

Multidisciplinary Modules on Sensors and Machine Learning

Abhinav Dixit, Arizona State University

I am a PhD student at School of Electrical, Computer and Energy Engineering at Arizona State University. My research interest includes early detection of neurological diseases through irregularities in speech. I also work as a Research Assistant at SenSIP Center, ECEE at ASU. I am currently involved in developed of JDSP HTML5, an interactive DSP software developed in HTML5.

Mr. Uday Shankar Shanthamallu, Arizona State University

"I received my B.S degree in Electronics and Communications from the National Institute of Engineering, India in 2011. I am currently pursuing my Master's and PhD program in Electrical Engineering at Arizona State University(ASU). I am advised by Dr. Andreas Spanias. I joined Sensor, Signal and Information Processing Center (SenSIP) at ASU in Jan 2016. My research interests lie at the overlap of sensors and Machine learning and Big Data including, but not limited to Pattern recognition and Anomaly detection. In summer 2016, I did a summer internship at NXP Semiconductors where I worked on sensor data analytics for anomaly detection. I worked on integrating machine learning algorithms on an embedded sensor systems for Internet of Things applications, which can identify anomalies in real time. Before joining ASU, I worked as Systems engineer for 4 years at Hewlett Packard Research and Development, Bangalore, India."

Prof. Andreas S. Spanias, Arizona State University

Andreas Spanias is a professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University. He is also the founder and director of the SenSIP industry consortium. His research interests are in the areas of adaptive signal processing, speech processing, and audio sensing. He and his student team developed the computer simulation software Java-DSP (J-DSP - ISBN 0-9724984-0-0). He is author of two text books: Audio Processing and Coding by Wiley and DSP; An Interactive Approach. He served as associate editor of the IEEE Transactions on Signal Processing and as General Co-chair of IEEE ICASSP-99. He also served as the IEEE Signal Processing vice-president for conferences. Andreas Spanias is co-recipient of the 2002 IEEE Donald G. Fink paper prize award and was elected Fellow of the IEEE in 2003. He served as distinguished lecturer for the IEEE Signal processing society in 2004.

Sunil Rao

Ms. Sameeksha Katoch, Arizona State University

Dr. Mahesh K. Banavar, Clarkson University

Mahesh K. Banavar is an assistant professor in the Department of Electrical and Computer Engineering at Clarkson University. He received the BE degree in Telecommunications Engineering from Visvesvaraya Technological University, Karnataka, India in 2005, the MS degree and the PhD degree, both in Electrical Engineering from Arizona State University in 2007 and 2010, respectively. His research areas is Signal Processing and Communications; and STEM education.

Mr. Gowtham Muniraju, Arizona State University

received the B.E degree in Electronics and Communication engineering from Visvesvaraya Technological University, Karnataka, India, in 2016 and the M.S degree in Electrical engineering from Arizona State University, Tempe, AZ, USA, in 2018.

Currently pursuing the Ph.D. degree with the School of Electrical, Computer and Energy Engineering, Arizona State University. Research interests include distributed sensor networks, machine learning, deep learning, convex and non-convex optimization.

Jie Fan, Arizona State University



Currently, I am working as a Ph.D. candidate in SenSIP lab at ASU.

I received my M.S. degree in electrical engineering-communications from school of electrical, computer and energy engineering at Arizona State University in Oct. 2016. Also the B.S. degree in electrical engineering is received from Huazhong University of Science and Technology (HUST) in May 2014.

In my master thesis, I developed a novel signal recovery algorithm for sensor array with failures. Comparing to conventional approaches based on interpolation and neural networks, our method requires no priori knowledge from the failure conditions and provides high success rate of recovering the missing data.

My current research is mainly about graphic digital signal processing (GDSP). I focus on applying DSP operators to graphic data applications, such as data classification and data denoising. Those GDSP based methods are very suitable for dataset with complex graphic structure. Comparing to traditional machine learning algorithms, such methods may also provide better performance in success rate of estimation.

Dr. Photini Spanias, Arizona State University

Photini Spanias is Principal Lecturer at the Mary Lou Fulton Teachers College at Arizona State University. She is teaching math methods classes. Her research interests are in math methods and in teacher preparation. She is also interested in online education research.

Andrew Strom, Corona Del Sol High School

Andrew Strom has been teaching mathematics at Corona Del Sol for 21 years. He has taught a variety of subjects: Algebra 1-2, Geometry, Algebra 3-4, Honors Algebra 3-4, Pre-Calculus, Honors Pre-Calculus, College Mathematics and AP Statistics. Andrew enjoys the beauty of mathematics and loves working with young people.

