# Extensible Data-driven Classification of Robot Sensor Data

Daniel H Grollman and Odest Chadwicke Jenkins
and Frank Wood

Department of Computer Science
Brown University
Providence, Rhode Island 02912

**CS-05-11**
July 2005

# Extensible Data-driven Classification of Robot Sensor Data

**Daniel H Grollman** and **Odest Chadwicke Jenkins** and **Frank Wood**
Brown University Department of Computer Science
Providence, RI 02906
{dang,cjenkins,fwood}@cs.brown.edu

**Abstract**

We present an unsupervised method for online classification of robot sensor data. Unlike previous work in which human intuitive correspondences are sought between sensor data and classes of physical space (room, wall, corner, door, etc.), our method learns intrinsic classes based on the characteristics of sensor data. We approximate the manifold underlying sensor data using Isomap nonlinear dimension reduction and use classical Bayesian clustering (Gaussian mixture models) with model identification techniques to discover classes. The learned model can then be used to classify new (out-of-sample) data in real time. We apply our method to sensor data of different modalities and from different physical spaces. Our results demonstrate the robustness of the method with respect to noise and robot location variance. The extensibility of our approach allows us to classify large and streaming data sets. We conclude that data-driven classification may serve as a more solid foundation for applications that require the ability to discriminate physical spaces surrounding a robot.

## Introduction

The understanding and interpretation of sensor data is a current issue in robotics research. While interacting with the world, many robots collect and act upon data from their sensors. For an autonomous robot, this data is one of its primary sources of knowledge about the world, and consequently, the robot's perception of the world is heavily dependent upon its sensor suite. Specifically, the classes of physical space that can be robustly recognized are limited by the robot's sensors. Additionally, the routines used to interpret data from sensors are usually provided by a human programmer. Many robot sensors are not completely understood or have no direct analogue to human senses (e.g laser rangefinders). Consequently, our understanding of how the world is perceived by robots with these sensors is biased and heuristic.

Consider the sonar sensor commonly used on robots. Under the standard "ray model", a sonar chirp exits the sensor in a line, hits an object, and returns along the same ray. The distance to the object is calculated based on the time-of-flight of the chirp. Research has shown that this model is somewhat inaccurate, for sonar has lobes and beam width, surfaces have specularity, and sound diffracts. Still, many simulators use this "ray model" to simulate sonar data (Lacroix & Dudek 1997). Additional work (Kuc 2004) has shown that by utilizing multiple return echos, texture information can be retrieved in addition to distance. Until now, these secondary echos have been mostly ignored.

A data-driven approach to sensor analysis could discover a more appropriate interpretation of sensor readings, as opposed to one based on our preconceptions of a sensor's functionality. Many common techniques for understanding the relationship between sensor output and the environment depend heavily on our understanding of a robot's sensors and reflect the bias of human developers. Humans identify areas in the robot's world, such as corners, hallways, and doors, and try to build models of how they would appear to the robot given its sensors. Such models are biased toward human intuition and human senses and may not be appropriate for autonomous robots. Through data-driven analysis, it may be possible to uncover intrinsic structure in sensor data. As a result, the robot's perceptual system is allowed to develop physical space classes based on its own, unique, sensory input.

In this paper, we present a data-driven method for classifying robot sensor input via unsupervised dimension reduction and Bayesian clustering. We view the input space of the system as a high dimensional space where each dimension corresponds to a reading from one of the robot's sensors. Our method attempts to uncover a lower-dimensional embedding that captures the intrinsic dimensions of the experienced sensor data. Distances in this embedding correspond to the variance among sensor readings as the robot moves about its environment. By clustering in this embedded space, we group regions that appear similar to the robot together with each cluster representing a physical space class.

We describe an out-of-sample classification procedure for quickly evaluating new samples with respect to our learned classification. This procedure uses an out-of-sample technique (Bengio *et al.* 2004) to project samples into the embedding, where they can be classified with a Gaussian mixture model (GMM). Out-of-sample classification provides extensibility for large or streaming datasets.

