# Machine Learning Performance in Automated Manufacturing Systems for Testing Sonobuoy Sub-assemblies


**Roy L Johnson III**
Sparton Defense and Security Systems
5612 Johnson Lake Road
DeLeon Springs, FL. 32130
rjohnson@sparton.com



## Abstract

The performance of machine learning methods are evaluated using data collected during the manufacture of calibrated omnidirectional, (CO), hydrophone sensors, one of several sensors incldued in passive sonobuoys such as the AN/SSQ-53F. The work presented in this paper focuses on the results observed when machine learning methods were applied to actual production data collected during the manufacture of CO sensors as compared to the static pre-existing limits enforced by automated manufacturing test systems. These pre-existing static limits were derived manually based on expected and observed sensor performance during the CO sensor's initial design-engineering cycle. The objective of this analysis is to determine the potential effectiveness of learning algorithims for the development of classification functions capable of accurately distinguishing acceptable sensors from unacceptable sensors in the absence of pre-defined test specifications. Instead we provide a training data set that includes sensitivity readings for sensors that are representative of each classification. The performance of the classifier is then evaluated by allowing the classifier to process previously un-observed data and comparing the results produced with the real-world disposition of the same dataset.


## Introduction

Sonobuoys are primarily used by militaries of the world to detect, locate, and track submerged submarines operating below the surface of the ocean. There are generally four classifications of sonobuoys to include:

- Passive Sonobuoys - operate primarily by measuring ambient acoustic properties with high-gain sensors without introducing any acoustic energy into the environment.
- Active Sonobuoys - operate by introducing acoustic energy into the water and then processing the reflected audio received to localize targets.
- Multi-static Sonobuoys - operate by using a combination of active and passive technologies with the introduction of mutliple passive sensors in array patterns to facilitate beam-forming.
- Special purpose Sonobuoys - perform miscelaneous tasks such as intellegence gathering, and environmental measurement such as salinity and water temperature.

The calibrated omnidirectional hydrophone, often abbrevieated simply as CO, is normally used to measure the sound pressure level of underwater audio emmissions that stimulate the sensor in absolute units of magnitude such as dB per micro-pascal across a specific range of frequencies. This particular sensor is one of the sensors used in the AN/SSQ-53F passive sonobuoy. Due to the requirement that these sensors establish a reference for detected sound pressure levels in the environment to absolute units of measure, the sensitivity of the sensor must be measured and recorded when it is manufactured to achieve the stated goal. Furthermore these sensors must be tested to ensure that their measured sensitivity falls within an acceptal range of values to satisfy the interface requirements of other sonobuoy sub-systems with which it interacts.

The work presented in this paper focuses on the data collected for these sensors specifically as they underwent automated manufacturing test procedures to verify their functionality and acceptable sensitivity levels. More concretely, the focus is specifically to evaluate the performance of a classification function derived as the result of performing a single-feature logistical regression on a training set to establish a boundary plane among CO sensor sensitivity values using a sigmoid function. The resulting factors of the training set are then used in the classification algorithim and applied to data that was not included during the learning operation. Finally the disposition of the sensors as defined by the machine learning method is compared to that of the known sensor classification to measure the effectiveness of this approach compared to the existing methodology.

## Background / Prior Work

The particular technique employed in this work is a classic example of how to approach a univariate or multivariate logistical classification problem. The algorithm choosen to minimize the cost function $J(\theta)$ was gradient-descent defined as

$$J(\theta) = -\alpha \sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

such that all $J(\theta)$ are updated simultaneously. The term $\alpha$ represents the learning rate and the term $y$ represents the actual known classification of the data being used in the training set by the hypothesis function to derive the terms of the vector $\theta$. The hypothesis function $h_\theta(X)$ is based on a sigmoid function applied to some $n$x$m$ sized matrix $X$ containing $n$ features for $m$ data points that are desired to be processed when establishing the decision boundary during the machine learning phase.

