COASTAL CAROLINA UNIVERSITY

C-SURF Technical Report

RESEARCH ARTICLE
Summer 2018

Classification and Predictive Modeling of Oceanographic Data using Data Mining Techniques

C. L. Chang[1] and O. Abuomar[2]

[1]Departments of Applied Science and Biology, College of William & Mary, Williamsburg, VA, USA
[2]Department of Computing Sciences, Coastal Carolina University, Conway, SC, USA

Key Points:

- Machine learning techniques can be applied to high dimensional datasets for ease and depth in visualization and analysis.
- Common features among PCA clusters in the first and second datasets are heading, temperature, latitude, longitude, bottom-tracking error velocity, and bottom-tracking range.
- SVM using the dot product and polynomial kernel functions were most effective at sorting data.

Correspondence to:
C. L. Chang,
clchang01@email.wm.edu

Citation:
C. L. Chang, O. Abuomar (2018), Classification and Predictive Modeling of Oceanographic Data using Data Mining Techniques, *C-SURF Technical Report*, **2**, 53-60.

Received 23 JUL 2018
Accepted TBD

Abstract Oceanographic data from devices such as the Acoustic Doppler Current Profiler (ADCP) used in this study are high in dimension and are intensive in processing and interpretation. Data mining techniques have proven useful in various applications to these types of datasets for ease and depth of data analysis. This study used both unsupervised and supervised machine learning to analyze and model data collected from an ADCP. Principal component analysis (PCA) was applied to reduce the dimensionality of the data for visualization and cluster analysis. The main common features in both datasets included physical and chemical properties, such as temperature, location, and error velocity. Similarities in the common features driving the formation of the clusters show that PCA was able to consistently identify the most important features in the data. Support vector machines (SVM) using various kernel functions and constant values were extremely accurate in organizing the data into classes defined by the transect it was collected from, with the dot product and polynomial kernel functions having the highest classification accuracy overall. These machine learning techniques were successful in the analysis of ADCP data and may be applied to other similar oceanographic datasets in future studies.

1. Introduction

An increase in large datasets with high dimensionality has created a need for more efficient ways to process high volumes of data. Data mining, defined as the "extraction of patterns or models from observed data" [*Goebel and Gruenwald*, 1999], is a combination of classical statistics, computer science, and large-scale data analytics [*Hand*, 2007]. It aids in processing large datasets, discovering new patterns and trends in the data, and classifying data based on the trends found. This method of classification can then be applied to previously unclassified or new data points to sort this data into classes for further analysis. Supervised data mining techniques, such as the use of support vector machines, require the use of a training set of data to design the classifier and a testing set of data to validate that the classifier is functional for unseen samples.

Data mining has been proven useful in the field of oceanography, as the datasets tend to be high in dimension. Previous applications of data mining techniques have been useful on oceanographic data regarding current patterns and phytoplankton density [*Healey*, 1998] as well as ocean temperatures and salinity data [*Huang et al.*, 2007]. Other data mining and machine learning techniques have also been applied to ocean climate indices to find patterns and trends in the time series [*Steinbach et al.*, 2002]. The goal of this project is to discover patterns in the data collected from an acoustic Doppler current profiler, as well as to classify this data into the transects it was collected from based on chemical, mechanical, and physical properties using data mining techniques.

1.1 The Acoustic Doppler Current Profiler

The acoustic Doppler current profiler (ADCP) has been used in several applications to detect ocean currents and circulation [*Ursella and Gacic*, 2001], sediment transport [*Kostaschuk, et al.*, 2005], and zooplankton distribution [*Lorke, et al.*, 2004]. This instrument emits a fixed frequency and

measures the echoes off sound scatterers. These scatterers can be any small particles such as particulate matter, copepods, and euphausiids. It then calculates the velocity in three directions using the Doppler shift, or the difference between the frequency heard from an object that is static and the frequency heard from an object in motion [*Teledyne RD Instruments* 2011]. The ADCP operates under the assumption that the scatterers are moving at the same speed as the water's speed.

The ADCP, manufactured by Teledyne RD Instruments, uses four beams to obtain velocity measurements in three directions: north-south, east-west, and vertical. Each beam is pointed in a different direction to obtain a different velocity component. Vertical velocity is measured twice to obtain an error velocity. In addition, the ADCP uses bottom-tracking capabilities to determine the ship's location and velocity based on the sea floor in addition to current velocity based on scatterers in the water.

However, there are some limitations to the ADCP. Because it operates on sound scatter and detection, any obstructions to the acoustic signals will potentially affect the data. Factors such as the fluid's attenuation and penetration of the acoustic pulse, the concentration of scatterers in the water, and the amount of bubbles near the ADCP can reduce the data quality received by the ADCP [*Flagg et al.*, 1998]. Bubbles can attenuate and scatter both incoming and outgoing signals, thus skewing the data and potentially causing a large spurious shear [*Flagg et al.*, 1998, *New*, 1992]. However, corrections such as the estimating function in Adrian New's study on ADCP data [1992] may be applied to help adjust for the shear and reduce the bias of the data should it occur.

1.2 Machine Learning Techniques

The field of machine learning, a subcategory of data mining, is broad and has many applications. In general, an algorithm is used to extract patterns, behaviors, and trends in the knowledge discovery phase. This algorithm is then used to classify new data into its appropriate categories in the classification phase [*Hall and Smith*, 1998]. Supervised learning involves the use of a training set and a testing set of data. The machine learning techniques used in this study are both supervised and unsupervised.

1.2.1 Principal Component Analysis

Principal component analysis (PCA) is an unsupervised linear feature reduction method that reduces the dimensions of the dataset into a smaller number of principal components that are linear combinations of the original data [*Vanhatalo et al.*, 2017]. The corresponding eigenvalues of the covariance matrix of the dataset represent the variation in each data column [*Bro and Smilde*, 2014]. PCA has been used to reveal new relationships between samples and variables, as well as to reduce large datasets into a more compact version [*Bro and Smilde*, 2014]. This study uses PCA to visualize and analyze the multidimensional data obtained from the ADCP.

1.2.2 Support Vector Machines

Support vector machines (SVM) is a supervised machine learning technique. The algorithm produces a classifier that uses a training set and a testing set to predict classes and label new data. It finds a hyperplane that separates the data samples into the appropriate classes with the maximum margin, or distance of separation [*Tong and Koller*, 2001]. The data closest to the margin are called the support vectors, and points are classified based on which side of the margin and hyperplane they are located. This study uses SVM to classify data based on different physical, chemical, and mechanical properties.

2. Method

2.1 Data Collection

Data was collected using a 300 kHz ADCP (Teledyne RD Instruments) on two cruises by Diane Fribance of the Department of Marine Science at Coastal Carolina University in July 2014 off the coast of McIntosh County, GA. The first cruise covered transects A, B, and C1-C9, with bin size 0.5 meter. The second cruise covered transect C with bin size 1 meter (Figure 1). Bin size refers to the depth intervals at which data was collected. Data collected includes temperature, location coordinates, date and time, pitch, roll, heading, north velocity, east velocity, vertical velocity, error velocity, bottom-tracking displacement, bottom-tracking north, east, vertical, and error velocities.

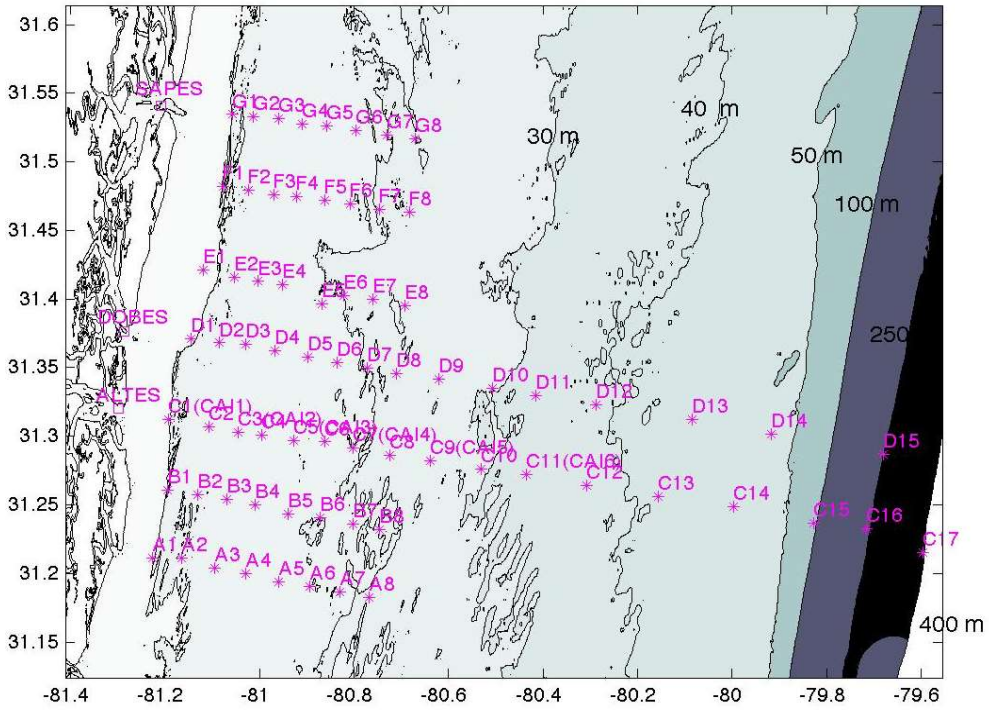


Figure 1. Map of transects covered by Diane Fribance. Data from SAV_14_18001 was taken from transects A, B, and C1:C9. Data from SAV_14_18002 was taken from transect C.

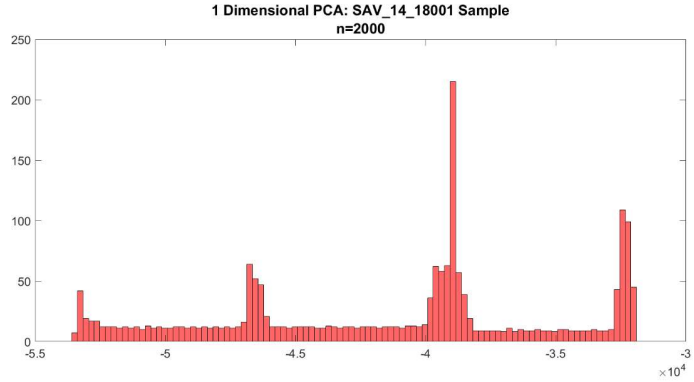
2.2 Data Processing

Both datasets were pre-processed and filtered. Missing data points for north, east, vertical, and error velocity were interpolated using the cubic spline interpolation method within each column (or represented time point), ignoring vector direction. The spline interpolation method utilizes low-order polynomials to estimate missing points in the data. The velocity data was then normalized by dividing each point by the largest value in the column. The average velocity across all the measured depths was obtained to give one velocity measurement per time point. A subset of each dataset (n = 2000) was taken and exported into a spreadsheet.