Prof. Constantinos Pattichis

Huan Song

Multidisciplinary Modules on Sensors and Machine Learning

Abstract – Integrating sensing and machine learning is important in elevating precision in several Internet of Things (IoT) and mobile applications. In our Electrical Engineering classes, we have begun developing self-contained modules to train students in this area. We focus specifically in developing modules in machine learning including pre-processing, feature extraction and classification. We have also embedded in these modules software to provide hands-on training. In this paper, we describe our efforts to develop an online simulation environment that will support web-based laboratories for training undergraduate students from Electrical Engineering and other disciplines in sensors and machine learning. We also present our efforts to enable students to visualize and understand the inner workings of various machine learning algorithms along with descriptions of their performance with several types of synthetic and sensor data.

Introduction

Sensors are part of everyday life and they have been integrated on cell phones, internet of things (IoT), automation and security systems. A smartphone has on average, more than 14 embedded sensors that acquire various types of natural data. With the widespread deployment of sensors, we are faced with the challenge to process and interpret vast amounts of data. In fact, data acquisition, processing and transmission created increased requirements for storage, networking [1, 2], and bandwidth and cyber security. The increase of data at the edge creates needs for information extraction at various levels and resolution. Machine learning [3] promises to provide solutions for compression, parameterization and interpretation of sensor data. Various levels of the system are employed to learn from the sensor data and further to carry out a predefined set of objectives such as classification or clustering of the data.

In this paper, we describe the development of an online learning environment that supports modules and laboratories for training undergraduate students in multiple disciplines in sensors and machine learning. This project is part of an NSF IUSE module development grant and describes a variety of sensor systems, their properties, and the process of interpreting signals using classification algorithms. We have developed machine learning functions to support learning modules in our award-winning J-DSP HTML5 [4] online environment. The theoretical content of some of the modules is described in another paper [28]. The software content relating specifically to machine learning is described here. The tool described here is capable of capturing raw sensor data and extracting features from the data. The machine learning algorithms are employed to cluster or classify the extracted features in order to obtain an interpretation of the type and quality of sensor signals captured. The structure of recently developed J-DSP HTML5 has been described in [5].

The paper organization is as follows. We first give a description of feature extraction from raw sensor data and provides details on the compaction properties of principal components. We then discuss basic machine learning methods, including the K-means algorithm for clustering, support vector machines for a two-class classification, and *multilayer perceptron* or artificial neural network. Then we describe the architecture of our software and explain its use in our undergraduate and graduate classes. Descriptions of laboratory exercises and their assessment are also discussed. also In addition, we provide the effectiveness of different algorithms for several types of data distributions. Finally, we describe the delivery of these materials as modules which are customized for use at several levels including senior high school classes, undergraduate level, and continuing education short courses for practitioners.

Feature Extraction from Sensor Data

Feature extraction [6, 7] from raw sensor data is essential in order to obtain values or features which are informative and non-redundant. Extracted features provide useful “compressed” information about the physics of sensor operation. In a typical sensor signal processing framework [3], the first step involves acquiring the raw sensor signal from the sensor and some form of pre-processing such as filtering and denoising is performed. Then the sensor signal is analyzed using window functions to extract meaningful features. This ensures a great reduction in the amount of the sensor data since the number of features extracted using the window is much smaller than the number of raw sensor signal samples. A variety of features can be extracted, some can be as simple as mean, standard deviation, zero crossing rate, and some can be complex features such as entropy, Fourier coefficients, and wavelet coefficients. Of course, not all features could be useful for a required task and hence feature selection is necessary.

Machine Learning

Machine learning (ML) can be used to organize, compress, interpret and classify data. ML has been applied to computer vision [8], speech applications [9], artificial intelligence [10], autonomous vehicles, IoT, energy systems, recommendation systems [11], online advertisements, healthcare and diagnosis [12], and finance and accounting [13]. Although machine learning includes a multitude of paradigms, it can be broadly classified into supervised and unsupervised types. Supervised machine learning algorithms have access to “ground truth” or true labels in addition to data whereas unsupervised algorithms have access only to data.

The K-means Algorithm

Clustering [14], using an unsupervised method, is the simplest way to group the data based on statistical properties and similarities. The K-means algorithm is an iterative algorithm which clusters the dataset into K separate groups. It performs clustering by reducing the variance of each group so that after the final clustering has K groups with equal variances. Each of these K

groups is represented by the mean of samples from that group. The mean is commonly known as the *Centroid* of the group. Figure 1 shows a well clustered dataset with centroids shown explicitly. The K-means is an iterative algorithm which repeatedly performs two stages until convergence. The two stages are E step (expectation) and M step (Maximization) of EM algorithm [15]. A detailed explanation of the same is given in [16].