We present results from applying our methodology to various types of real-world sensor data. We demonstrate the utility of our method in: 1) a streaming scenario where an explored space can be classified consistently on future visits and 2) a large dataset scenario where sensor data classes are learned from a small subset of the data and used to classify the remainder of the dataset.

## Related Work

A more robust physical space class identification system would be beneficial to many areas of mobile robotics research. For instance, topological mapping depends on the ability to discover regions in the explored area (Thrun 1998). This process is usually done by extracting features from the sensor data that indicate the robot's current location. The decision of which features to extract is usually made by humans who decide which region types exist in the world and choose the features they think would be noticable to the robot's sensors (Tomatis, Nourbakhsh, & Siegwart 2003). Sometimes, the definitions of those features are written in terms of the robot's sensors by humans. However, as mentioned in the previous section, some common sensors are imperfectly understood. A system that operated on real sensor data alone to develop and identify physical space classes could reduce this human bias.

Localization techniques can also benefit from enhanced region identification. Landmarking, or the identification of unique places, is commonly used to let a robot know when it has returned to a previously visited location on a map (i.e., revisiting, loop closure). The revisiting problem is key when it comes to map-making because it allows a robot to discover loops in the world (Howard 2004), or in the case of multiple exploration robots, it allows one robot to discover when it has entered space explored by a different one (Stewart *et al.* 2003). Landmarking is often accomplished by modification of the environment, adding beacons or some other unique, easily distinguishable identifier to a place. When locations are not uniquely identifiable, localization depends on estimating the location of the robot, using, for example, a Hidden Markov Model (Shatkay 1998). Another technique uses the connections between regions to localize the robot and build a map (Howard, Matarić, & Sukhatme 2001). All of these approaches require a robust way of identifying the class of space that the robot is currently in.

One of the obstacles that has to be surmounted when learning physical space classes is that a space's class has to be identifiable from many different viewpoints for robust classification. In addition, every place within a particular region of space should be classified a being in the same class. If the sensor modality is vision, this means that any image of a space has to be recognized as coming from that space, even if the image was previously unseen. In (Kosecká & Li 2004), features were extracted from multiple camera images of a space to come up with a representation of images of the space itself. This method allows for new images to be correctly classified as being from that space. However, the actual segmentation and labeling of the training images into categories was done by humans. Additionally, (Weng & Chen 2000) uses linear subspace methods with a partition tree on robot vision data. Because robot sensor data is potentially non-linear, we consider a non-linear alternative to subspace embedding.

In a more general case, (Tapus, Tomatis, & Siegwart 2004) learns a ring of features (a fingerprint) around the robot to identify a place. These fingerprints involve data from both vision and laser sensors. By modeling the occlusion of features in their fingerprint identification algorithm, they enable the robot to identify a location from multiple positions inside the location.

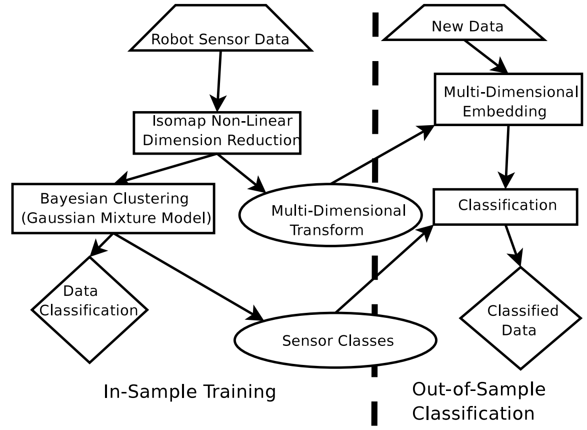



Figure 1: Our method in flowchat form. Robot sensor data is collected and analyzed with Isomap to obtain a low-dimensional embedding. The embedded data is then clustered to develop sensor classes. Out-of-sample data can be quickly transformed and classified using the information developed from the in-sample training.

Similarly, (Klingspor, Morik, & Rieger 1996) presents a technique where sensory and action concepts are learned directly from the sonar data of a robot, once the data is segmented and categorized by hand. In contrast, the proposed technique segments and categorizes the sensor data in an unsupervised way.