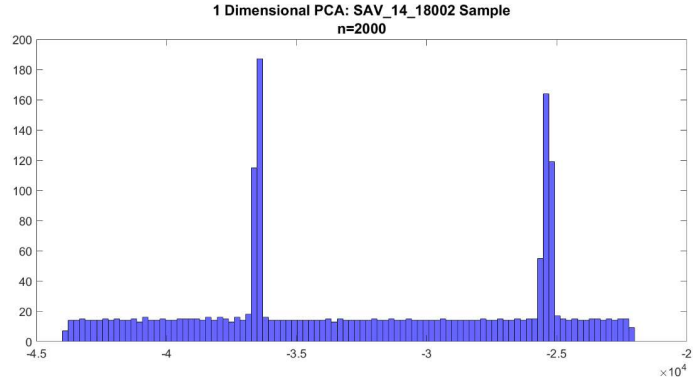
2.3 Application of PCA

To reduce the data into principal components, the covariance matrix was calculated for each data set. Eigenvalues and the corresponding eigenvectors were calculated for this covariance matrix. PCA was applied three separate times to reduce the datasets to one dimension, two dimensions, and three dimensions. To reduce the data set to one dimension, the data matrix was multiplied by the eigenvector corresponding with the smallest eigenvalue. To reduce the data set into two and three dimensions, the data matrix was multiplied by the eigenvectors corresponding to the two and three smallest eigenvalues, respectively. Each multiplication resulted in one dimension of the new reduced data set. The reduced data sets are presented in Figure 2, a-f.

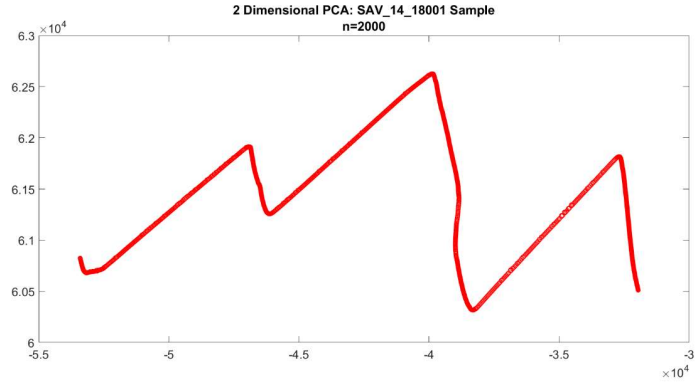
(a)


1 Dimensional PCA: SAV_14_18001 Sample
n=2000
250
200
150
100
50
0
-5.5
-5
-4.5
-4
-3.5
-3
×10^4

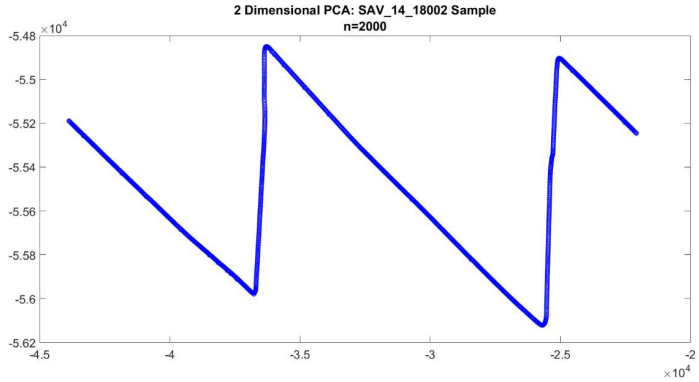
(b)


1 Dimensional PCA: SAV_14_18002 Sample
n=2000
200
180
160
140
120
100
80
60
40
20
0
-4.5
-4
-3.5
-3
-2.5
-2
×10^4

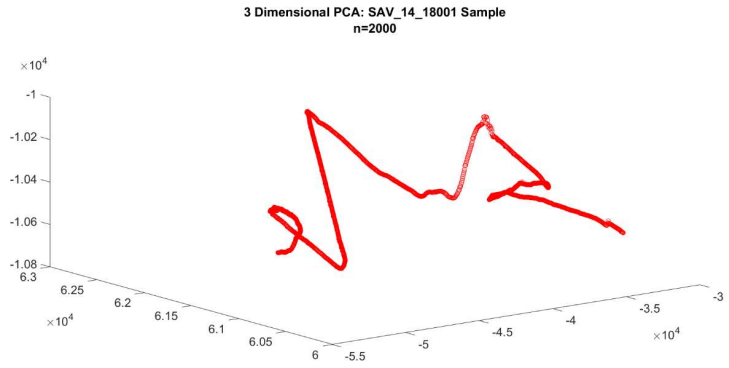
(c)


2 Dimensional PCA: SAV_14_18001 Sample
n=2000
×10^4
6.3
6.25
6.2
6.15
6.1
6.05
6
-5.5
-5
-4.5
-4
-3.5
-3
×10^4

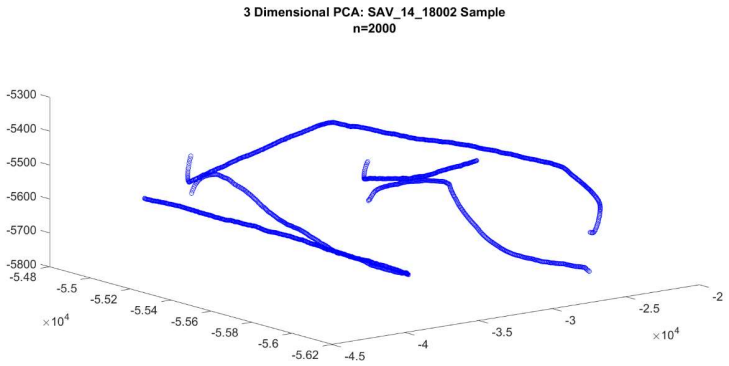
(d)


2 Dimensional PCA: SAV_14_18002 Sample
n=2000
×10^4
-5.48
-5.5
-5.52
-5.54
-5.56
-5.58
-5.6
-5.62
-4.5
-4
-3.5
-3
-2.5
-2
×10^4

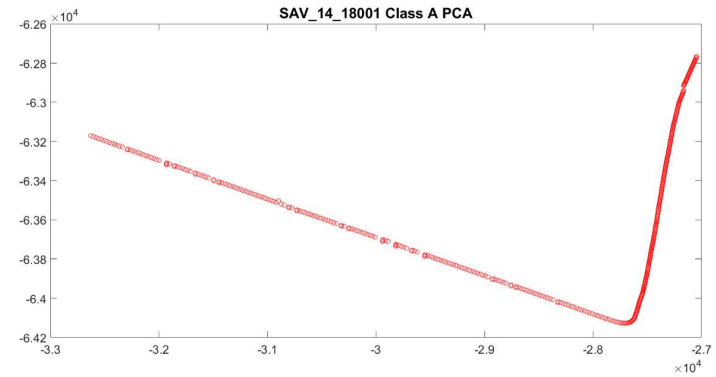
(e)



(f)



(g)



(h)

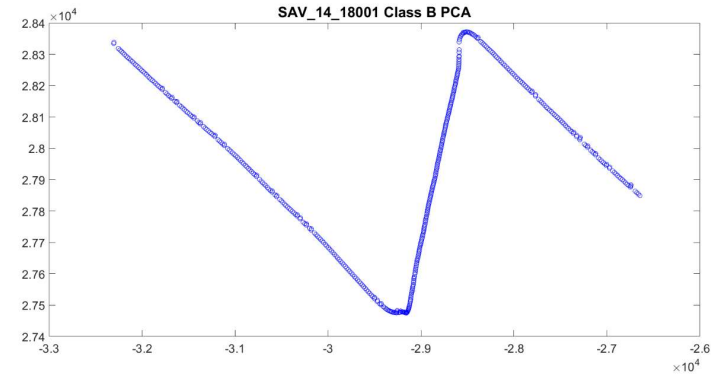


Figure 2. Graphs of PCA results. (a) One-dimensional PCA results for SAV_14_18001. (b) One-dimensional PCA results for SAV_14_18002. (c) Two-dimensional PCA results for SAV_14_18001. (d) Two-dimensional PCA for SAV_14_18002. (e) Three-dimensional PCA for SAV_14_18001. (f) Three-dimensional PCA for

SAV_14_18002. (g) Two-dimensional PCA for Transect A from SAV_14_18001. (h) Two-dimensional PCA for Transect B from SAV_14_18001.

Clusters were observed in the plots resulting from the reduced data. The peaks in the one-dimensional plots (Figure 2a, 2b) and the line segments in the two-dimensional plots (Figure 2c, 2d) are examples of the clusters. A random sample of points (n = 4) was taken from each cluster and their original features and dimensions extracted to analyze and identify the features that are driving the clustering.

Two-dimensional PCA and cluster analysis was also performed on one subset (n = 500) of the data taken from transect A and one subset (n = 500) of the data taken from transect B. The reduced datasets are presented in Figure 2, g-h.

2.4 Application of SVM

The ADCP data was taken and optimized to find the support vectors. The purpose of using SVM is to classify the data and determine which transect a data point was taken from. The sample data that was obtained from the ADCP was not linearly separable into the two different classes for transects, and thus the application of SVMs and kernel functions were needed. SVMs map the data into a higher dimension using a kernel function in the search for the linear hyperplane with the maximum margin, or distance between the hyperplane and the support vectors (Figure 3).

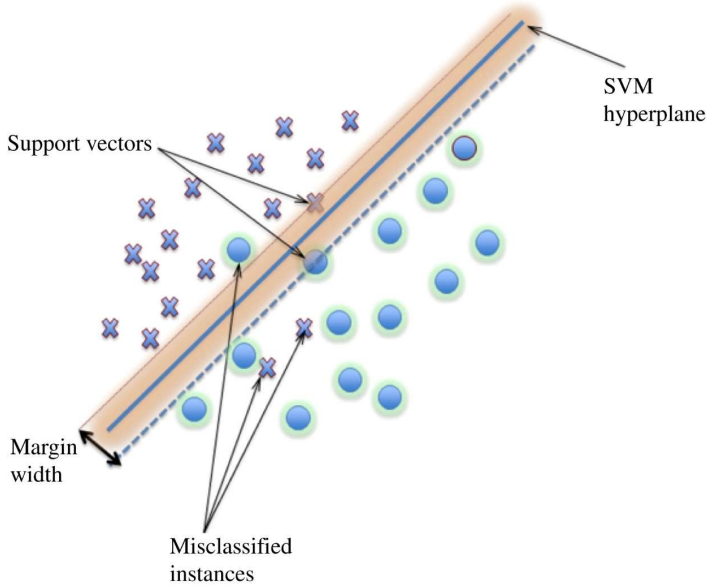


Figure 3. Example of SVMs from Mountrakis, et al. (2011). Let the crosses represent one class and the circles represent a second class. The hyperplane divides the two classes. Data points that lie on the margin (marked by the dashed blue line and the solid orange line) are support vectors.

