from the state kernel, where $W_{iq} = K_{x,\Delta_x}(||x_q - x_i||)/(\sum_{i=1}^{N} K_{x,\Delta_x}(||x_q - x_i||))$. Then the $B \times Q$ matrix of predicted histograms at the query states is HW (Figure 4). Note that the histograms from the training data are linearly smoothed to make a prediction at a query state. This is a recurring feature of nonparameteric methods. With a histogram density estimator, $N = 10^3$ training states, $M = 100$ observations and $Q = 200$ query states, prediction takes about $0.5s$ when implemented in MATLAB. Histogram estimation is independent across training data, and linear smoothing is independent across queries.

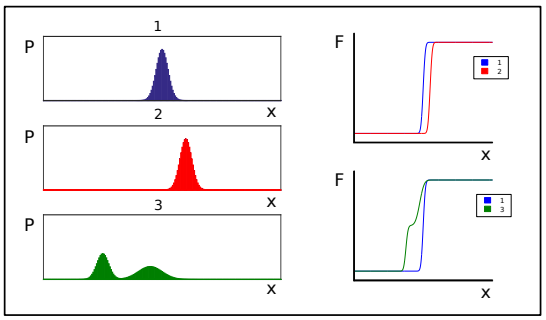


Fig. 5: The plots on the left are the histograms. Plots on the left show corresponding cdfs. Under the euclidean metric, histogram 1 matches 3 more closely than 2. Under the histogram match metric, histogram 1 matches 2 more closely than 3.

We need to specify the histogram metric d used to compute the loss l of prediction. The euclidean norm $d(h, \hat{h}) = \sum_{b=1}^{B}(h^b - \hat{h}^b)^2$ is not a good choice. It favors placing probability mass in the same bins, which may be low even with placing mass arbitrarily in other bins. This point is illustrated in Figure 5. Instead we use the histogram match, $d(h, \hat{h}) = \sum_{b=1}^{B} |\sum_{c=1}^{b} h^c - \sum_{c=1}^{b} \hat{h}^c|$. It is the $L1$ norm of the cumulative sums, and takes into account the position and distribution of probability mass.

IV. EXPERIMENTS

Our first two experiments are applying NP-REG to modeling real sensors. Our third experiment applies NP-REG as an observation model to a filtering problem.

A. Laser Sensor

We considered predicting observation distributions for a planar laser range sensor on a Neato robot. The maximum range, resolution, and other details are listed in Table I. The sensor has 360 equiangular bearings. We treat them identically and combine data from all bearings. The sensor was mounted on a robot, and training data $\mathcal{D}$ was collected by driving the robot around inside a known map. The map was an indoor scene, made of rectangular wooden walls. From the 2D pose and the map, local geometric information was extracted. For each bearing, the nominal range r and angle of incidence α to the nearest target were computed by raycasting. The state was then $x = (r\ \alpha)^T$, which may

sensor	frequency	number of bearings	[min range, max range]	resolution	B
laser	5 Hz	360	$[0, 4.5]$m	1mm	4501
field sensor	100 Hz	1	$[0, 20]$cm	0.1mm	2001

TABLE I: Sensor properties

also be interpreted as features extracted for prediction. As in other learning tasks, the choice of features can simplify the task. Local geometric features are more useful than the raw pose and map information. The observation z is simply the range reading.

We chose the number of observations M to be logged at each state based on the histogram metric d. A large number of observations were collected, and the resulting histogram was treated as ground truth. Picking a metric threshold of 0.5, we chose M such that the resulting histogram is within this threshold of the ground truth. For a representative distribution, a plot of loss in estimation versus M is shown in Figure 7a. Based on the threshold, $M = 70$ observations are enough.

For P-REG, we use a Gaussian parametric model. Parameters estimated at a state are $\theta = (\mu\ \sigma)^T$. The REGRESS procedure used in P-REG is locally weighted linear regression. This method also has bandwidths as hyperparameters, which we optimize on holdout data. Component algorithms for NP-REG are histogram density estimation and kernel smoothing. Empirical risk $\hat{R}$ on test data versus amount of training data N is shown in Figure 8a. Sample histogram predictions are shown in Figure 9a. NP-REG makes noisier predictions than P-REG, but this is appropriate since the true histogram is also noisy at a similar scale. Overall, P-REG performs better, because the observation distributions are well-approximated by the parametric model. This case supports the intuition that a parametric model is preferable when it is a good approximation. NP-REG will eventually achieve low risk, but the increased expressiveness is unnecessary.