In the field of dimension reduction (DR), many methods (Duda, Hart, & Stork 2000) exist that could be applied to robot sensor data. Principal component analysis and independent component analysis are well suited to extracting linear subspaces from limited dimensional data. Self organizing maps are sensitive to initialization and parameter tuning. Recently, Isomap (Tenenbaum, de Silva, & Langford 2000) has emerged as a capable non-linear DR algorithm with convergence guarantees and out-of-sample extensions (Bengio *et al.* 2004).

## Methodology

Our method, outlined in Figure 1, views $d$-dimensional robot sensor data as lying on a manifold in $\mathbb{R}^d$. Furthermore, we model each sensor datum $\vec{x}$ as having been generated by a mixture model in which each mixture density corresponds to an intrinsic sensor data class. Here we closely follow the methods and notation of (Bengio *et al.* 2004).

### Training

Let $D = [\vec{x}_1, \ldots, \vec{x}_N]$ be a collection of $N$ sensor outputs used to train the model. We compute an affinity matrix $M$ by approximating the geodesic distance between points on the sensor data manifold. As in (Tenenbaum, de Silva, & Langford 2000; Bengio *et al.* 2004), we define the geodesic distance between points $a$ and $b$ to be:

$$\tilde{D}(a,b) = \min_p \sum_i d(p_i, p_{i+1})$$

where $p$ is a sequence of points of length $l \geq 2$ with $p_1 = a$, $p_l = b$, and $p_i \in D \forall i \in 2, \ldots, l-1$ and $(p_i, p_i + 1)$ are neighbors. The neighbors for each point are determined by Euclidian distance, using either an $\epsilon$-radius or $k$-nearest neighbors algorithm. We compute $\tilde{D}$ by using Dijkstra's algorithm (Cormen, Leiserson, & Rivest 1990) to expand a neighborhood graph (as defined by the neighborhood function) into a full shortest-path matrix.

An affinity matrix $M$ is formed with elements $M_{ij} = \tilde{D}^2(x_i, x_j)$. These distances are converted to equivalent dot products using the "double-centering" formula:

$$\tilde{M}_{ij} = -\frac{1}{2}(M_{ij} - \frac{1}{2}S_i - \frac{1}{N}S_j + \frac{1}{N^2}\sum_k S_k)$$

where $S_i = \sum_j M_{ij}$. In practice, this grows as $N^2$ and is thus currently infeasible to calculate for more than a few thousand points.

The $k$ dimensional embedding $\vec{e}_i$ of each sensor output $\vec{x}_i$ on the sensor data manifold is approximated by the vector $\vec{e}_i = [\sqrt{\lambda_1}v_{1i}, \sqrt{\lambda_2}v_{2i}, \ldots, \sqrt{\lambda_k}v_{ki}]$ where $\lambda_k$ is the $k^{th}$ largest eigenvalue of $\tilde{M}$ and $v_{ki}$ is the $i^{th}$ element of the corresponding eigenvector. We reduce the dimensionality of the sensor data by setting $k < d$ thus removing many of the low eigenvalue coordinates of the embedding. In practice, depending on the data, we keep the 5-10 most significant coordinates of $\vec{e}_i$.

Let $E = \vec{e}_1, \ldots, \vec{e}_N$ be the reduced dimensionality embedding of the training sensor data $D$. From this point forward we use "sensor embedding" to refer to the reduced dimensionality embedding of the sensor data.

We assume that the sensor embeddings were generated by exactly $M$ statistically distinct intrinsic classes of physical environment (regions). We assume that the distribution of each of these classes is Gaussian and fit $E$ with a Gaussian mixture model with $M$ mixture densities.

The probability that $\vec{e}_i$ was output by the robot's sensors while in physical space class $j$, $1 < j < M$ given these assumptions is:

$$P(\vec{e}_i|j) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{det(\Sigma_j)}} exp(-\frac{1}{2}(\vec{e}_i - \mu_j)^T \Sigma_j^{-1}(\vec{e}_i - \mu_j))$$

where $\mu_j$ and $\Sigma_j$ are the mean and covariance of the sensor output generated while in physical space class $j$.