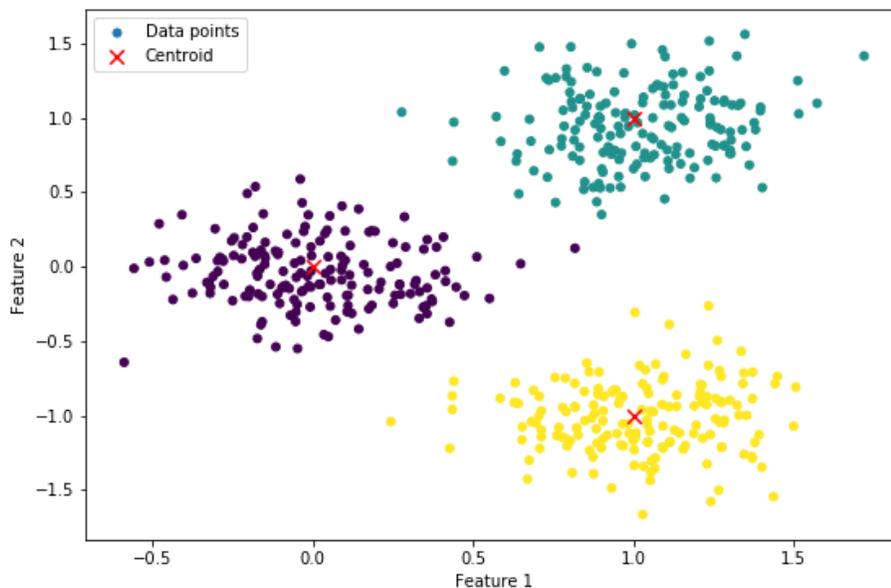


Figure 1. The K-means algorithm produces 3 clusters after convergence. Respective centroids are shown for each cluster.

Software Architecture and Description

J-DSP HTML5 [5] is an online laboratory system, developed for web-based DSP simulations. In this project, we make use of object-oriented code written in JavaScript to develop and host a number of machine learning functions. ML is implemented here in a block-based programming environment that is very intuitive for running the simulations. The front end of the software is developed in HTML5. This is equivalent to an elaborate interactive website, which provides access to the online simulation environment through the browser. Students can perform various ML exercises and web-based laboratories using this system. Upon exiting the browser the current session of the laboratory is closed. The back-end, that is, the processing part, has been developed in JavaScript. The online software has various DSP blocks to carry different DSP functions such as up-sampling, down-sampling, filtering, FIR/IIR filter design, speech processing and several others (jdsp.asu.edu). This new and more secure J-DSP HTML5 includes blocks for plotting the signals to visualize the frequency response, spectrograms, pole-zero plots, and time domain plots. A variety of operations on signals are performed as the signals pass through distinct DSP blocks.

Users can perform sophisticated DSP operations including adaptive algorithms by simply connecting different blocks in a specified algorithmic manner [4]. An implementation of filtering using IIR filter design block in JDSP-HTML5 is shown in Figure 2.

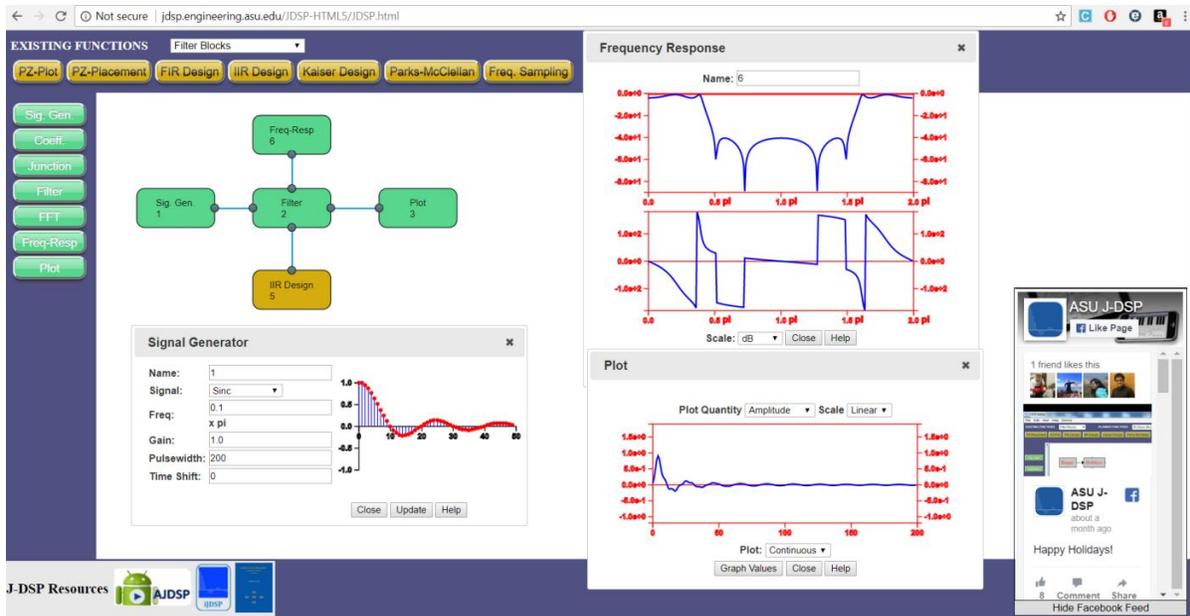


Figure 2. IIR Filter design and implementation in JDSP-HTML5. The filter block is used to change the design parameters of the IIR filter, and sinc function is used as an input for filtering. Note that this new HTML 5 is more secure and provides several new features and access to social nets.