$$h_\theta(x) = \frac{1}{1 - e^{-\theta^T x}}$$

The matrix $\theta(X)$ is equivilant to

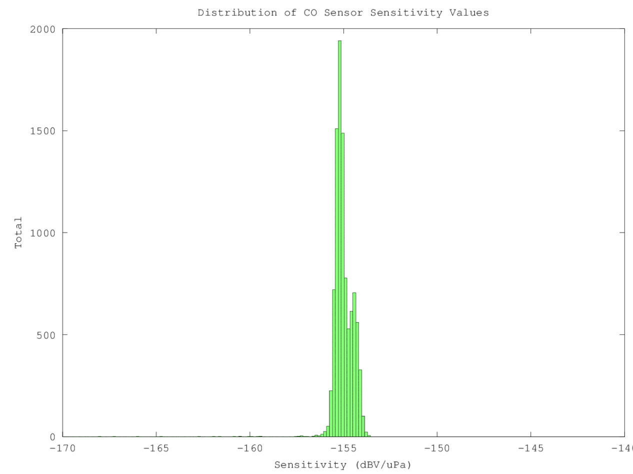
$$\theta(X) = \theta_0 X_0 + \theta_1 X_1 + ... + \theta_n X_n$$

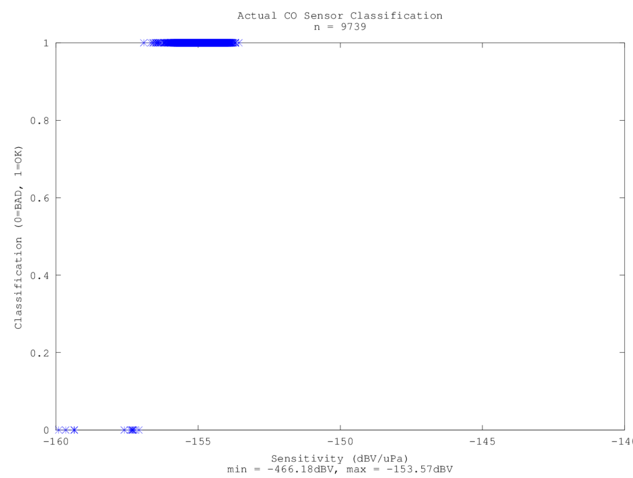
These equations allow predictions for some set $X$ using the hypothesis function $h_\theta(x)$ where the result represents the probability $P(\text{ y } = 1|x;\theta)$ such that when $P >= 0.5$ the prediction can be classified as $\text{ y } = 1$ else $0$.

There are two notable advantages to minimizing the logistical function $J(\theta)$ as described in the gradient descent algorithm employed. The first being that it is derived using the Principle of Maximum Likelyhood Estimation which helps to improve the accuracy of the predictions. The second being that the partial derivatives of the cost function needed to determine it's minima remain convex despite the potential polynomial terms defined in $\theta$ that are in turn being utilized by the cost function which is non-linear.

## Work

A sample data set of 9,739 CO sensor sensitivity values were queried from a manufacturing test database at Sparton DSS in DeLeon Springs, FL., a U.S. government defense contractor that manufactures sonobuoys for the Navy and others, for use in this evaluation. The sample was limited to units that had either passed all automated manufacturing tests or to those that failed the manufacturing test specifically for having a measured sensitivity value that was outside of the processes existing test limits. Sensors evaluated were manufactured and tested from the period of 1-May-2013 through 15-Jun-2013. From this sample 20 records were randomly selected such that 10 represented units with acceptable sensitivity readings and the remaining 10 represented those that did not meet sensitivity standards. This collection of 20 records formed the training set used by the learning algorithm to derive the vector $\theta$. Figure 1 shows a histogram of the sensitivity readings from the entire sample. Figure 2 illustrates the real-world classification of the sample data based on manufacturing test results.




## Method of Evaluation

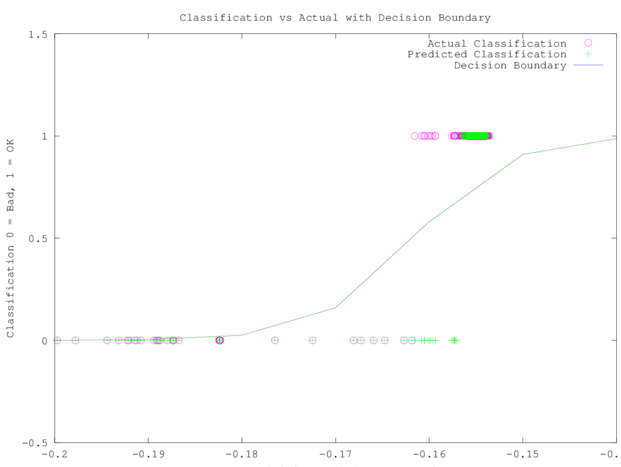
The machine learning application developed to implement the logistic classifier was done in GNU Octave. This tool was used to model the cost function, hypothesis function as well as the gradient descent algorithm to minimize the cost function to derive values for $\theta$. The evaluation started by executing several queries on an Oracle database to get the sample data into a CSV comma-separated value file format. Once the entire sample was obtained in CSV form, the training data was selected at random and placed into a separate csv file.