B. Field Sensor

Our second application was predicting observation distributions for a magnetic field sensor, the Ascension trakSTAR. It consists of a small probe whose pose is measured when it is placed within a generated magnetic field. The small size of the probe makes it suitable for medical applications. For simplicity, we consider only one axis of operation. Other details are in Table I. The state x is the coordinate value along the axis, and the observation z is the state measurement. Data was collected by moving the sensor along the axis. Even within a workspace of $20cm$, the sensor shows interesting behavior. Referring to Figure 6, when the environment around the sensor is stationary, the observation distribution is well captured by a Gaussian. In reality, however, there could be metal objects around the probe that disturb the magnetic field, resulting in noisy distributions. These could be, for example, metallic surgical tools that are brought in the vicinity of the probe. It is difficult in practice to account for all such foreign objects in the state x, so they can be considered the unmodeled states y, which affect the observation distribution as noise.

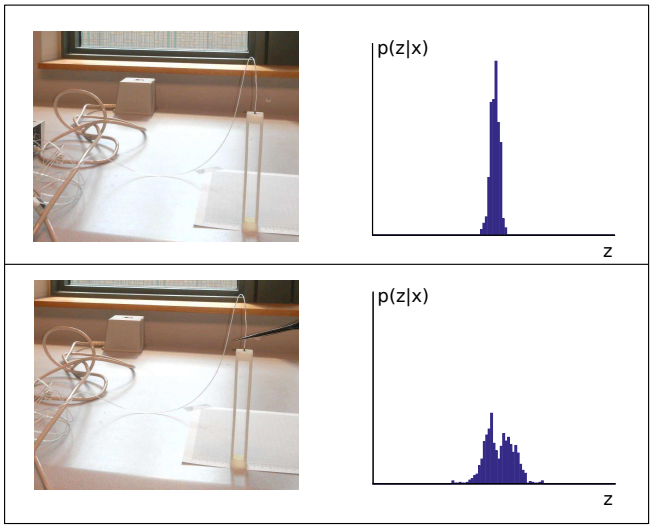


Fig. 6: The effect of unmodeled state on the observation distribution. Bringing a metal object near the sensor affects the field, making the observations noisier and non-Gaussian.

In our experiments, we deliberately introduced unmodeled states. A pair of metallic forceps was waved near the sensor while collecting data.

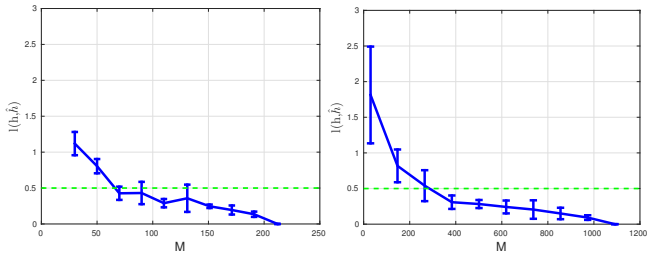
M was chosen in a similar manner as described for the laser sensor. The loss in estimation versus M is shown in Figure 7b. Reaching the threshold required about 300 samples. The more complicated distributions of the field sensor require a larger M for nonparametric modeling than the laser.

The choice of P-REG was the same as in the experiment with the laser. For both procedures, empirical risk on test data versus N is shown in Figure 8b. Not only does NP-REG have lower risk in this case, it continues to improve with increasing data, while the risk for P-REG converges quickly. The hard to model distributions benefit from the nonparametric modeling. Sample histogram predictions are in Figure 9b. NP-REG comes closer to capturing the multimodality of the observation distribution.

C. Particle Filter

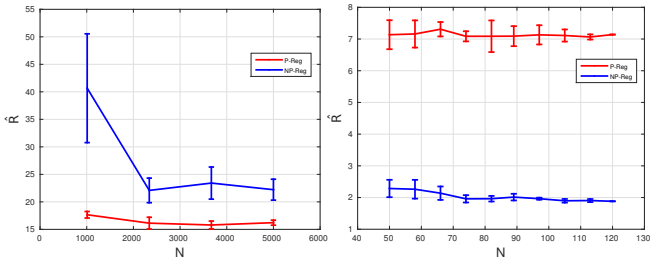
We studied the utility of NP-REG to a problem of state estimation using a particle filter. A particle filter maintains a set of particles and associated weights, $\{(x_i, w_i)\}$, that are updated in response to readings. When an observation is received, an observation update is performed. The state of the particles is unchanged, but the weights are updated. Weights are proportional to the probability of the observation, $w_i \propto P(z|x_i)$. This is followed by a resampling step. Further details of the particle filter can be found in [1].