The K-means block performs the clustering of data passed to it using the K-means algorithm [17]. The data from the Dataset block is passed to the K-means block. Figure 3 shows a simple block diagram for running K-means algorithm on a dataset.

The K-means plot displays the clusters in different colors. The algorithm runs iteratively, and the results of each iteration can be seen while the algorithm runs in the background. The clusters are updated after each iteration and the colors of the data points change according to the cluster they are assigned to. A student can visualize the whole convergence as an animation. It also marks the centroids for each cluster that get updated after each iteration.

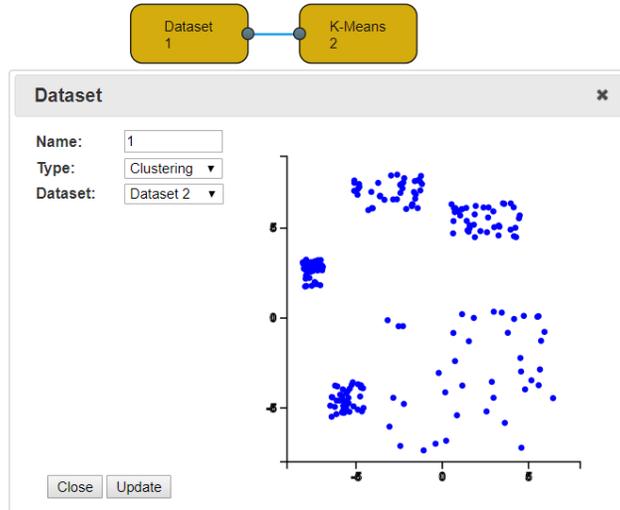


Figure 3. K-means algorithm implemented as a simple block in JDSP-HTML5.

The only parameter that is required from the user as an input is the number of clusters i.e. K . As soon as the calculate button is pressed, the algorithm starts running and the clustering begins. The normalized cost is plotted along with the clusters for each iteration. Once the algorithm runs for all iterations, the values of the final cost and centroids can be viewed. Users can change the value of K and see the performance of the algorithm on the same dataset. The clustering performed by K-means in JDSP-HTML5 is shown in Figure 4.

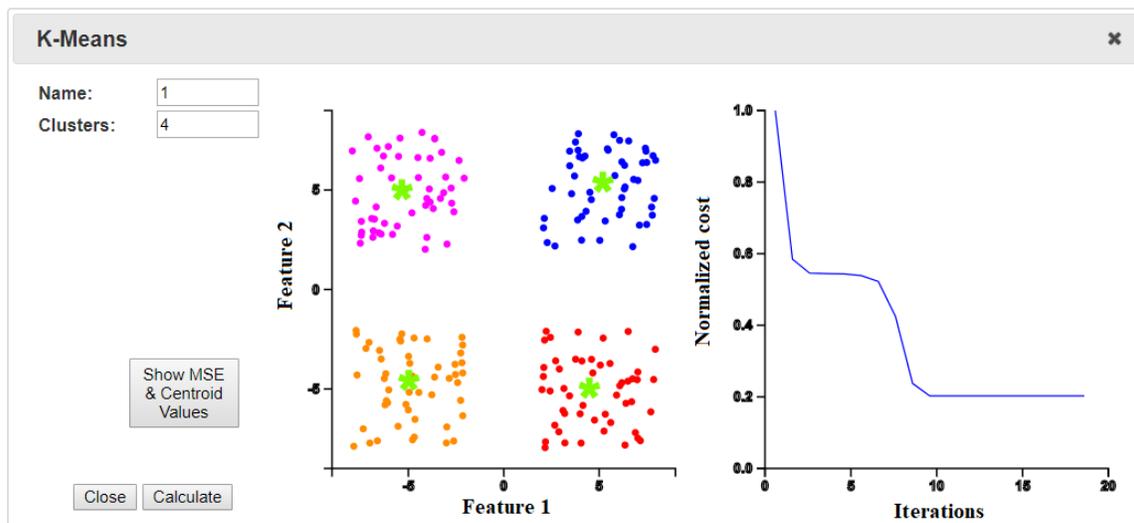


Figure 4. K-Means algorithm implemented in JDSP-HTML5 performed on dataset for 4 clusters. The left plot shows the data points clustered in 4 clusters, and the curve on the right is the convergence curve showing normalized cost w.r.t. to the number of iterations.

From this simulation students can easily learn, visualize, and experiment with signals and implement real-time DSP applications that integrate ML function blocks. Students can see the workings and properties of the K-means algorithm and visualize the convergence with each iteration. They can also change the number of clusters input as a parameter in the K-means block. Through this, they can learn about the working of algorithms for different input parameters. Note that, the centroids in the algorithms are randomly initialized, so students can run the algorithm multiple times and see how different initialization of the initial location of centroids can affect the working and convergence of the K-means algorithm [18-21].