Assuming that each sensor datum is independent, then the probability of $E$ according to the mixture model is:

$$P(E) = \prod_{i=1}^{N}\sum_{j=1}^{M} \alpha_j P(\vec{e}_i|j)$$

where the $\alpha_j > 0$ are mixing coefficients and $\sum_{j=1}^{M} \alpha_j = 1$.

The EM algorithm (Bilmes 1997; McLachlan & Basford 1988) is used to maximize $P(E)$ by solving for optimal distribution parameters and membership weights. This is accomplished by the iterative optimization of a log likelihood function:

$$\log(\mathcal{L}(\Theta|E, \mathcal{Y})) = \sum_{i=1}^{N} \log(\sum_{j=1}^{M} \alpha_{y_i} P(\vec{e}_i|\Sigma_j, \mu_j))$$

where $\Theta = \{\mu_1, \ldots, \mu_M, \Sigma_1, \ldots, \Sigma_M\}$ is a set of unknown distribution parameters corresponding to the mean sensor data embeddings and covariance matrices for the $M$ posited physical space classes,

$$\mathcal{Y} = \{y_i\}_{i=1}^{N}, 1 < y_i < M, y_i \in \mathbb{Z}$$

is an array of unknown indicator variables such that $y_i = j$ if $\vec{e}_i$ came from mixture component $j$. These $y_i$ can be thought of as "hidden" variables.

Model selection is a central issue in clustering and corresponds to determining the number of clusters in the data; in this context determining the distinct instrinsic physical space classes in the sensor data. We do not solve this problem; instead we use two empirical criteria for choosing the model. One, we look for an inflection or maxima in a penalized training data log-likelihood. We used the Bayesian Information Criteria (BIC) which penalizes the likelihood as a function of the complexity of the model. If $\kappa$ is the number of free parameters in the model, then we calculate the BIC as:

$$-2\log(\mathcal{L}(\Theta|E, \mathcal{Y})) - \kappa(\log(N) + 1)$$

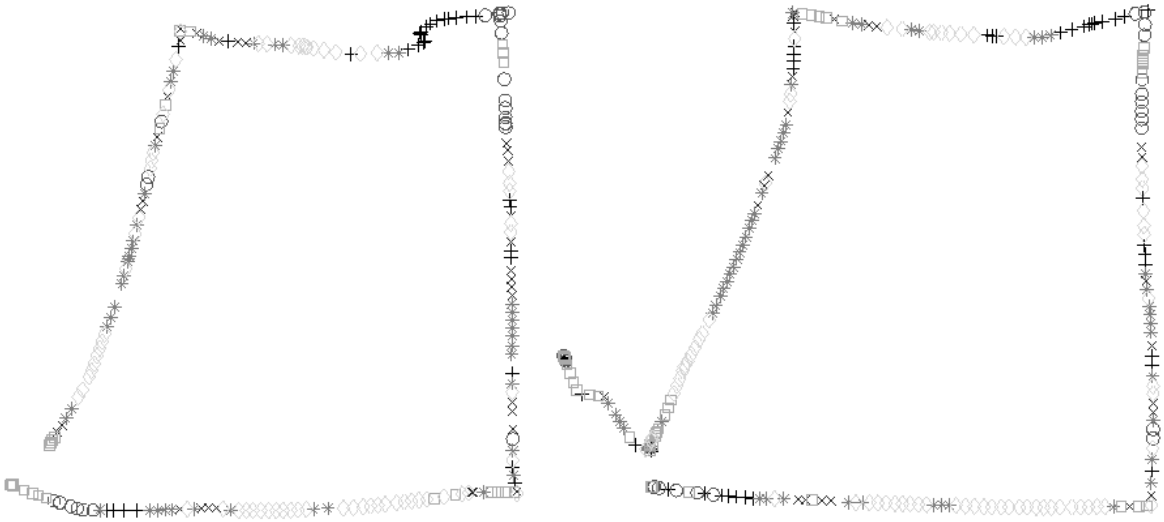
In practice, the BIC often doesn't sufficiently penalize complex models, so we additionally use cross-validation on held-out data to check for overfitting. To do this we train our model on half the training data and then compute the unpenalized likelihood of the held out half. When too many classes are posited, i.e. the model is over-fit, the likelihood of the held out data decreases. These two techniques are often in agreement.