An observation model is utilized in the observation update step, to predict $\hat{P}(z|x_i)$ at the unseen state x_i. We consider



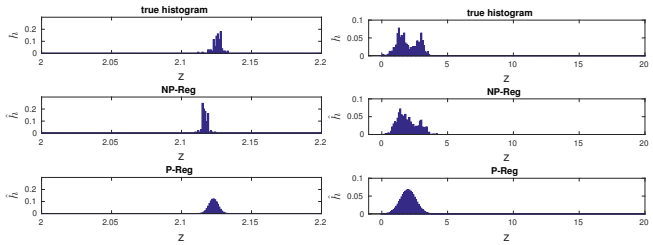
(a) Laser sensor (b) Field sensor

Fig. 7: Loss $l(h, \hat{h})$ versus number of observations M. x-axis scales are different.



(a) Laser sensor (b) Field sensor

Fig. 8: Empirical risk $\hat{R}$ versus number of states N. Scales are different.



(a) Laser sensor (b) Field sensor

Fig. 9: Sample predictions. Scales are different. For the laser sensor, a section of the z-axis is zoomed into for clearer visualization, since the distributions have low variance. For the field sensor, the entire range is on the z-axis.

state estimation for a mobile robot in a planar environment. The state x consists of the pose of the robot (location and orientation), and a nominal map. Only the pose of the robot is estimated. The robot is fitted with a range scanner. An observation z is a range reading at a particular bearing. For this case, a standard observation model is described in [1], which we refer to as the classic model. Alternately, we can use NP-REG to obtain particle weights. A challenging scenario for estimation is the presence of unmodeled objects in the environment. The state includes only the nominal map, but the observation is affected by unmodeled objects. The classic model is parametric. It has a Gaussian distribution component to account for range readings from the nominal map, and an exponential distribution component to account for those from unmodeled objects. The parameters of the classic model, such as the means and the mixing weights of the components, are tuned on training data. Ideally, the data contains range readings from unmodeled objects, so the observation model is trained appropriately.

Observation model	Classic model	NP-REG
Mean pose error (m)	0.004 ± 0.041	0.002 ± 0.020

TABLE II: Filter errors

To compare the classic model and NP-REG, we built an environment in simulation. The nominal map consists of a straight, wide corridor, and the unmodeled objects are blocks scattered near the walls. Further, the unmodeled objects are sampled at different positions for each observation, in order to represent dynamic objects, such as people. See the first half of Figure 11. The map is in black, and unmodeled objects in grey. The true robot pose is a red arrow, and a set of particles is shown in green. Of these, we focus on a single particle, whose pose is also shown in green. This particle's pose is close to the true pose, and we would expect it to be assigned a high weight. We focus on a reading at a specific bearing of the range sensor. As depicted, the observed range is smaller than the nominal range, due to an unmodeled obstacle. The second half of Figure 11 shows the predicted distributions. The observation has a low probability under the classic model, despite the exponential term. On the other hand, the distribution predicted by NP-REG has peaks at both the nominal and observed reading. Note that the same data (consisting of unmodeled objects) was used to train both models. It is possible to reason about dynamic obstacles in more detail and build a better parametric model, as in [4]. The point of the example is that NP-REG required no modifications for this specific situation. Being a nonparametric procedure, it adapts to the training data to predict realistic distributions.

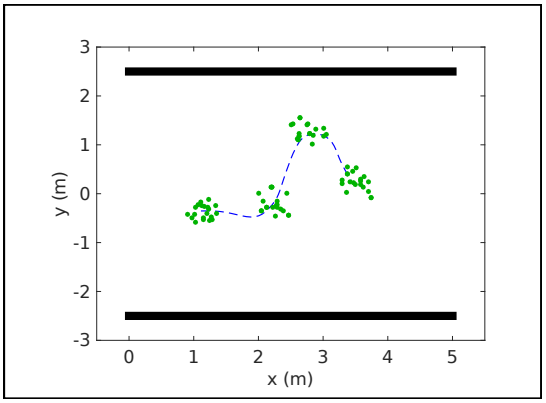


Fig. 10: Snapshots from the operation of a particle filter. The nominal map is shown in black. Unmodeled objects are not shown. The true trajectory is a blue dotted path. Particles are shown in green.

Overall particle filter performance is a non-trivial function of a number of variables in addition to the observation model. Figure 10 shows a particular trajectory we ran both filters on. For the most likely estimate of the trajectory, errors are shown in Table II. Due to space constraints, we omit results on other trajectories. We observed that the filter with NP-REG had lower error and variance than that with the classic model.