The equation for the hyperplanes that separate the samples into two classes is:

$$g(\vec{x}) = \vec{\omega}^T \vec{x} + \omega_0 \quad (1)$$

where $\vec{\omega}$ is the set of weights orthogonal to the hyperplane, $\vec{x}$ is the vector of sample data used, and ω_0 is the threshold, or bias. Data points are classified as Class 1 if $g(\vec{x}) > 1$ and Class 2 if $g(\vec{x}) < -1$. Points that lie on the plane are support vectors. z is the distance from the decision hyperplane to the edge of the margin, and is defined as

$$z = \frac{|g(\vec{x})|}{\|\vec{\omega}\|} \tag{2}$$

In order to find the hyperplane with the maximum margin, or the maximum z, a criterion function $J(\vec{a})$ was defined and optimized such that $J(\vec{a})$ is minimized if $\vec{a}$ is the solution vector.

$$J(\vec{a}) = J\big(\vec{a}(k)\big) + \nabla J^T\big(\vec{a} - \vec{a}(k)\big) + \frac{1}{2}\big(\vec{a} - \vec{a}(k)\big)^T H(\vec{a} - \vec{a}(k)) \tag{3}$$

where $\vec{a}$ is a solution vector, $\vec{a}(k)$ is the k^{th} iteration through the solution vectors, and *H* is the Hessian matrix for *J*. The function *J* is then applied to the weights and bias. Since z can be calculated, ω can then be scaled such that $g(x)$=1 at the nearest point for ω_1 and $g(x)$=-1 at the nearest point for ω_2. For each data point $\vec{x}_i$, there should be a corresponding label y_i that is 1 for ω_1 and -1 for ω_2. The goal then became to minimize

$$J(\vec{\omega}, \omega_0) = \frac{1}{2}\|\vec{\omega}\|^2 \tag{4}$$

subject to

$$y_i(\vec{\omega}^T\vec{x} + \omega_0) \geq 1 \tag{5}$$

for *i*=1, 2, …, N, where N is the number of points. To do so, the technique of Lagrangian multipliers was applied to transform the constrained optimization problem to a single unconstrained problem, combining Equations 4 and 5. The Lagrangian function for this optimization is:

$$L\big(\vec{\omega}, \omega_0, \vec{\lambda}\big) = \frac{1}{2}\vec{\omega}^T\vec{\omega} - \sum_{i=1}^{N}\lambda_i[y_i(\vec{\omega}^T\vec{x} + \omega_0 - 1)] \tag{6}$$

where λ_i is the Lagrangian multiplier, $\overrightarrow{\omega}$ is the weight vector, ω_0 is the threshold, or bias, $\overrightarrow{x_i}$ is the data sample vector, y_i is the label associated with that vector, and N is the number of samples.

The systems were placed under Karush-Kuhn-Tucker (KKT) conditions, which state that for the Lagrangian function $L\big(\overrightarrow{\omega},\ \omega_0, \vec{\lambda}\big)$ (Equation 6) the following must be true:

$$\frac{\partial}{\partial\vec{\omega}}L\big(\vec{\omega}, \omega_0, \vec{\lambda}\big) = 0 \tag{7}$$

$$\frac{\partial}{\partial\vec{\omega}}L\big(\vec{\omega}, \omega_0, \vec{\lambda}\big) = 0 \tag{8}$$

$$\lambda_i \geq 0, i = 1, 2, \dots, N \tag{9}$$

$$\lambda_i[y_i(\vec{\omega}^T\overrightarrow{x_i} + \omega_0) - 1] = 0 \tag{10}$$

After applying KKT conditions, the dual Lagrangian function was found to be:

$$L_D(\vec{\omega}, \omega_0, \lambda) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \overrightarrow{x_i}^T\overrightarrow{x_j} \tag{11}$$

subject to

$$\sum_{i=1}^{N}\lambda_i y_i = 0 \tag{12}$$

where N is the number of points and

$$\lambda_i \geq 0 \tag{13}$$

which shows that this optimization is only a function of the support vectors. Quadratic programming was used to complete the optimization and determine the support vectors, the weights for the support vectors, and the bias.

The data points may not be linearly separable at their current dimension. The application of kernel functions to map the data to a higher dimension is used to further separate the data to find a linear decision hyperplane. To determine the best classification performance of SVMs, different kernel functions were applied. The kernel functions used were:

1) Polynomial of degree *Q* where *Q*=2:

$$(\vec{x}^T\vec{z}+1)^Q \tag{14}$$

2) Radial basis function (RBF) where σ^2 is the variance of the data:

$$e^{-\frac{\|\vec{x}-\vec{z}\|^2}{\sigma^2}} \tag{15}$$

3) Dot product:

$$\vec{x}^T\vec{z} \tag{16}$$

4) Hyperbolic tangent function, where β and γ are constants:

$$\tanh(\beta\vec{x}^T\vec{z}+\gamma) \tag{17}$$

where $\vec{x}$ and $\vec{z}$ are data vectors and support vector samples.

With this equation, there are three possible cases and thus three different constraints for each data point. The data point can either be correctly identified on either side of the margin, incorrectly identified on either side of the margin, or identified as being within the margin. To reduce the number of constraints, a slack variable ξ was introduced to the criterion function *J* (Equation 4). This new criterion function

$$J(\vec{\omega},\omega_0,\vec{\xi}) = \frac{1}{2}\|\vec{\omega}\|^2 + c\sum_{i=1}^{N}\xi_i \tag{18}$$

where *c* is a constant, was then minimized subject to:

$$y_i[\vec{\omega}^T\vec{x_i}+\omega_0] \geq 1-\xi_i \tag{19}$$

for *i*=1, 2, …, N, and

$$\xi_i \geq 0 \tag{20}$$

for *i*=1, 2, …, N.

Different constant values (*c* = 0.1, *c* = 10, *c* = 100) were used with the different kernel functions to find the highest classification performance.

The re-substitution method was first used to train the set. A subset of sample points (n = 500) were taken from each class (A for transect A and B for transect B) to form the training set. The trained algorithm was then applied to the data set containing all the data points from transects A and B. Classification accuracy was determined from the produced confusion matrices.

The three-fold cross-validation (CV) method was then used to train the SVM. 700 sample points from each class A and class B were taken to form a dataset. The dataset was split into three folds. For each fold, a different section was left out to be used as the testing set. For the first iteration, sections 1 and 2 were used for training and section 3 was used for testing. For the second iteration, sections 2 and 3 were used for training and section 1 was used for testing. For the third iteration, sections 1 and 3 were used for training and section 2 was used for testing. After the three folds were completed, the classification accuracy was calculated using the produced confusion

matrices. This was performed for both the original high-dimensional dataset, a normalized dataset, and the dataset after applying PCA and reducing the data into 2 dimensions.

The SVM algorithms were then applied to a multiclass set, with 700 data points from each of transects A, B, and C. The one-against-all method was employed, with each class tested against the other classes combined. Confusion matrices were produced for each of the classes and kernel functions.

3. Results

3.1 PCA

The application of PCA and the reduction of data dimensionality into one, two, and three dimensions showed evidence of clustering within the data (Figure 2, a-f). The main features driving the clustering are listed in Table 1. The common features in both datasets were heading, temperature, latitude, longitude, bottom-tracking error velocity, and bottom-tracking range. These features are the most significant features for both of the analyzed ADCP datasets.

Table 1. The main driving features for each data set. Dataset 1 refers to transects A, B, and C1:C9 (from SAV_14_18001). Dataset 2 refers to transect C (from SAV_14_18002). The common driving features for both sets are highlighted in red.

Dataset 1	Dataset 2
Heading	Heading
Temperature	Temperature
Latitude	Latitude
Longitude	Longitude
Bottom-tracking error velocity	Bottom-tracking east velocity
Bottom-tracking range	Bottom-tracking error velocity
	Bottom-tracking range
	Navigational east velocity

In one-dimensional PCA (i.e. reduced to only one principal component), four clusters were evident in the graph for the first dataset, whereas only two clusters were evident in the second dataset. There were consistently more clusters found in the first dataset than in the second for two-dimensional and three-dimensional PCA (i.e. for the cases with two and three principal components).

The main driving features for PCA applied to the transects are listed in Table 2. These features were identical between the two transect sets. The common features in the transects that overlapped with the common features of the first dataset were heading, temperature, latitude, longitude, and bottom-tracking range. This was identical to the common features of the transects and the second dataset.

Table 2. The main driving features for each transect. Note that all the driving features are the same for both transects.

Transect A	Transect B
Heading	Heading
Temperature	Temperature
Latitude	Latitude
Longitude	Longitude
Bottom-tracking range	Bottom-tracking range
North velocity	North velocity

3.2 SVM

3.2.1 Re-substitution Method

The plots of the SVM results for each of the kernel functions with c = 0.1 are presented in Figure 5, with the remainder of the SVM results for the polynomial, dot product, and RBF kernels and confusion matrices presented in the appendix. Due to the low classification accuracy, the hyperbolic tangent kernel results are not included in the appendix of this study. The overall most accurate kernel function was the polynomial function with an average classification rate of 86.08%, followed by the dot product kernel with an average classification rate of 82.72%. The least accurate kernel was the hyperbolic tangent function with an average classification accuracy of 50%. The RBF kernel also had a low average classification rate of 50.25% when applied to the entire dataset.

3.2.2 Three-fold CV Method

The average confusion matrices across the three folds for each of the four kernel functions using the full datasets are presented in Figure 4a. The plots of the most accurate SVM algorithms are presented in Figures 6 and 7, with the remaining confusion matrices and SVM results for the polynomial, dot product, and RBF kernel functions presented in the appendix. With the two-dimensional reduced dataset, all kernel functions had 100% classification accuracy with each of the constant values. With the normalized dataset, the polynomial kernel with c = 0.1 had the highest classification accuracy of 99.91%. With the full unmodified dataset, the dot product kernel with c = 0.1 had the highest classification accuracy of 99.52%. In both the normalized and full datasets, the hyperbolic tangent function had the lowest classification accuracy.

(a)

Radial Basis Function: c = 10

	True Class 1	True Class 2
Classifier 1	311.33	194
Classifier 2	38.67	156

Radial Basis Function: c = 0.1

	True Class 1	True Class 2
Classifier 1	310.67	0
Classifier 2	39.33	350

Radial Basis Function: c = 100

	True Class 1	True Class 2
Classifier 1	311.33	194
Classifier 2	38.67	156

Dot Product: c = 0.1

	True Class 1	True Class 2
Classifier 1	346.67	0
Classifier 2	3.33	350

Dot Product: c = 10

	True Class 1	True Class 2
Classifier 1	346.33	0
Classifier 2	3.67	350

Dot Product: c = 100

	True Class 1	True Class 2
Classifier 1	345.67	0
Classifier 2	4.33	350

Hyperbolic Tangent: c = 0.1

	True Class 1	True Class 2
Classifier 1	0	233.33
Classifier 2	350	116.67

Hyperbolic Tangent: c = 10

	True Class 1	True Class 2
Classifier 1	0	233.33
Classifier 2	233.33	0

Hyperbolic Tangent: c = 100

	True Class 1	True Class 2
Classifier 1	0	233.33
Classifier 2	233.33	0

Polynomial: c = 0.1

	True Class 1	True Class 2
Classifier 1	336	38
Classifier 2	14	312

Polynomial: c = 10

	True Class 1	True Class 2
Classifier 1	350	85.33
Classifier 2	0	264.67

Polynomial: c = 100

	True Class 1	True Class 2
Classifier 1	342	38
Classifier 2	8	312

(b)

Radial Basis Function: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	0
Classifier 2&3	0	1000

Radial Basis Function: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	0
Classifier 1&3	0	1000

Radial Basis Function: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	0
Classifier 1&2	0	1000

Dot Product: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	0
Classifier 2&3	0	1000

Dot Product: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	0
Classifier 1&3	0	1000

Dot Product: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	0
Classifier 1&2	0	1000

Hyperbolic Tangent: Transect A

	True Class 1	True Class 2&3
Classifier 1	0	0
Classifier 2&3	500	1000

Hyperbolic Tangent: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	1000
Classifier 1&3	0	0

Hyperbolic Tangent: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	1000
Classifier 1&2	0	0

Polynomial: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	91
Classifier 2&3	0	909

Polynomial: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	500
Classifier 1&3	0	500

Polynomial: Transect C

	True Class 3	True Class 2&3
Classifier 3	500	310.33
Classifier 2&3	0	689.67

Figure 4. Confusion matrices for the kernel functions and constant values. (a) Average confusion matrices for three-fold cross validation using the full unmodified dataset. (b) Confusion matrices for the multiclass cases.