## Online Classification

Online classification of a new point $\vec{p}$ is simple and rapid. We refer the reader to (Bengio *et al.* 2004) for full details. The embedding is given by:

$$e_k(\vec{p}) = \frac{1}{2\sqrt{\lambda_k}} \sum_i v_{ki}(E_{\vec{x}}[\tilde{D}^2(\vec{x}, \vec{x}_i)] + E_{\vec{x}'}[\tilde{D}^2(\vec{p}, \vec{x}')] - E_{\vec{x},\vec{x}'}[\tilde{D}^2(\vec{x}, \vec{x}')] - \tilde{D}^2(\vec{x}_i, \vec{p}))$$

where $E$ is an average over the training data set. This new embedded point may be classified by finding the weighted cluster under which it is most likely.



(a) trip1 (b) trip2

(c) Class1 (d) Class2 (e) Class3 (f) Class4 (g) Class5 (h) Class6

Figure 2: Results from the Crunch (8 sonar, 8 IR) data set. 2(a):Sensor data from the first of two similar trips has been clustered into 6 classes. There is a unique shape and grayscale value for each class overlaid on the odometry from the robot to show how the classes correspond to regions of space. 2(b):The data from the second trip are classified using the model learned from the first trip. Many areas are similarly classified across both trips. 2(c)-2(h) Show the average sensor readings for each cluster under the standard ray model of sonar. Classes 1, 2 and 4 may appear similar to a human under this model, but are discernable to our method.

## Experiments

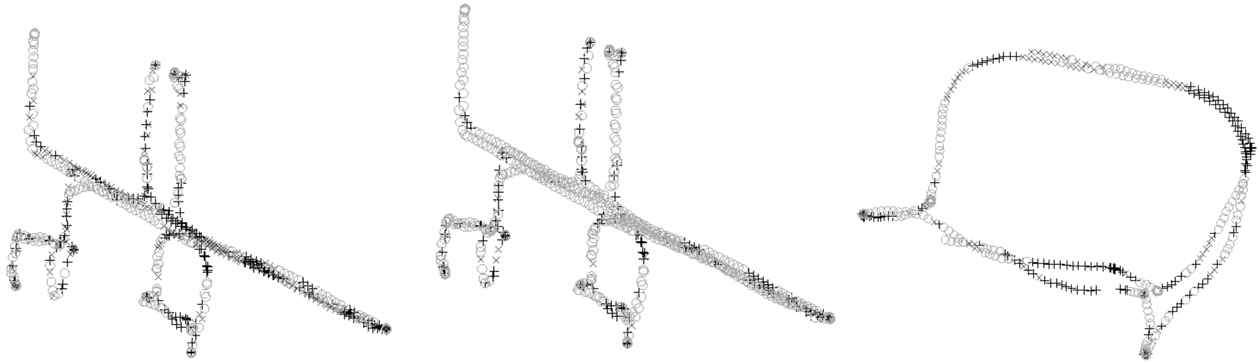
### Crunch

Our first experiments use sensor data from a small, cylindrical inverted pendulum robot named Crunch. Crunch is equipped with eight sonar and eight IR sensors arranged in dual rings around its body. During operation, these sensors are sampled and transmitted back to a base laptop where they are logged at around 10Hz. Our data consists of the robot's logs as it takes two trips along a path ($\sim$1000 points each). We use the sensor readings from the first trip to learn sensor classes and treat the second trip as streaming data, classifying the data in real time to test the robustness of the classification.

Figure 2 shows the results of our experiments with the Crunch data set. After computing the geodesic distance and MDS embedding of the first trip, we retain 10 of the resulting dimensions. Based on the BIC and holdout calculations, we estimate that there are 6 classes in the data. Each datapoint is assigned a class, and each class is assigned a shape and grayscale value. Using the odometry from the robot, we overlay these classes on the path that the robot followed as shown in 2(a).