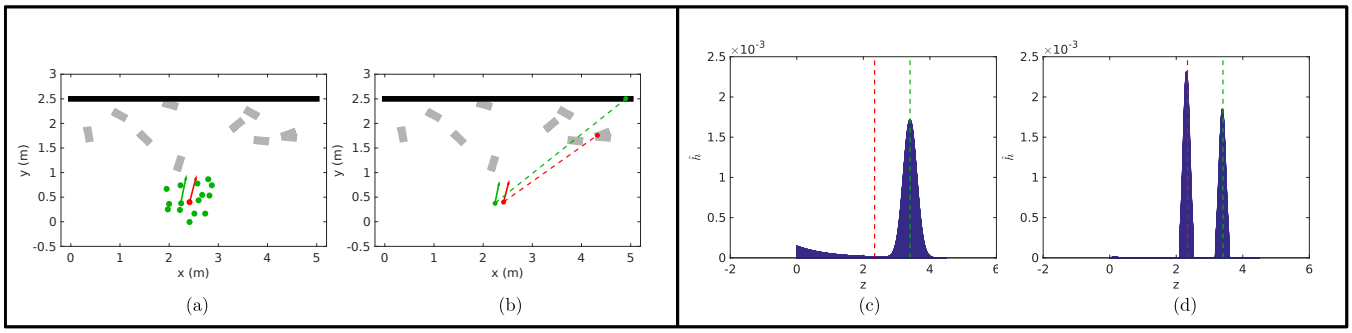


Fig. 11: Observation probabilities under the different models. Figure 11(a) depicts the environment. We focus on a specific particle. Figure 11(b) depicts the nominal and observed range readings. Figure 11(c) depicts the histogram predicted by the classic model. Figure 11(d) depicts that predicted by NP-REG.

V. CONCLUSION

We have addressed the problem of modeling and predicting observation distributions. We proposed nonparametric distribution regression, developed in the learning community, as a procedure to model sensors. It is theoretically well-founded and simple to implement. We believe it can play a useful role in any modeler's toolbox, as a procedure that adapts to data to capture inherent noise, without requiring the designer to make parametric assumptions.

There are a number of sensing conditions which are inherently noisy, such as range sensors in smoky environments [16], and sonar sensors underwater [17]. NP-REG could be a good modeling procedure for such problems.

Experiments in our work involved one-dimensional observation spaces. Multi-dimensional observations were dealt with by predicting for each dimension independently. Maintaining histograms for high dimensions is infeasible. A useful next step in this work will be to incorporate density estimators under which prediction remains tractable. Our work required multiple observations to be logged at a state, which can be difficult in some cases. Therefore, another relevant future direction will be to relax this requirement. Finally, speeding up the procedure by intelligently selecting data will help use the procedure in real-time.

VI. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant Number 1317803. The authors also thank Sanjiban Choudhury, Humphrey Hu and Arun Srivatsan of the Robotics Institute for discussions and access to sensors.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[2] S. M. Ross, *A course in simulation*. Prentice Hall PTR, 1990.

[3] L. Kleeman and R. Kuc, "Sonar sensing," in *Springer Handbook of Robotics*. Springer, 2008, pp. 491–519.

[4] T. De Laet, J. De Schutter, and H. Bruyninckx, "A rigorously bayesian beam model and an adaptive full scan model for range finders in dynamic environments." *J. Artif. Intell. Res.(JAIR)*, vol. 33, pp. 179–222, 2008.

[5] B. Póczos, A. Rinaldo, A. Singh, and L. Wasserman, "Distribution-free distribution regression," *arXiv preprint arXiv:1302.0082*, 2013.

[6] J. Oliva, B. Póczos, and J. Schneider, "Distribution to distribution regression," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1049–1057.

[7] J. Müller, A. Rottmann, L. M. Reindl, and W. Burgard, "A probabilistic sonar sensor model for robust localization of a small-size blimp in indoor environments using a particle filter," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3589–3594.

[8] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash ladar devices," in *International Conference of Sensing and Technology*, vol. 2, 2005.

[9] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[10] C. Plagemann, K. Kersting, P. Pfaff, and W. Burgard, "Gaussian beam processes: A nonparametric bayesian measurement model for range finders." in *Robotics: Science and Systems*, 2007.

[11] W. Vega-Brown, A. Bachrach, A. Bry, J. Kelly, and N. Roy, "Cello: A fast algorithm for covariance estimation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3160–3167.

[12] B. Browning, J.-E. Deschaud, D. Prasser, and P. Rander, "3d mapping for high-fidelity unmanned ground vehicle lidar simulation," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1349–1376, 2012.

[13] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[14] S. Efromovich, "Orthogonal series density estimation," *Wiley Interdisciplinary Reviews: Computational statistics*, vol. 2, no. 4, pp. 467–476, 2010.

[15] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[16] O. Formsma, N. Dijkshoorn, S. van Noort, and A. Visser, "Realistic simulation of laser range finder behavior in a smoky environment," in *RoboCup 2010: Robot Soccer World Cup XIV*. Springer, 2011, pp. 336–349.

[17] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1445–1464, 2012.