3.2.3 Multiclass Case

The confusion matrices for the multiclass cases with each of the kernel functions are presented in Figure 4b, and the overall plots presented in Figure 8. The one-against-all plots are presented in the appendix. Both the RBF and the dot product kernels had 100% classification accuracy for all classes. The hyperbolic tangent kernel had 50% classification accuracy for all classes. The polynomial kernel had the highest classification accuracy for class 1, or transect A, and the lowest classification accuracy for class 2, or transect B.

4. Discussion

4.1 PCA and Clustering

The dimensionality reduction of the data set assisted in visualizing the data without eliminating any of the features. The graphs shown reveal that the data tends to cluster into groups with common main features that are likely to be the defining characteristics of the data. The common features driving the clustering are: heading, temperature, latitude, longitude, bottom-tracking error velocity, and bottom-tracking range.

Some of these driving features are determined more by physical properties. Heading refers to the direction that the bow of the ship is pointing and would need to be corrected for in the velocity measurements. Data points with similar values for heading would likely have been taken close to each other as the ship was facing a particular direction and have similar properties. The bottom-tracking error velocity is a calculated value that involves the vertical velocity measurements, and thus may influence clustering. This value helps assess the quality of the data collected at a certain point. Temperature may also be a driving feature, as the temperature of the water can affect its density as well as other physical properties. Changes in density would also potentially change the motion of scatterers and the velocity measurements. Measurements at similar temperatures may be close together as well as have other similar physical properties.

The other driving features are based on location. Latitude and longitude are the location marks of each data point and would likely have a positive clustering effect on the data. Data taken at points close to each other would likely have similar properties based on the continuity of the ocean. Similarly, range, or the depth of the measurement, would be a driving feature. The closer the measurements of the depths are, the more likely that the velocity and other features would be similar. However, temperature, latitude, longitude, and bottom-tracking error velocity all have small ranges and thus may be similar through the clusters as they are similar throughout the datasets in general.

In the PCA applied on transects A and B separately, northward velocity was also a common feature although it was not considered a common feature in the PCA clusters from the first or second datasets. Northward velocity may be more similar within the samples taken from transects A and B than in the sample taken from the first dataset.

The high overlap in the common driving features of the first and second dataset show that PCA is accurately clustering and reducing the features while preserving the effects of the most important features. Since the datasets are similar, the driving features should also be very similar. However, the differences between the driving features in each of the datasets may be caused by the variation in the transects samples and the bin size when data was collected. Transects A, B, and C1:C9 were all sampled in areas with similar depths, as well as had a bin size of 1 m. Transect C was sampled across a range of depths, from on shore to open water, and had a bin size of 0.5 m. Since there was more frequent sampling across depths with transect C, which was recorded into the second dataset (SAV_14_18002), the variation in the data may be higher in the second dataset. This variation could change the way each feature interacts with the others as well as how much effect each feature has on the dataset overall.

4.2 SVM Kernel Functions

4.2.1 Re-substitution Method

In the training and testing phases of the re-substitution method, it is expected for the classification accuracy to be high as the same data is used to train and test the model. In this case, the highest classification accuracy was obtained with the polynomial kernel function for all values of c, ranging from 84.34% to 86.95%. The dot product kernel function also had high classification accuracy for all values of c, ranging from 81.16% to 83.57%. The polynomial kernel had the highest classification

accuracy of 86.95% for both c = 0.1 and c = 100, while the dot product kernel had the highest classification accuracy of 83.57% with c = 10. The success of the dot product kernel suggests that the data is linearly separable.

Both the RBF kernel and the hyperbolic tangent kernel had low classification accuracy for this dataset, ranging from 50% to 50.3%. The data points were placed too closely together, so it is quite difficult for these functions to separate the classes accurately. Most of the points used in the training set were found to be support vectors due to this proximity of the data points.

4.2.2 Three-Fold Cross Validation Method

The highest classification accuracy for all the constant values (c values) resulted from the dot product kernel function, with classification rates ranging from 99.38% to 99.58%. The polynomial kernel function also had high classification rates for all of the constant values, ranging from 87.81% to 93.43%. For c = 0.1, the RBF kernel had a classification rate of 94.38%, although for c = 10 and c = 100, the classification rate dropped down to 66.76% and 65.43%. The hyperbolic tangent kernel function was unable to run for several folds, and thus had classification rates ranging from 0% to 16.67%. This is consistent with the kernel functions and results from the re-substitution method previously used.

The dataset was then normalized within each column to determine if normalizing the data would increase or decrease the classification rates of the SVM for each kernel function. For the dot product, the classification rates overall decreased on the order of 0.1%. The classification rate for the RBF kernel significantly decreased for c = 0.1, but increased for c = 10 and c = 100. The classification rates increased overall for the hyperbolic tangent function to 20.29%. Although there was an increase, the classifier is still not useful with such low classification rates. The polynomial function kernel benefitted the most from normalizing the data. The new classification rates ranged from 96.1% to 99.91%. For the ADCP dataset, the additional step of normalizing the data was only useful for the polynomial kernel function in terms of increasing classification rate. However, classification rates that are 100% or extremely close to 100% may simply be a sign of overtraining the model to only fit this specific dataset. The classifier may work well for this dataset, but it fails to correctly classify other datasets.

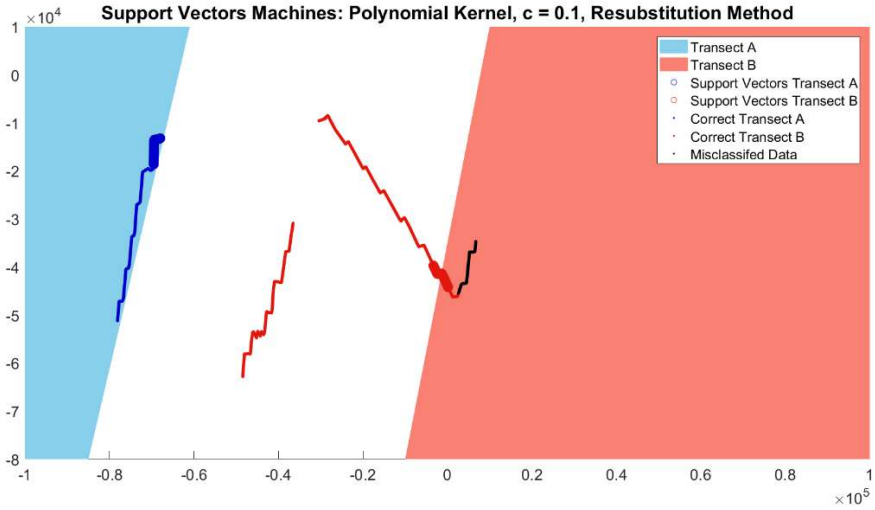
With the dataset reduced to two dimensions from PCA, all the kernel functions had 100% classification accuracy. The additional step of compressing the data may have reduced the clustering of the data and reduced the effects of the less important features. Although the models were able to correctly classify the points, the models may also be over-trained and only be useful in classifying the data in this particular dataset. The accuracy may decrease significantly if given other data points not included in the training or testing sets.

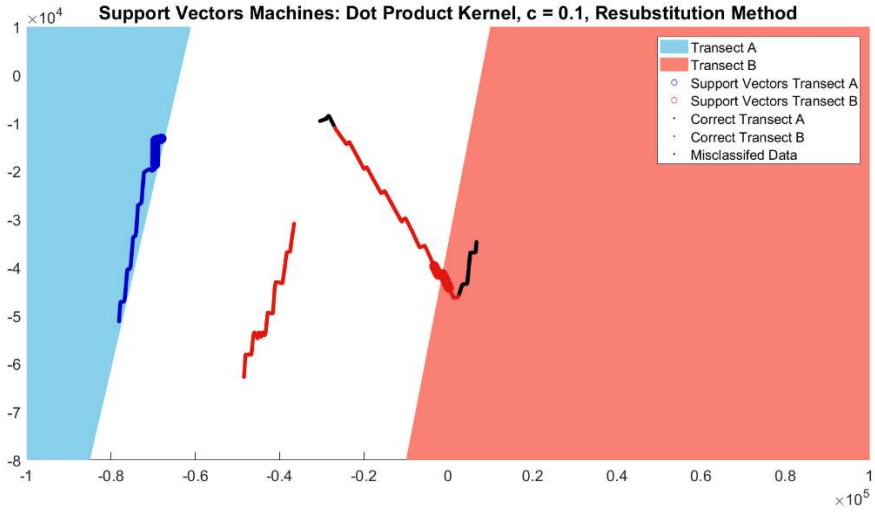
4.2.3 Multiclass SVM Application

The one-against-all strategy was used to design the multiclass classifier. For each run, this classifier separated one class from all of the others. Both the RBF kernel and the dot product kernel had 100% classification accuracy for all classes and all constant values. These kernels were able to evenly and accurately separate each transect or class from the others. The hyperbolic tangent kernel function had 50% classification accuracy for all constant values and classes. This kernel was not effective for separating any of the transects from each other. The polynomial kernel function had varying degrees of accuracy across the different transects. The classification accuracy for transect A was the highest, ranging from 90.45% to 98.15% across the constant values. It was more difficult for the algorithm to separate transect B from the other transects, as the classification rate decreased to 75% across all constant values. This is consistent with the lower classification rates for transect C, which ranged from 75.95% to 96.05%. The data from transects B and C were more similar and thus harder to separate when using the polynomial kernel.

4.3 Kernel Function and Constant Value Analysis

Across the different applications of SVM on all datasets, the dot product kernel and the polynomial kernel had the best classification performance. The data tended to remain close together and clustered, therefore making it difficult to separate. The dot product kernel separates the classes with a straight line, and therefore works well with data that is linearly separable. It is also a special case of the polynomial kernel function in which a line can be considered a polynomial kernel of degree one (i.e. Q = 1). This may help explain the high classification rates and success of both the dot product and polynomial functions. The polynomial kernel function separates data points into different classes using a curved line and has been successful in classifying the data in nearly all applications. The RBF checks the class of each data point individually, but it may be more difficult for this kernel function to distinguish classes when the data is closely clustered together. The hyperbolic tangent kernel had the lowest performance overall. It has a sinusoidal nature and will apply the wave-like function to the data, even though the data does not necessarily match this trend. This kernel function does not raise the dimensionality of the data and may be more useful for data points that initially have more separation and a sinusoidal trend.