Using the embedding and the classes derived from the first trip, we classified all the datapoints in the second trip. As the robot followed approximately the same path as it did in the first trip, we expect the physical spaces along the path to be classified similarly. The results from the out-of-sample classification are shown overlaid on the odometric map in Figure 2(b). Comparing this with the map of the first trip, one sees that the classification of physical space is similar across both trips. (The robot did not follow the *exact* same path each time, so some variance is expected).

Sensor readings from one class of physical space should be similar to each other and different from those in from other classes of space. Figures 2(c) to 2(h) show the average readings from each of the 6 classes discovered by our method. These images were generated using a "ray model" of Crunch's sensors. Classes 1, 2 and 4 appear similar under this model to humans, but they are distinguishable to the robot. This fact suggests that there is information in the sensor readings that is not captured by this model, but is used by the system to make distinctions between classes of physical space.



(a) All points (b) First 500pts as landmarks (c) Loop Closeup

(d) Class1 (e) Class2 (f) Class3

Figure 3: The Fr079 data set from Cyrill Stachhniss. In 3(a), the entire data set was processed to determine the embedding and classes. In 3(b), similar results were obtained by using the first 500 datapoints as landmark points. 3(c) shows a closeup of one of the loops in the dataset. The same locations are classified into the same classes when revisited. 3(d)-3(f) Show the average laser ranges of each of the 3 classes discovered in the data.

### Fr079

A second, larger dataset was the Fr079 dataset obtained from the Robotics Data Set Repository, Radish (Howard & Roy 2003)[1]. This set consists of $\sim$3000 datapoints of 360 degree laser rangefinder scans. This data set contains several loops which can be examined for robustness (i.e, regions of space should be classified similarly upon revisiting), as we did with the Crunch data. Because of its size, this data set serves as a useful test bed for examining how to select “landmark” points for training when processing large amounts of data. Although the entire data set can be processed, training can be greatly speeded up by only performing the training phase (Geodesic distance, MDS, GMM) on a small subset of the data, and treating the rest as out-of-sample points. We examined two possible landmark selection procedures. For the first we took the first 500 points as the landmarks, and in the second we took a random sampling of 500 points.

Figure 3 displays our results from the Fr079 data set. We found that using only the first 500 datapoints as landmarks yielded comparable results to using the entire data set, in a fraction of the time. However, there are differences between the class-coded odometric plots of these experiments. Using random data samples as the landmarks produced similar results. We believe this is because some classes did not appear, or were not adequately represented, in the landmark points.

Figure 3(c) shows a closeup of one of the loops made by the robot in its explorations. Each time the robot is at the same location in the world (or very nearby), the system detects it as being in the same region type.

[1] http://radish.sourceforge.net

## Discussion and Future Work

Our approach attempts to remove human bias from the analysis of robotic sensor data by identifying latent structure in the sensor readings themselves. By using unmodified real (as opposed to simulated) sensor data from multiple sensor types, we remove the reliance on any particular model of sensor operation or noise and show the generality of our approach with respect to sensor type.

In the real world, computational and temporal limits restrict a robot’s ability to process data. Thus, it is useful for a system to do as much processing offline and apply what it learns to new data in real time. While it would be best to run our entire system everytime the data changes, this is infeasible. The out-of-sample extensibility of our system provides the ability to deal with new data as it arrives. However, this introduces a new problem: choosing training data. Our experience with the Crunch and Fr079 data sets confirm the intuition that it is best to have every physical space class represented in the training data. This is not always possible, particularly if a robot enters a heretofore unseen environment during operation. Thus, it should be able to choose new landmarks and recompute the embedding and classes as needed. Future work will focus on detecting when this recomputation is required.