Support Vector Machine (SVM) Algorithms for Classification

We describe here a supervised type of machine learning method, namely, Support Vector Machine (SVM) which can be used for classification. Here the objective is to classify the data sample into a discrete set of classes. Binary SVM is the simplest classification algorithm. It separates positive data examples from negative data examples. For the given dataset consisting of data samples from two distinct classes, binary SVM provides a hyperplane or a boundary of separation which separates positive examples from negative examples. SVM's accuracy is much higher compared to other classification algorithms like k-NN or logistic regression. This is due to the fact that SVM's loss function is convex and provides a global minimum, which happens to be the best possible decision boundary. Geometrically, the decision boundary drawn is as far as possible from either class and hence the SVM algorithm maximizes the margin or the width of separation.

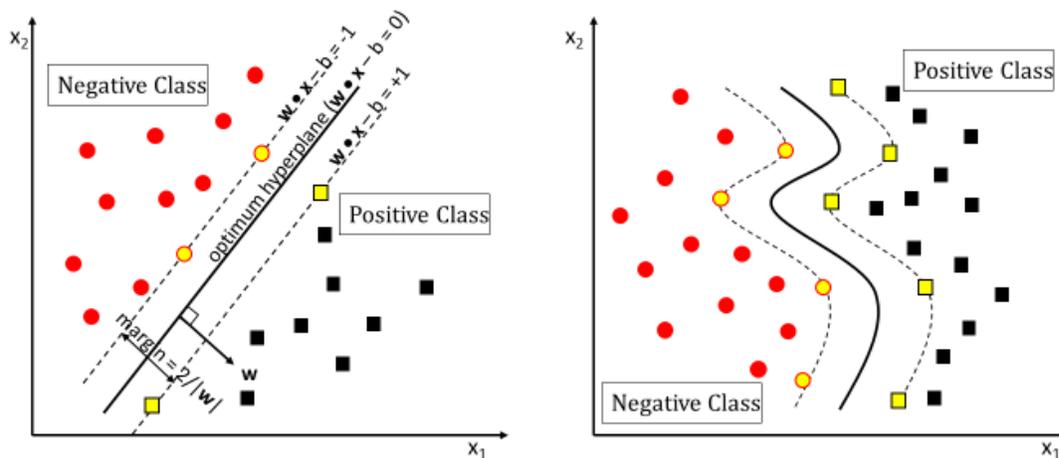


Figure 5. SVM algorithm for binary classification. On the left, maximum margin intuition is shown with a linear decision boundary. On the right, decision boundary for the non-linearly separable data is shown. It is impossible to draw a linear hyperplane for the data on the right. In the figure, red circles represent negative class, and black squares represent positive class. The points in yellow are known as the support vectors.

Figure 5 shows the maximum margin between two classes. Maximum margin separation is not the only advantage of SVM algorithm; it is also capable of providing a non-linear decision

boundary to perform classification on non-linearly separable data. It achieves this task by employing kernels such as Gaussian kernels or radial basis function. This is again depicted in the Figure 5.

In J-DSP HTML5 online environment, the SVM algorithm is available as a block in the existing functions. The user can create a dataset from the Dataset generation block and output of that block can be fed to SVM block for classification as shown in Figure 6. For now, the SVM block performs binary classification. We are adding functionality to enable a multi-class classification (more than 2 classes).



Figure 6. Using SVM algorithm in J-DSP HTML5.

Description of the K-means Exercise used for Speech Formant Clustering

Speech formant estimation [22-24] and clustering was assigned as an undergraduate DSP class project for a cohort of graduate and undergraduate students. The goal of the exercise was to reinforce several concepts in speech processing and to provide an exposition to machine learning and clustering algorithms. For this exercise, speech signal dataset from 100 different speakers (male and female) with extracted formant frequencies F1 and F2 for four different vowels, namely: /i/, /u/, aa, and ae was provided to the students. Generally, vowel formants are represented in a chart known as vowel chart or a vowel diagram. It is a schematic arrangement of the vowels based on first two formant frequencies. Vowel charts usually take the form of a quadrilateral as shown below in Figure 7.

The dataset consisting of the above four vowels for 100 different speakers is readily available in the “Dataset” block available under the “Machine learning blocks” in the existing functions. One can easily add Dataset block and select speech formant dataset and one can also immediately see the plot of F1 and F2 formant frequencies in the dataset block as shown below in Figure 8.