The Octave function csvread() was utilized to import the raw data as well as the training set. A two-element $\theta$ vector was initialized to 0 and then passed to the logGradDesc() function written by the author for this evaluation to perform the logistical regression on the training set and derive values for $\theta$ which were $32.079$ and $198.471$.

When the logGradDesc() function completed, the next step was to pass the $\theta$ vector to the hypothesis() function implemented to return classification probabilities for the training set used. Initially an arbitrary learning rate $\alpha = 0.1$ was chosen but the logGradDesc() function did not converge well enough as evidenced by the disparity in classification probabilities produced for the training set compared to known classification values. Several iterations of this occurred to search for a reasonable learning rate that produced acceptable classification of the training set as compared to the actual classification values. All work was ultimately performed with an $\alpha = 0.4$. This resulted in 19/20 records in the training set being successfully classified by the hypothesis() function.

Having reasonable $\theta$ values the process of running the hypothesis() function was duplicated using the entire sample of co sensitivity features returning a matrix of classification probabilities. A new row was then added to the matrix based on evaluating the expression $result(1, :) >= 0.5$ which returns a $1$ if true and a $0$ otherwise. Ultimately this data was used in conjunction with the original sample data and known classification to create figure 3. This illustrates the decision boundary defined by the derived terms for $\theta$ when passed to the hypothesis() function in combination with all $x$ possible feature values to plot the sigmoid function. The actual classification is plotted as red circles and the predicted classification is plotted with green plus characters. If a prediction is successful, then ideally all of the plus symbols and circles would overlap leaving none of them on the scatter-plot independently. As evidenced by the figure the machine learning algorithm performed very well based on the size of the training set relative to the test data sample size. It is apparent that the hypothesis() function has classified some records as bad that were in fact good.

## Results




The total sample evaluated was 9,739 units, 103 of which were classified by the manufacturing test system as not meeting CO sensitivity requirements, and 9,636 which did. The result of the work presented here classified 9,654 units as having acceptable CO sensitivity values and 85 that did not.

$$
\begin{aligned}
PredictionError &= \frac{\text{Class1 Count} - \text{Pred Class1 Count}}{\text{Sample Size}} \\
&= \frac{103 - 85}{9739} \\
&= 0.0018
\end{aligned}
$$

## Conclusions and Suggestions for Future Work

There are a number of ways that this work can be continued. There are more efficient algorithms that have been developed for minimizing a logistical cost function that also have the added benefit of not requiring the specification of the learning rate $\alpha$ but instead derive $\alpha$ independently. Examples of such algorithms would include Conjugate gradient, BFGS, and L-BFGS.

Another area of focus would be to prototype the work presented in a manufacturing test system using an application development language such as C or C++. There are libraries that will allow Octave functions to be executed from within C++ but the author of this paper was unsuccessful in getting this to work in the time allotted for this effort. By integrating the learning step into the application the $\theta$ vector could be improved continuously and automatically over time periodically sampling new training data from recorded test results increasing prediction accuracy. A very interesting thing to consider would be including other metrics evaluated by the manufacturing test system and including them in the regression. The Octave code presented and developed for this effort is already capable of $n$ feature regressions subject only to processor and memory limitations of the host device. The implementation is done using vector methods as opposed to itterative techniques for performing the calculations outlined. This would allow for more features to be added to the regression without having to modify any of the Octave functions developed to calculate $\theta$ values or classification probabilities.

## Acknowledgments

I would like to acknowledge my employer, Sparton Defense and Security Systems, 5612 Johnson Lake Road, DeLeon Springs, FL. 32130 for their support in allowing the use of real-world data for this research. The end result may provide a starting point for introducing enhancements to existing applications and processes.

## References

*Ng, Andrew*
Machine Learning. 2013. videos. Coursera.org Web. 24 Jul 2013. ¡http://www.coursera.org/course/ml¿.

*Russell, Norvig*
2003. Artificial Intelligence: A Modern Approach (2 ed.). Pearson Education.

*Yuan, Ya-xiang*
"A modified BFGS algorithm for unconstrained optimization." IMA Journal of Numerical Analysis 11.3 (1991): 325-332.

*Shewchuk, Jonathan Richard.*
"An introduction to the conjugate gradient method without the agonizing pain." (1994).

*Michie, Donald, David J. Spiegelhalter, and Charles C. Taylor.*
"Machine learning, neural and statistical classification." (1994).