### Interpretation

When a space is fully explored and processed, as in the Crunch experiments and the full processing of the Fr079 set, our method can be viewed as a robust way of detecting that the robot is in the same class of space when a location is revisited (as in the second trip in the Crunch set, and in the loops of the Radish set). However, the classes found by our method do not always correspond to human intuition. The Fr079 classes (Figures 3(d) - 3(f)) seem to correspond to, respectively, a left turn, a straight hallway, and a right turn, but some of the Crunch classes (Figures 2(c) - 2(h)) do not submit to such an easy description. We believe this is because the laser rangefinder used in the Fr079 set is very well modeled by the "ray model", while the sonar and IR sensors on Crunch are not. When Crunch sensor data are viewed under that model, the "intrinsic" classes discovered by our method are fairly indistinguishable.

### Issues

We currently empirically determine the neighborhood function and size, the number of embedding coordinates to retain, and the number of intrinsic physical space classes. Each of these could in theory be automatically determined by finding maxima, etc. Our contribution is demonstrating that intrinsic classes may form a better foundation for applications that require classifying physical space from sensor data. For a production system, and in our future work we will need to find solutions for these problems.

In addition, we treat each sensor reading as independent. Better performace may result from leveraging more information about the data. For instance, ST-Isomap, (Jenkins & Matarić 2004), provides a way to leverage the spatial and temporal nature of the data when calculating the distance matrix between data points. Stochastic Neighbor Embedding (Hinton & Roweis 2002) is yet another, probabilistic approach to DR that may be useful.

## Conclusion

This paper presents an extensible method for data-driven discovery of region classes in robot sensor data. Using Isomap nonlinear dimension reduction and Bayesian clustering, the system discovers robot-distinguishable sensor data classes. Discovered classes are based on the actual data seen by the robot, so they depend on no particular model of the sensors or environment.

In order to deal with large data sets and streaming data, our method can be trained on a few "landmark" datapoints and used to classify new, previously unobserved data.

## References


Bengio, Y.; Paiement, J.; Vincent, P.; Delalleau, O.; Roux, N. L.; and Ouimet, M. 2004. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*.

Bilmes, J. 1997. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, University of Berkley.

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. MIT Press/McGraw-Hill.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Hinton, G., and Roweis, S. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*. 857–864.

Howard, A., and Roy, N. 2003. The robotics data set repository (radish).

Howard, A.; Matarić, M. J.; and Sukhatme, G. S. 2001. Relaxation on a mesh: a formalism for generalized localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1055–1060.

Howard, A. 2004. Multi-robot mapping using manifold representations. In *IEEE International Conference on Robotics and Automation*, 4198–4203.

Jenkins, O. C., and Matarić, M. J. 2004. A spatio-temporal extension to isomap nonlinear dimension reduction. In *The International Conference on Machine Learning (ICML 2004)*, 441–448.

Klingspor, V.; Morik, K. J.; and Rieger, A. D. 1996. Learning concepts from sensor data of a mobile robot. *Machine Learning* 23(2-3):305–332.

Kosecká, J., and Li, F. 2004. Vision based topological markov localization. In *The International Conference on Robotics and Automation (ICRA 2004)*.

Kuc, R. 2004. Neuro-computational processing of moving sonar echos classifies and localizes foilage. *Acoustical Society of America* 116(3):1811–1818.

Lacroix, S., and Dudek, G. 1997. On the identification of sonar features. In *The International Conference on Intelligent Robotics and Systems (IROS 1997)*, volume 2, 586–592.

McLachlan, G. J., and Basford, K. E. 1988. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker.

Shatkay, H. 1998. *Learning Models for Robot Navigation*. Ph.D. Dissertation, Brown University.

Stewart, B.; Ko, J.; Fox, D.; and Konolige, K. 2003. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *The Conference on Uncertainty in Artificial Intelligence (UAI 2003)*.

Tapus, A.; Tomatis, N.; and Siegwart, R. 2004. Topological global localization and mapping with fingerprints and uncertainty. In *The International Symposium on Experimental Robotics (ISER 2004)*.

Tenenbaum, J. B.; de Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290.

Thrun, S. 1998. Learning maps for indoor mobile robot navigation. *Artificial Intelligence* 99:21–71.

Tomatis, N.; Nourbakhsh, I.; and Siegwart, R. 2003. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems* 44:3–14.

Weng, J., and Chen, S. 2000. Visual learning with navigation as an example. *IEEE Intelligent Systems* 15(5):63–71.