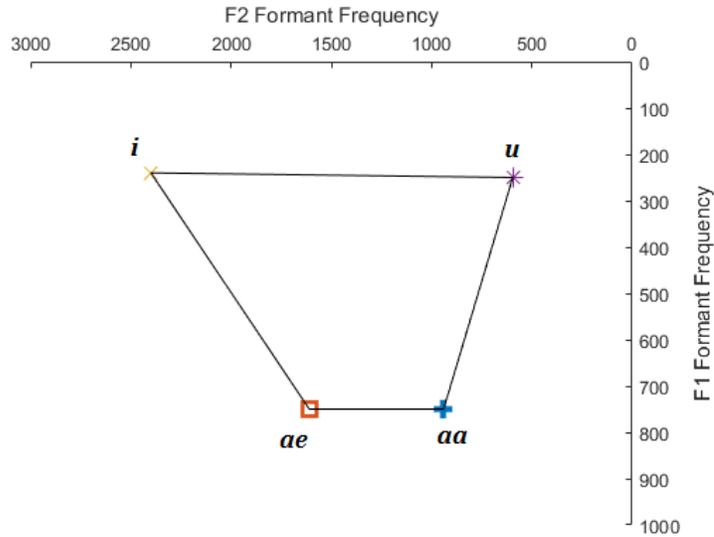


Figure 7. Vowel chart showing four different vowels: aa, ae, i, and u.

The goal of this exercise is to perform clustering on these formants, and obtain the centroids for each cluster. The education objective is to teach students the concepts and inner working of unsupervised machine learning algorithm and to also expose them to a practical application in the area of speech processing.

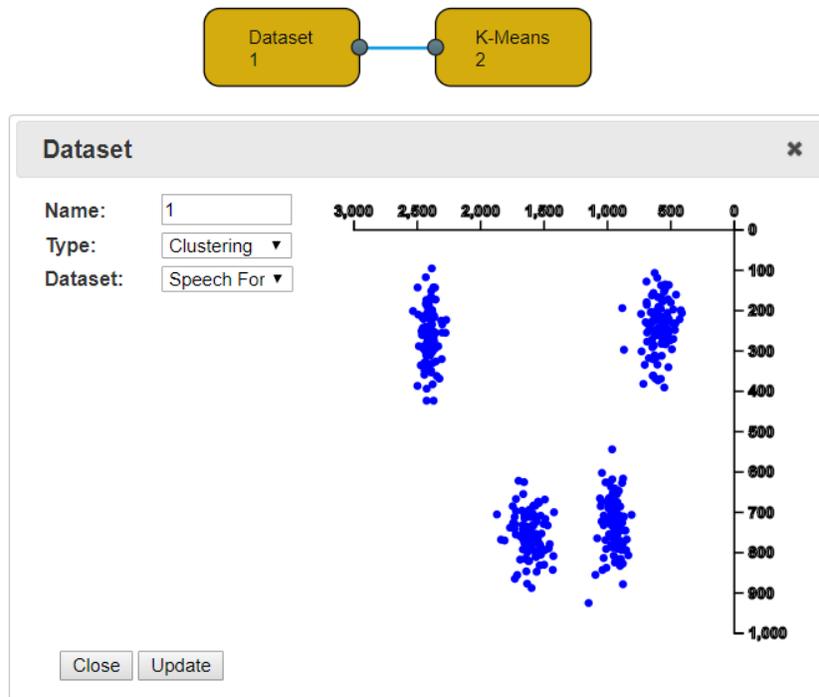


Figure 8. Dataset generation block showing the speech formant dataset.

Assessment Process

The goal of the project is to enable undergraduate and graduate students to learn and understand a simple yet practical speech processing and machine learning application. After working on this project, students can relate the concepts learned in course work to real life applications. A pre-quiz was given to students before assigning speech formant clustering exercise as a class project; we refer to this quiz as pre-assignment quiz. The assessment questions are based on the K-means algorithm and clustering exercise. Simple multiple-choice questions and True or False based questions were asked. Assessment questions related to K-means clustering scheme and mean square error curves were posed. A post-quiz was also given to the same class after completion of the exercise.

There were 15 questions in the quiz. We have tabulated the pre-assignment quiz and post-assignment quiz results (Fig. 9) according to the number of students who responded correctly. Questions 1-6 were based on speech formants, questions 7-9 were based on linear prediction, and questions 10-15 were on machine learning and k-means clustering. We have seen improvements in some of the questions, particularly those associated with feature extraction. The basic clustering process seems to have been understood reasonably well before the post quiz, and hence the scores did not have significant difference. The exercise is being revisited and expanded to include comparisons and more complex data samples. The assessments will also be adapted to the new exercise that will be disseminated this spring in the DSP class.