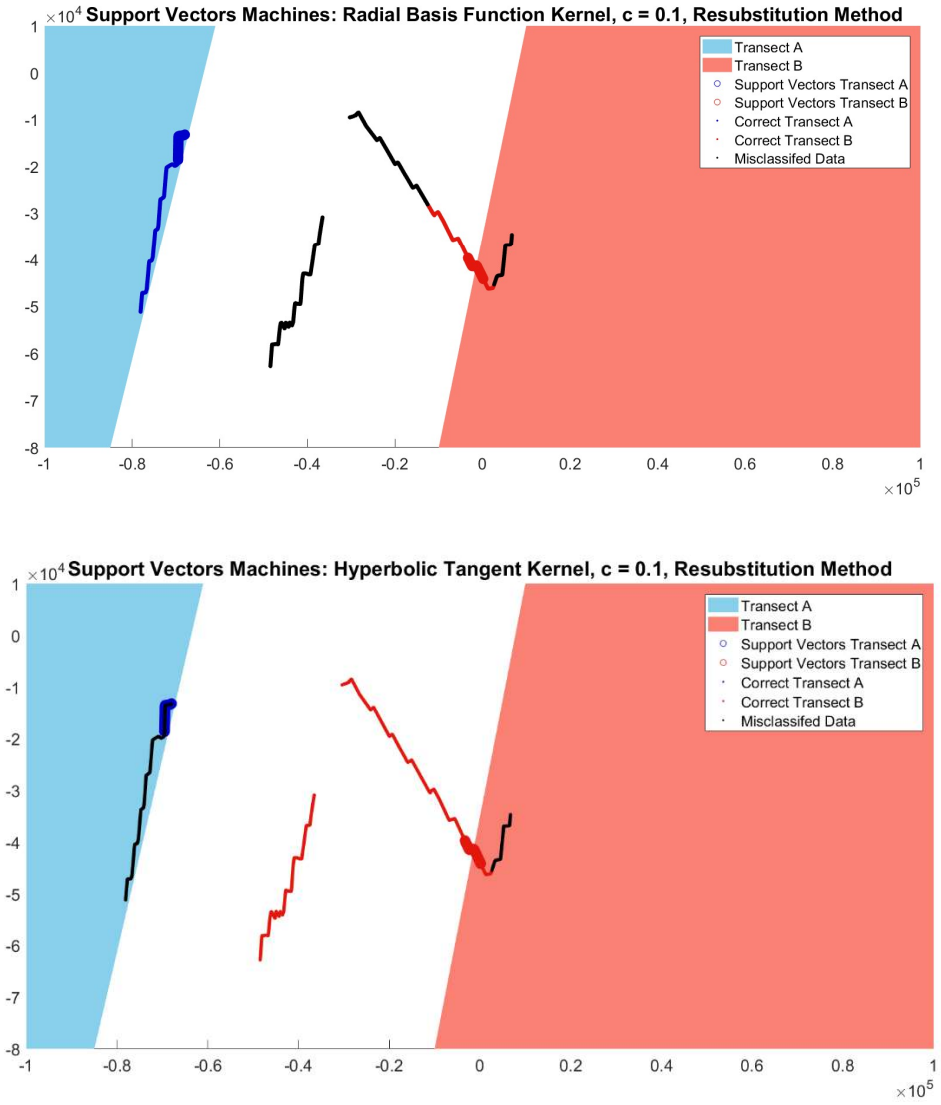
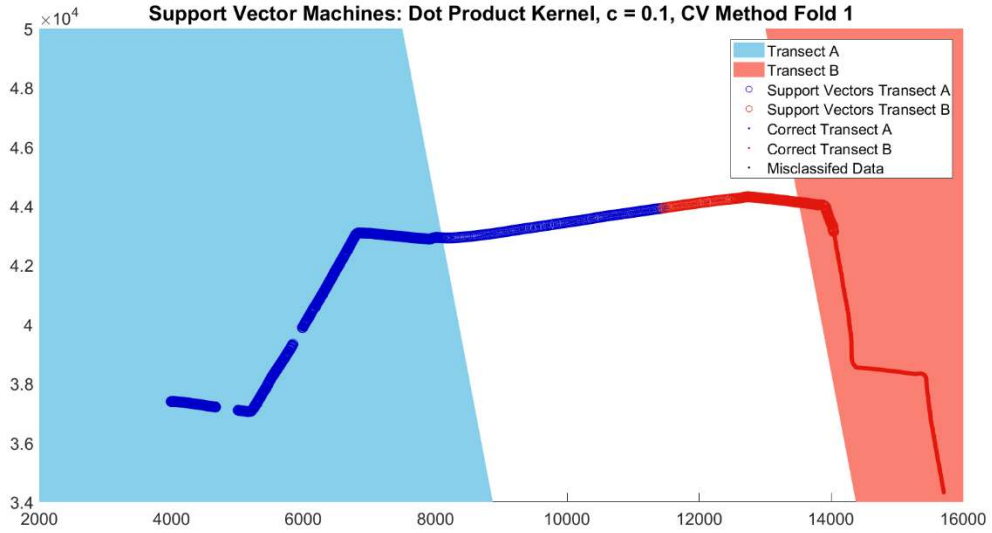
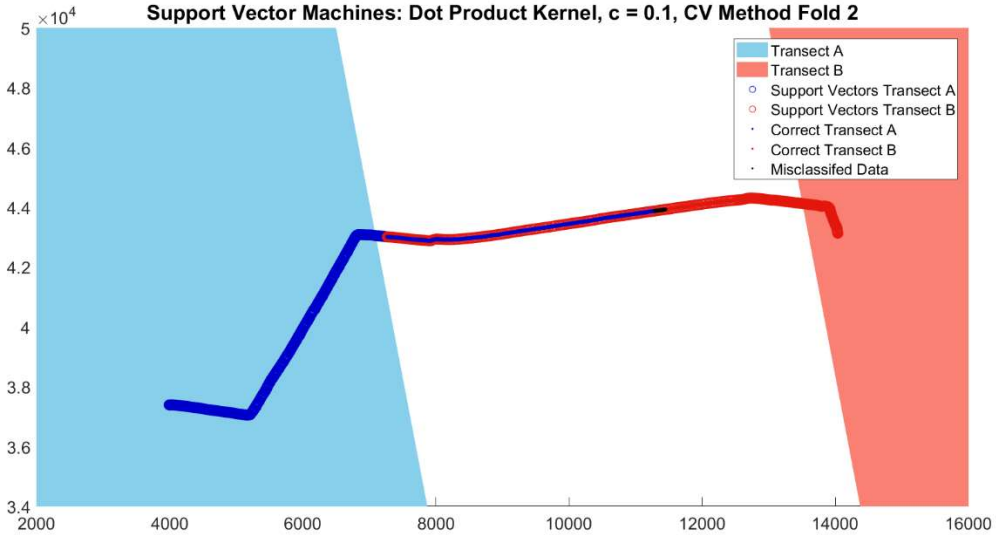


Figure 5. Support vectors for the re-substitution method, $c = 0.1$.


Support Vector Machines: Dot Product Kernel, c = 0.1, CV Method Fold 1
×10^4
5
4.8
4.6
4.4
4.2
4
3.8
3.6
3.4
2000
4000
6000
8000
10000
12000
14000
16000
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data


Support Vector Machines: Dot Product Kernel, c = 0.1, CV Method Fold 2
×10^4
5
4.8
4.6
4.4
4.2
4
3.8
3.6
3.4
2000
4000
6000
8000
10000
12000
14000
16000
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

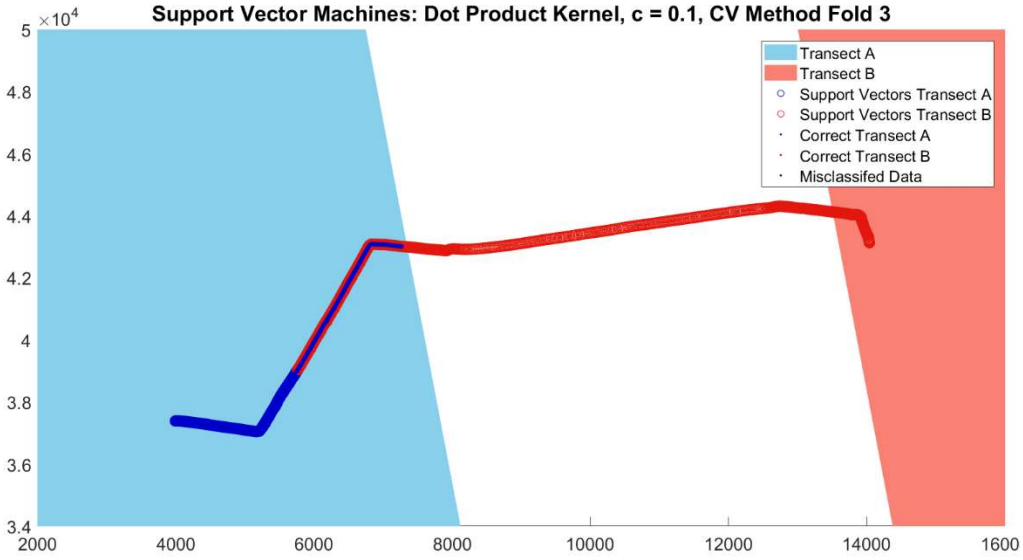
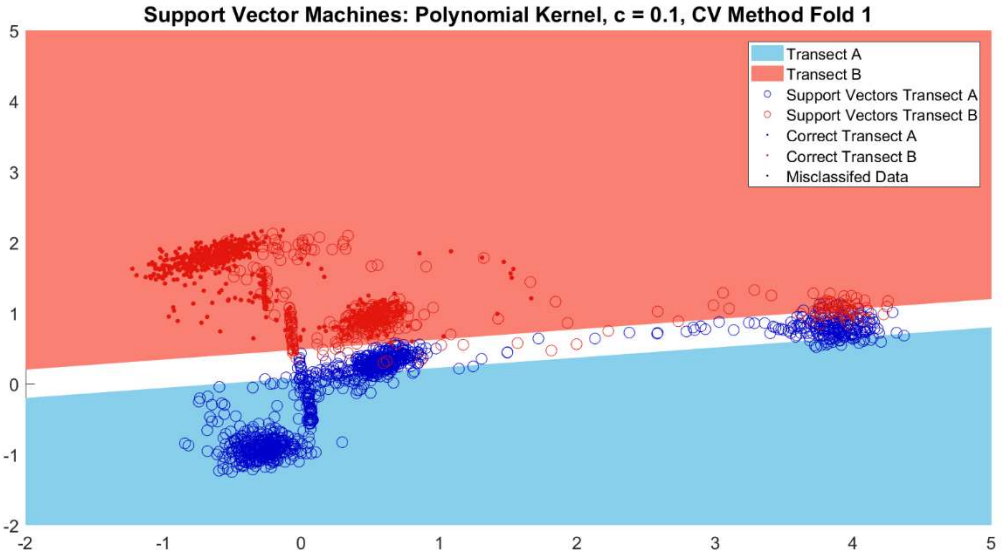
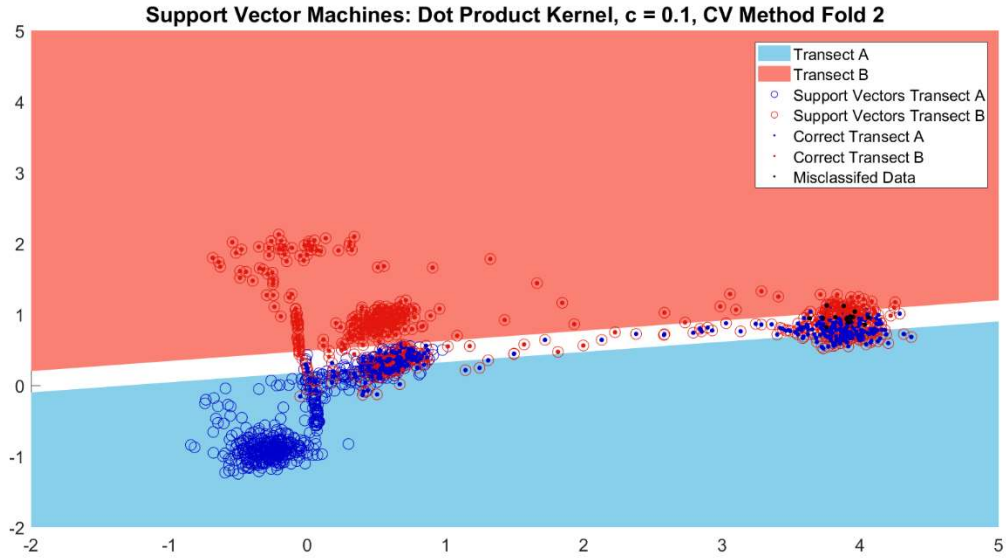

Support Vector Machines: Dot Product Kernel, c = 0.1, CV Method Fold 3
×10^4
5
4.8
4.6
4.4
4.2
4
3.8
3.6
3.4
2000
4000
6000
8000
10000
12000
14000
1600
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

Figure 6. Support Vectors for three-fold cross-validation method on the full dataset using the dot product kernel and c = 0.1, which had the best results. The remaining graphs for the polynomial and RBF kernel may be found in the appendix.


Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 1
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data


Support Vector Machines: Dot Product Kernel, c = 0.1, CV Method Fold 2
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

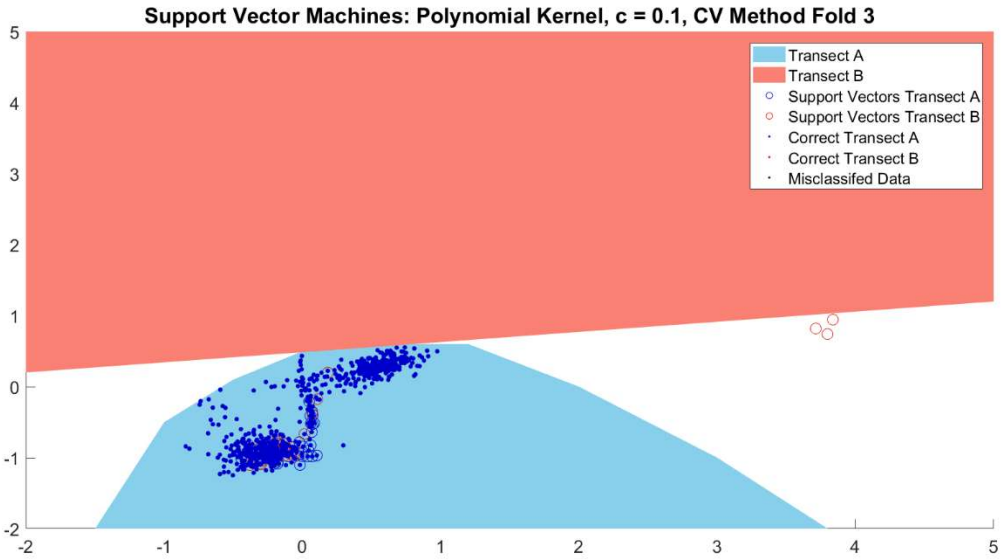
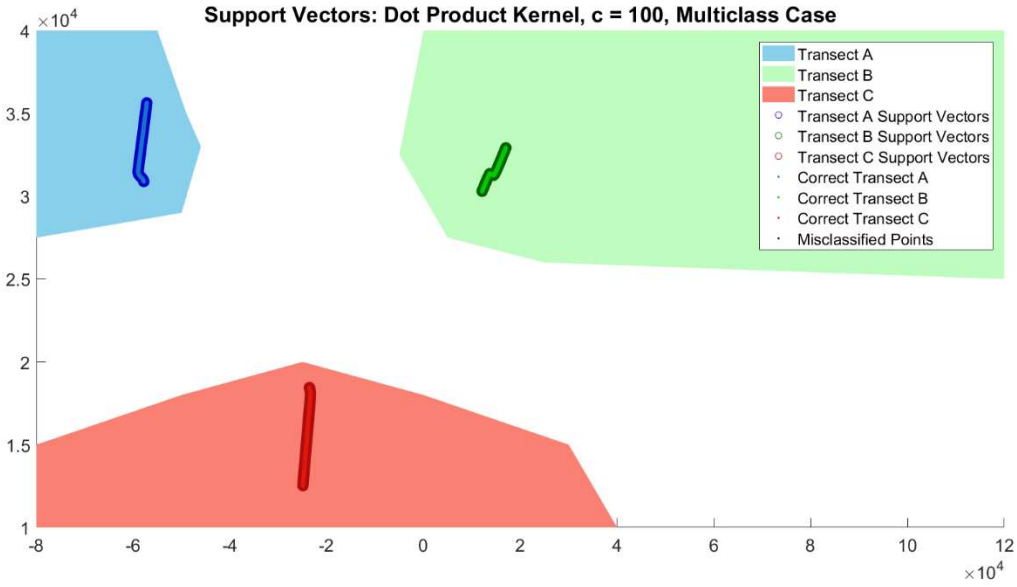
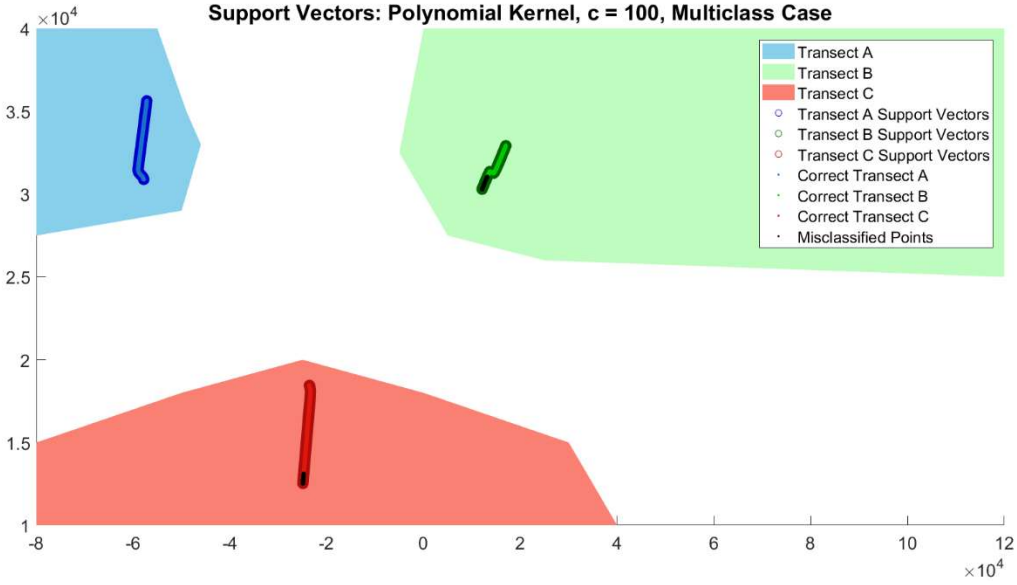

Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 3
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

Figure 7. Support Vectors for three-fold CV using the normalized dataset. The best results for each fold are presented here, and the remaining results are presented in the appendix.


Support Vectors: Dot Product Kernel, c = 100, Multiclass Case
×10^4
4
3.5
3
2.5
2
1.5
1
-8
-6
-4
-2
0
2
4
6
8
10
12
×10^4
Transect A
Transect B
Transect C
Transect A Support Vectors
Transect B Support Vectors
Transect C Support Vectors
Correct Transect A
Correct Transect B
Correct Transect C
Misclassified Points


Support Vectors: Polynomial Kernel, c = 100, Multiclass Case
×10^4
4
3.5
3
2.5
2
1.5
1
-8
-6
-4
-2
0
2
4
6
8
10
12
×10^4
Transect A
Transect B
Transect C
Transect A Support Vectors
Transect B Support Vectors
Transect C Support Vectors
Correct Transect A
Correct Transect B
Correct Transect C
Misclassified Points

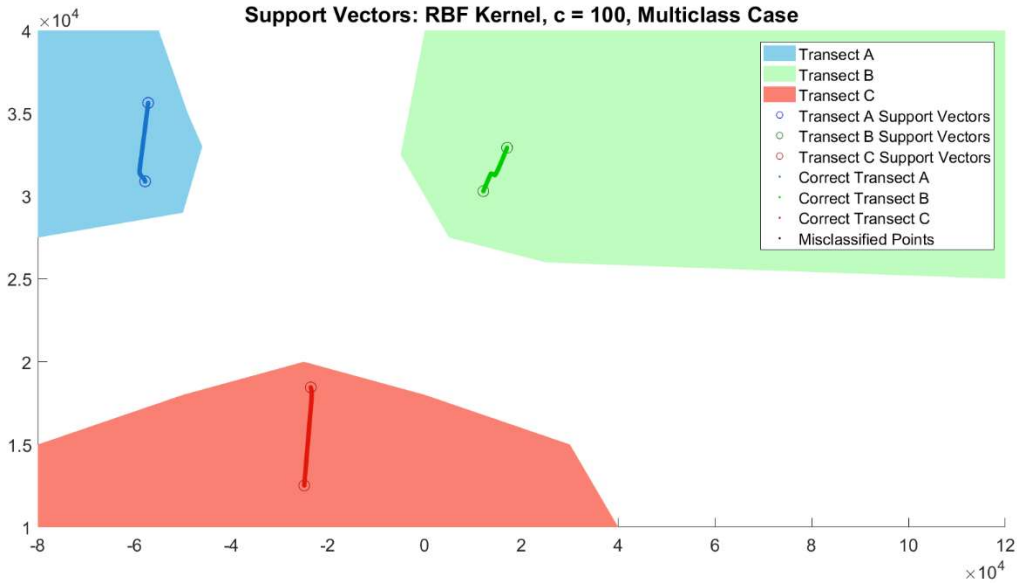


Figure 8. Support Vectors for multiclass re-substitution method. The overall graphs for the dot product, polynomial, and RBF kernel are presented here. The one-against-all method graphs for all classes and kernel functions are presented in the appendix.

The constant value (c value) determines how much the hyperplane must move to best separate non-linearly separable data points with the maximum margin. In the re-substitution and three-fold cross validation method, the constant values that tended to have the highest accuracy were c = 0.1 and c = 100 (for example, Figure 5). The classification rates generally did not tend to change drastically with different constant values, suggesting that the data was very closely packed together and that the hyperplane was not able to move to better classify the data. The changes tended to stay within 10%. However, there were some cases where changing the value of the constant changed the results of SVM classification accuracy. The RBF kernel had relatively high variation. In the three-folds CV with the normalized dataset, the classification rate jumped from 43.91% at c = 0.1 to 89.24% at c = 10 and c = 100, approximately doubling in accuracy. In the three-folds CV using the entire dataset, the classification rate dropped from 94.38% at c = 0.1 to 66.76% at c = 10 and 65.43% at c = 100. The type of data used may affected the performance of each constant value. The polynomial kernel function also tended to have variation across the different constant values, but tended to remain closer together. The largest difference came from the multiclass case, where the classifier ranged from 75.95% accuracy with c = 10 to 96.05% with c = 100 when separating transect C samples from the remaining data. However, the classifier experienced more difficulty in separating transect B from the data than it did with transects A and C, which suggests that data samples from transect B and transect C have more similarities among them. The use of c = 100 in the entire multiclass polynomial case would yield better accuracy than the use of the other constant values.

5. Conclusion

The application of PCA onto the ADCP datasets for dimensionality reduction revealed clustering trends within the data. The most important features in the data tended to be the temperature, heading, location of the data, and average depth at which each measurement was taken. The consistency of the various applications of PCA in determining common features suggests that the analysis is robust. The driving features were similar for one-dimensional, two-dimensional, and three-dimensional PCA, as well as for the applications onto individual transects. PCA allowed for the visualization of the data, which was extremely difficult to do with such high dimensionality, while preserving the qualities of all original dimensions. This reduced dataset can be used in graphing and other data analytics techniques, such as SVM.