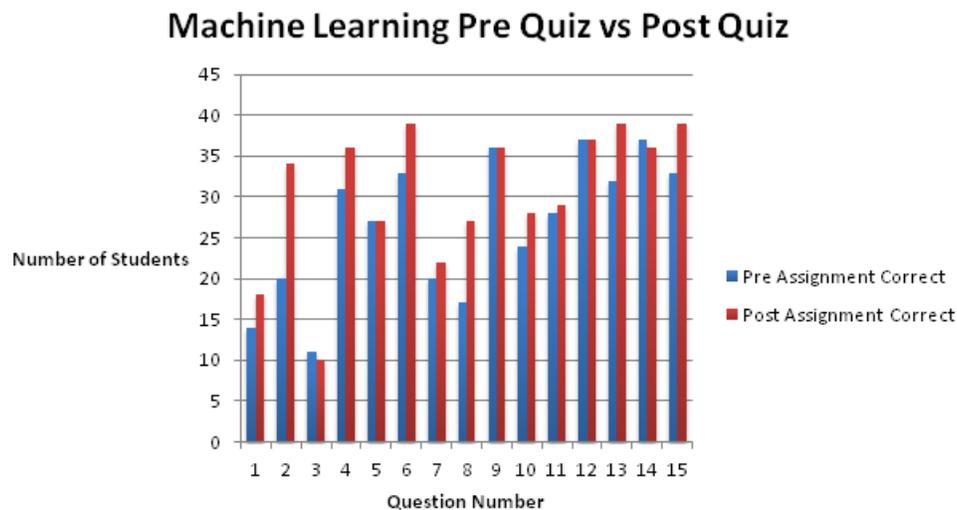


Figure 9. Pre-assignment and post-assignment quiz result. On the x-axis we have the question numbers, and on y-axis we have the number of students who answered the question correctly.

Comparison of Different Clustering Algorithms

Using the online J-DSP HTML5 environment, we can also show the efficacies of different clustering algorithms [25-27] and provide a comparison of different clustering algorithms on several types of data distribution. Although the K-means algorithm is the simplest clustering

algorithm, it does not take into consideration the actual data distribution. It works based on a similarity metric alone. Other algorithms like Gaussian mixture modeling and spectral clustering perform with a higher efficiency as they take the data distribution into account. In Figure 10, three different datasets from different distribution are shown.

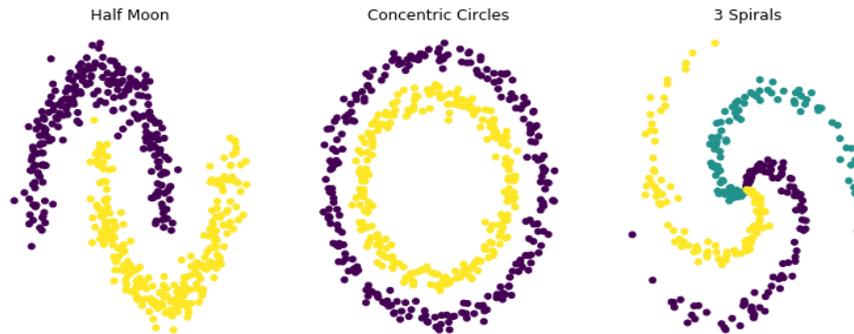


Figure 10. Three different datasets having different types of distribution.

When the K-means algorithm is applied on these datasets, it fails, and clustering efficiency is not as high as the spectral clustering or Gaussian Mixture modeling. Through this comparison the students learn that the underlying data distribution is important before choosing a clustering algorithm.

Conclusion

In this paper, we describe the JDSP-HTML5 machine learning functions that were developed for use in the undergraduate DSP class. Using the JDSP-HTML5, one can perform various DSP operations and apply machine learning functions on different datasets. We have outlined different ways of extracting features from the raw sensor signal. We discussed a variety of paradigms in machine learning and described K-means clustering algorithm in some detail. Support vector machines were also described. We then discussed the K-means exercise for speech formant clustering which used the JDSP-HTML5 online environment. Finally we provided some initial assessments. The project was received well by the class as evidenced by assessments and interviews. Additional functions and exercises are being developed and deployed in our DSP class. Details on assessments of the new functions deployed this spring in our DSP class will be reported at the conference.

Acknowledgement: This research is funded in part by the NSF IUSE program.

References

Material Type	Works Cited
Journal article	[1] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," <i>IEEE Transactions on Knowledge and Data Engineering</i> , vol. 26, pp. 97–107, Jan 2014
Conference paper	[2] S. Zhang, J. Lee, C. Tepedelenlioglu and A. Spanias, "Distributed Estimation of the Degree Distribution in Wireless Sensor Networks," <i>IEEE Global Communications Conference</i> , pp. 1-6, Dec. 2016.
Conference paper	[3] U. Shanthamallu, A. Spanias, C. Tepedelenlioglu, M. Stanley, "A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications," <i>Proceedings 8th International Conference IEEE IISA</i> , Larnaca, August 2017.
Journal article	[4] A. Spanias, V. Atti, A. Papandreou-Suppappola, K. Ahmed, M. Zaman, T. Thrasyvoulou, "On-line signal processing using J-DSP", <i>IEEE Trans. Signal Process. Lett.</i> , vol. 11, no. 10, pp. 1-5, Oct. 2004.
Conference paper	[5] A. Dixit, S. Katoch, P. Spanias, M. Banavar, H. Song, A. Spanias, "Development of Signal Processing Online Labs using HTML5 and Mobile platforms," <i>IEEE FIE 2017</i> , Indianapolis, October, 2017.
Conference paper	[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," <i>2005 IEEE CCPR</i> , pp. 886-893, San Diego, 2005.
Chapter in book	[7] Nixon, Mark S., and Alberto S. Aguado. <i>Feature extraction & image processing for computer vision</i> . Academic Press, 2012.