In the various applications of SVM on *reduced*, *normalized*, and *full* datasets, the dot product and the polynomial (degree 2; Q=2) kernel functions were found to be the most effective for this data regardless of the constant (c) values. The data was closely clustered together which made it difficult for certain kernel functions, such as the hyperbolic tangent kernel, to separate the classes. The SVM algorithms were ultimately able to accurately categorize data by their location or transect collected using physical, mechanical, and chemical properties such as velocity and temperature.

Both PCA and SVM were found useful in this oceanographic data application. PCA techniques may be applied to other large data sets, such as those from remote sensing devices, as a method of data mining. SVM can assist with classifying this data into classes or categories based on their various properties. The application in this study resulted in high accuracy in classifying data points based on their location and properties, and thus would be useful in finding common characteristics between other data points in different locations. Future directions include using other machine learning techniques, such as semi-supervised learning, to discover more about the data collected. These machine learning techniques may help oceanographers and other marine scientists to interpret and organize their data analysis, especially with large datasets when the data is difficult to visualize and when the traditional experimental procedures required to analyze the data may be tedious and cumbersome.

Appendix A: Confusion matrices with the correctly classified and misclassified data. The kernel function and the c value used in each run or fold is listed in the figure title.

A1: Confusion matrices for the re-substitution method.

Radial Basis Function: c = 0.1

	True Class 1	True Class 2
Classifier 1	3375	6247
Classifier 2	1192	2276
b = 0.0188		

Radial Basis Function: c = 10

	True Class 1	True Class 2
Classifier 1	3375	6256
Classifier 2	1192	2267
b = 0.0207		

Radial Basis Function: c = 100

	True Class 1	True Class 2
Classifier 1	3375	6256
Classifier 2	1192	2267
b = 0.0207		

Dot Product: c = 0.1

	True Class 1	True Class 2
Classifier 1	3375	600
Classifier 2	1192	7923
b = -12.466		

Dot Product: c = 10

	True Class 1	True Class 2
Classifier 1	3375	577
Classifier 2	1192	7946
b = -18.2212		

Dot Product: c = 100

	True Class 1	True Class 2
Classifier 1	3375	987
Classifier 2	1192	7536
b = -13.6747		

Hyperbolic Tangent: c = 0.1

	True Class 1	True Class 2
Classifier 1	0	0
Classifier 2	4567	8523
$b = -8.08 * 10^{10}$		

Hyperbolic Tangent: c = 10

	True Class 1	True Class 2
Classifier 1	0	0
Classifier 2	4567	8523
$b = -8.08 * 10^{12}$		

Hyperbolic Tangent: c = 100

	True Class 1	True Class 2
Classifier 1	0	0
Classifier 2	4567	8523
$b = -8.08 * 10^{13}$		

Polynomial: c = 0.1

	True Class 1	True Class 2
Classifier 1	3375	0
Classifier 2	1992	8523
b = $-9.4177 * 10^{22}$		

Polynomial: c = 10

	True Class 1	True Class 2
Classifier 1	3375	446
Classifier 2	1992	8077
b = $-6.7621 * 10^{24}$		

Polynomial: c = 100

	True Class 1	True Class 2
Classifier 1	3375	0
Classifier 2	1992	8523
b = $-5.3406 * 10^{25}$		

A2: Average confusion matrices for three-fold CV using the normalized dataset

Radial Basis Function: c = 0.1

	True Class 1	True Class 2
Classifier 1	174.67	217.33
Classifier 2	175.33	132.67

Radial Basis Function: c = 10

	True Class 1	True Class 2
Classifier 1	274.67	0
Classifier 2	75.33	350

Radial Basis Function: c = 100

	True Class 1	True Class 2
Classifier 1	274.67	0
Classifier 2	75.33	350

Dot Product: c = 0.1

	True Class 1	True Class 2
Classifier 1	345	0
Classifier 2	5	350

Dot Product: c = 10

	True Class 1	True Class 2
Classifier 1	345	0
Classifier 2	5	350

Dot Product: c = 100

	True Class 1	True Class 2
Classifier 1	345	0
Classifier 2	5	350

Hyperbolic Tangent: c = 0.1

	True Class 1	True Class 2
Classifier 1	73	281
Classifier 2	277	69

Hyperbolic Tangent: c = 10

	True Class 1	True Class 2
Classifier 1	73	281
Classifier 2	277	69

Hyperbolic Tangent: c = 100

	True Class 1	True Class 2
Classifier 1	73	281
Classifier 2	277	69

Polynomial: c = 0.1

	True Class 1	True Class 2
Classifier 1	349.33	0
Classifier 2	0.67	350

Polynomial: c = 10

	True Class 1	True Class 2
Classifier 1	331.33	0.33
Classifier 2	18.67	349.67

Polynomial: c = 100

	True Class 1	True Class 2
Classifier 1	323.33	0.66
Classifier 2	26.67	349.33

A3: Average confusion matrices for three-fold CV using the reduced dataset.

Radial Basis Function: c = 0.1

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Radial Basis Function: c = 10

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Radial Basis Function: c = 100

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Dot Product: c = 0.1

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Dot Product: c = 10

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Dot Product: c = 100

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Hyperbolic Tangent: c = 0.1

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Hyperbolic Tangent: c = 10

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Hyperbolic Tangent: c = 100

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Polynomial: c = 0.1

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Polynomial: c = 10

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

Polynomial: c = 100

	True Class 1	True Class 2
Classifier 1	166.67	0
Classifier 2	0	166.33

A4: Confusion matrices for multiclass cases

Radial Basis Function: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	0
Classifier 2&3	0	1000

Radial Basis Function: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	0
Classifier 1&3	0	1000

Radial Basis Function: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	0
Classifier 1&2	0	1000

Dot Product: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	0
Classifier 2&3	0	1000

Dot Product: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	0
Classifier 1&3	0	1000

Dot Product: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	0
Classifier 1&2	0	1000

Hyperbolic Tangent: Transect A

	True Class 1	True Class 2&3
Classifier 1	0	0
Classifier 2&3	500	1000

Hyperbolic Tangent: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	1000
Classifier 1&3	0	0