Conference paper	[8] Rosten, Edward, and Tom Drummond. "Machine learning for high-speed corner detection." In <i>European conference on computer vision</i> , pp. 430-443. Springer, Berlin, Heidelberg, 2006.
Conference paper	[9] Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. "Emotions from text: machine learning for text-based emotion prediction." In <i>Proceedings of the conference on human language technology and empirical methods in natural language processing</i> , pp. 579-586. Association for Computational Linguistics, 2005.
Conference paper	[10] Sebastiani, Fabrizio. "Machine learning in automated text categorization." <i>ACM computing surveys (CSUR)</i> 34, pp. 1-47, 2002
eJournal (from internet)	[11] Pazzani, Michael J., and Daniel Billsus. "Content-based recommendation systems." In <i>The adaptive web</i> , pp. 325-341. Springer, Berlin, Heidelberg, 2007.
eJournal (from internet)	[12] Magoulas, George D., and Andriana Prentza. "Machine learning in medical applications." In <i>Advanced Course on Artificial Intelligence</i> , pp. 300-307. Springer, Berlin, Heidelberg, 1999.
Conference paper	[13] H. Zan, H. Chen, C. Hsu, W. Chen, and S. Wu. "Credit rating analysis with support vector machines and neural networks: a market comparative study." <i>Decision support systems</i> 37, no. 4, pp. 543-558, 2004.
Chapter in book	[14] Aggarwal, Charu C., and Chandan K. Reddy, eds. <i>Data clustering: algorithms and applications</i> . CRC press, 2013.
Chapter in book	[15] McLachlan, Geoffrey, and Thriyambakam Krishnan. <i>The EM algorithm and extensions</i> .

	Vol. 382. John Wiley & Sons, 2007.
eBook	[16]C. M. Bishop, "Mixture Models and EM," in <i>Pattern Recognition and Machine Learning</i> , Springer, New York, pp. 423-430, 2006.
Journal article	[17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. "An efficient k-means clustering algorithm: Analysis and implementation", <i>IEEE Transactions on Pattern Analysis and Machine Intelligence</i> , 2002
Conference paper	[18] G. Anagnostopoulos, M. Georgiopoulos, K. Ports, R. Samuel, M. White, V. Kepuska, P. Chan, P., A. Wu, M. Kysilka, M. , "Engaging Undergraduate Students In Machine Learning Research: Progress, Experiences And Achievements Of Project Emd Mlr" <i>2006 Annual Conference & Exposition</i> , Chicago, Illinois, June 2006
Conference paper	[19] L. Zapata Rivera, M. Larrondo-Petrie, "An Initial Study Applying Data Analysis and Machine Learning to Analyze Dissertations and Theses in the Engineering Education Field," <i>ASEE Annual Conference & Exposition</i> , Columbus, Ohio, 2017.
Conference paper	[20] Y. Shibberu, Y., " <i>Introduction to Deep Learning: A First Course in Machine Learning</i> ", <i>ASEE Annual Conference & Exposition</i> , Columbus, Ohio,2017.
Conference paper	[21] H. Song, J. J. Thiagarajan, P. Sattigeri, K. Ramamurthy, and A. Spanias. "A deep learning approach to multiple kernel fusion." <i>IEEE ICASSP-2017</i> , pp. 2292-2296. IEEE, 2017.
Journal article	[22] A. Spanias, "Speech coding: A tutorial review." <i>Proc. of the IEEE</i> , 82, no. 10, pp.1541-1582,1994

Chapter in book	[23] Andreas Spanias, Ted Painter, Venkatraman Atti, Audio Signal Processing and Coding, Hardcover 544 pages, ISBN: 9780471791478 and 0-471-79147-4, Wiley, Textbook with theory, problems, and MATLAB computer exercises, March 2007.
Chapter in book	[24] Spanias, Digital Signal Processing; An Interactive Approach – 2nd Edition , 403 pages, Textbook with JAVA exercises, ISBN 978-1-4675-9892-7, Lulu Press On-demand Publishers Morrisville, NC, May 2014.
Conference paper	[25] Eick, Christoph F., Nidal Zeidat, and Zhenghong Zhao. "Supervised clustering-algorithms and benefits." In <i>Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on</i> , pp. 774-776. IEEE, 2004.
eJournal (from internet)	[26] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. "Unsupervised learning." In <i>The elements of statistical learning</i> , pp. 485-585. Springer, New York, NY, 2009.
eJournal (from internet)	[27] Ben-Hur, Asa, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. "Support vector clustering." <i>Journal of machine learning research</i> 2, pg. 125-137, Dec 2001.
Conference paper	[28] A. Spanias, M. K. Banavar, H. Braun, P. Spanias, Y, Zhang, "Development of course modules for multidisciplinary STEM education," <i>IEEE FIE 2016</i> , Erie, Oct. 2016.