Hyperbolic Tangent: Transect C

	True Class 3	True Class 1&2
Classifier 3	500	1000
Classifier 1&2	0	0

Polynomial: Transect A

	True Class 1	True Class 2&3
Classifier 1	500	91
Classifier 2&3	0	909

Polynomial: Transect B

	True Class 2	True Class 1&3
Classifier 2	500	500
Classifier 1&3	0	500

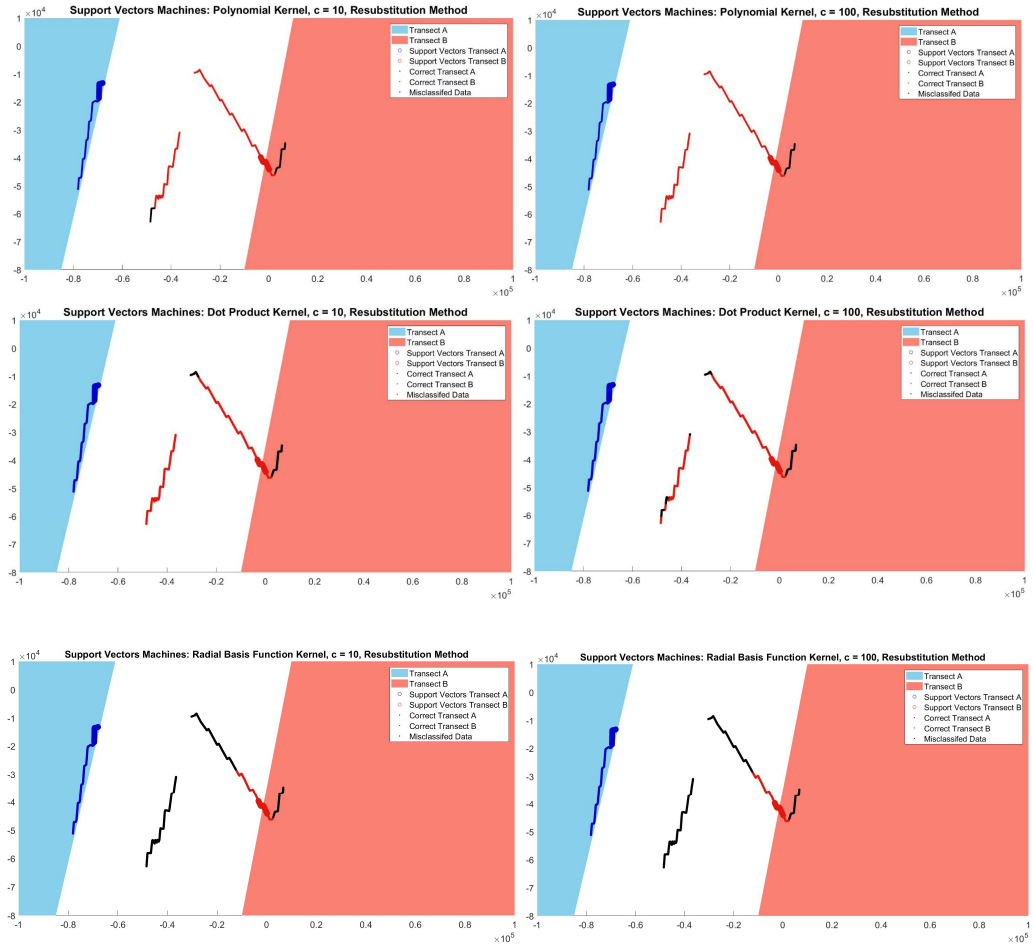
Polynomial: Transect C

	True Class 3	True Class 2&3
Classifier 3	500	310.33
Classifier 2&3	0	689.67

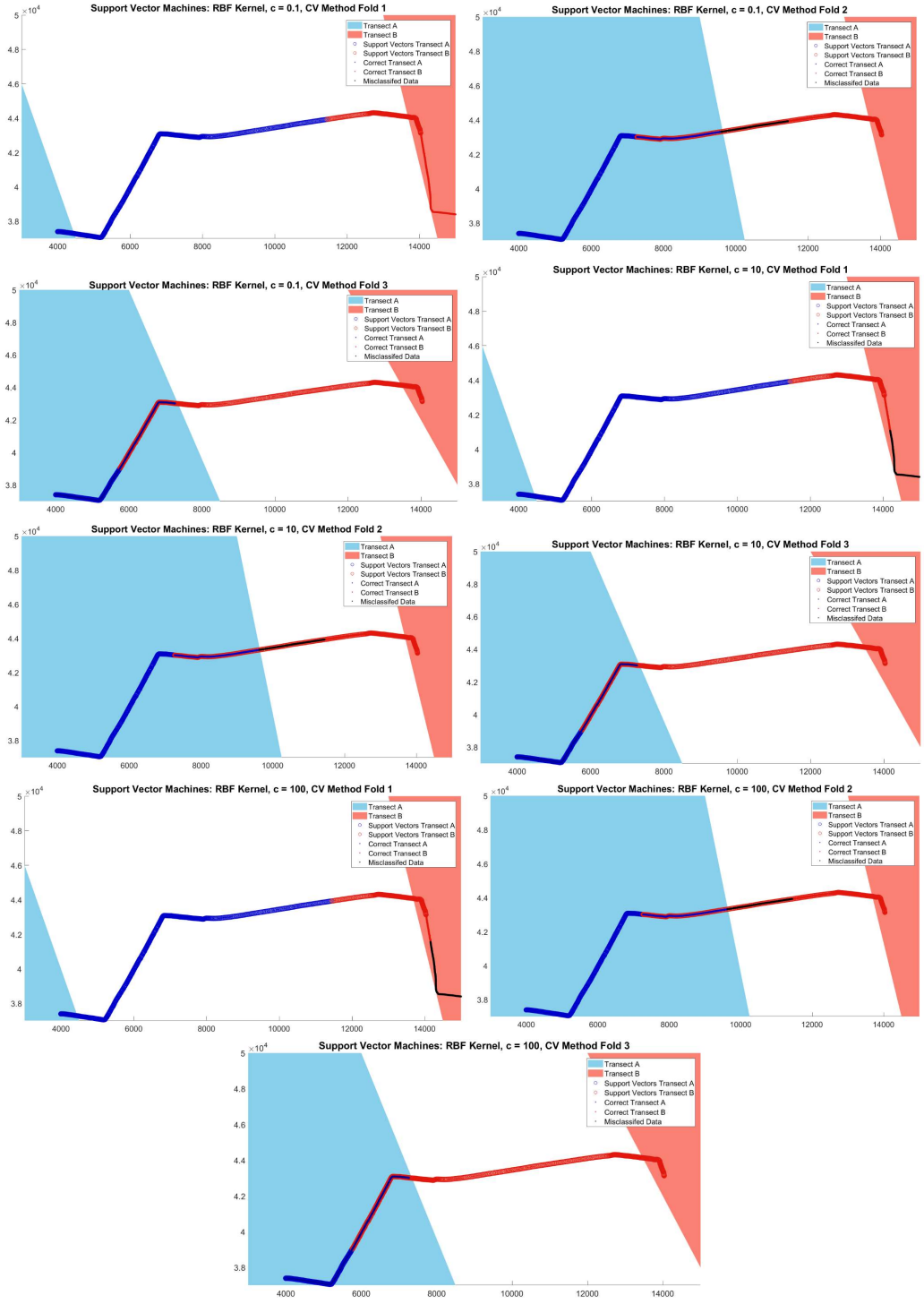
Appendix B: Plots of the SVM results with the correctly classified and misclassified data. The kernel function and the c value used in each run or fold is listed in the figure title.

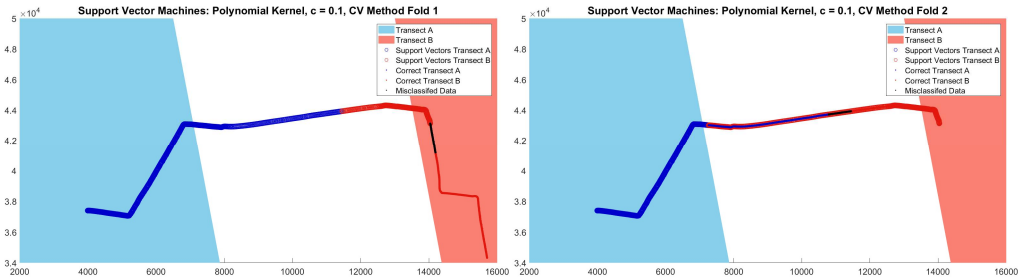
B1: SVM Re-substitution method, c = 10 and c = 100

a.

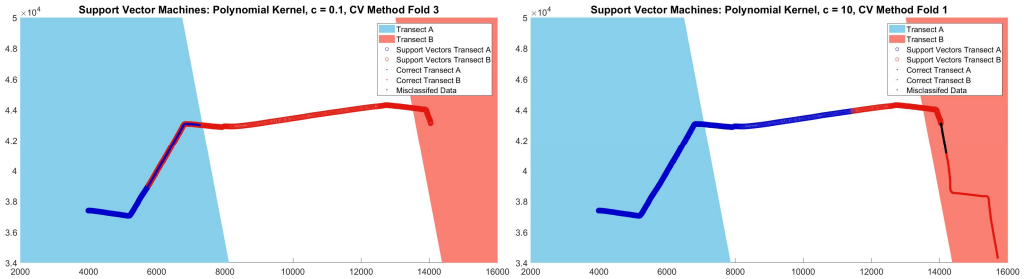


B2: SVM results using the full dataset

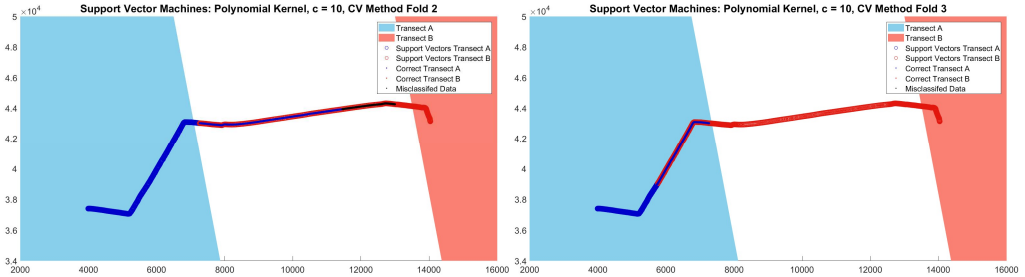



Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 1
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

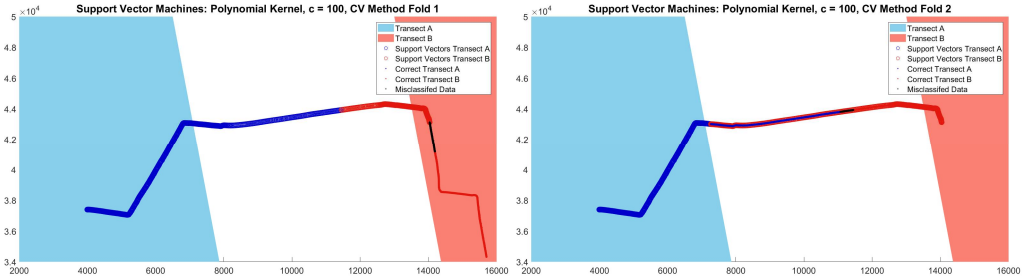
Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 2
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data


Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 3
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

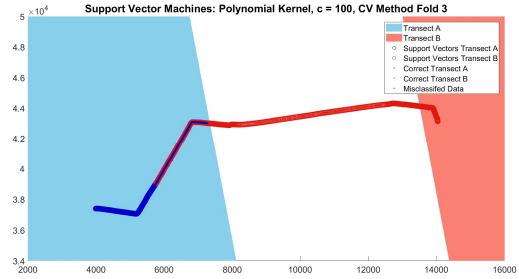
Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 1
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

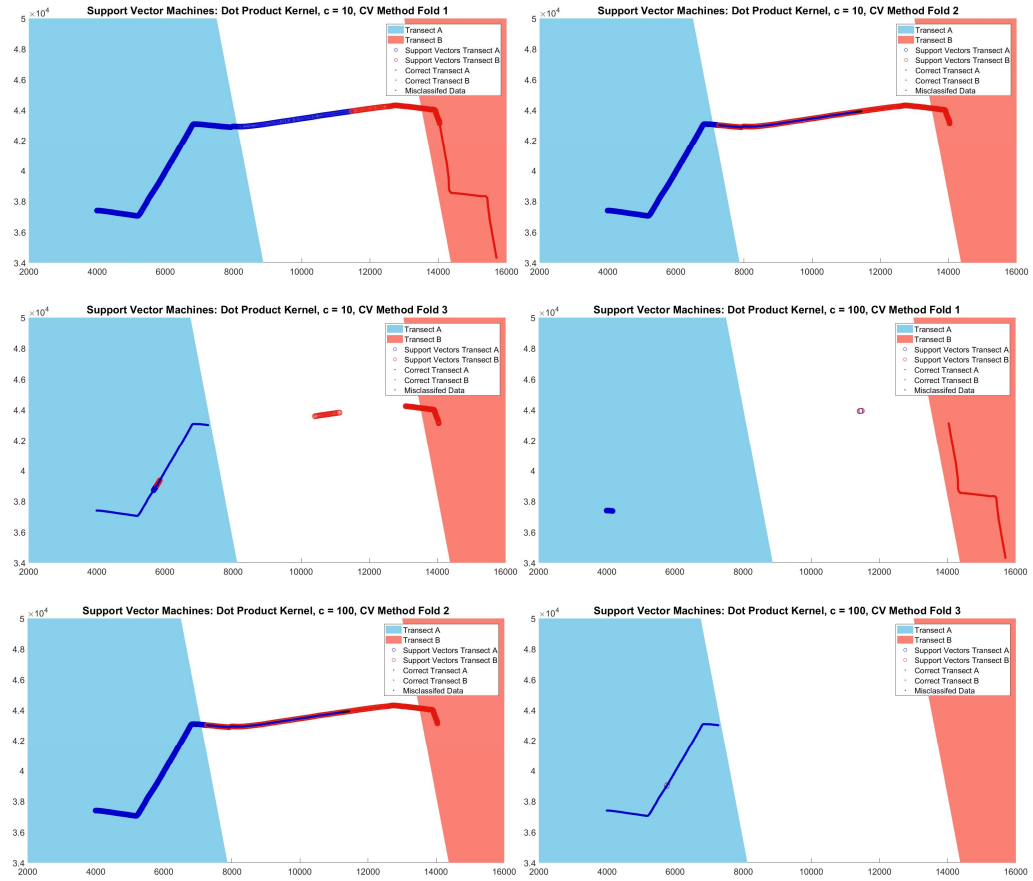

Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 2
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 3
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

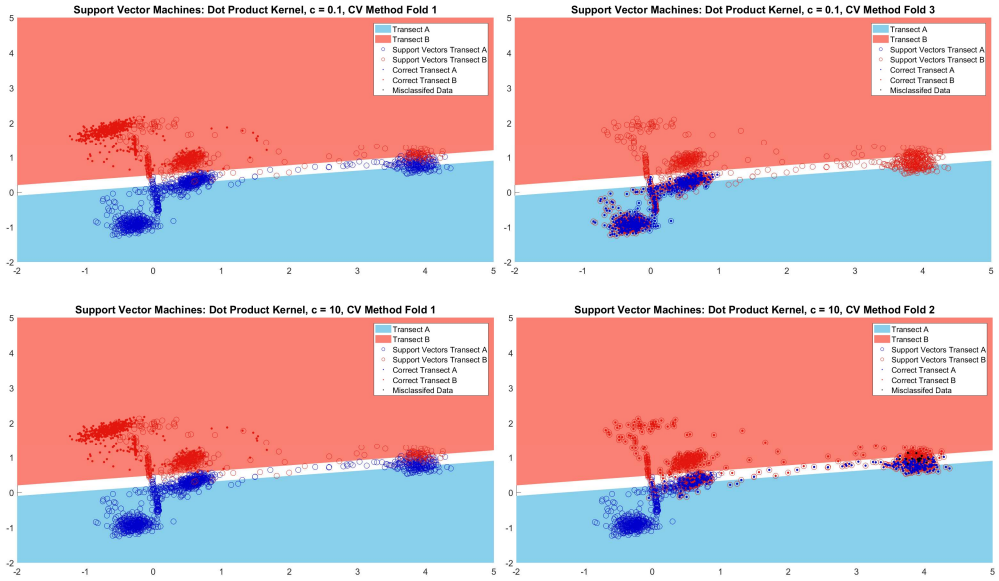

Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 1
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

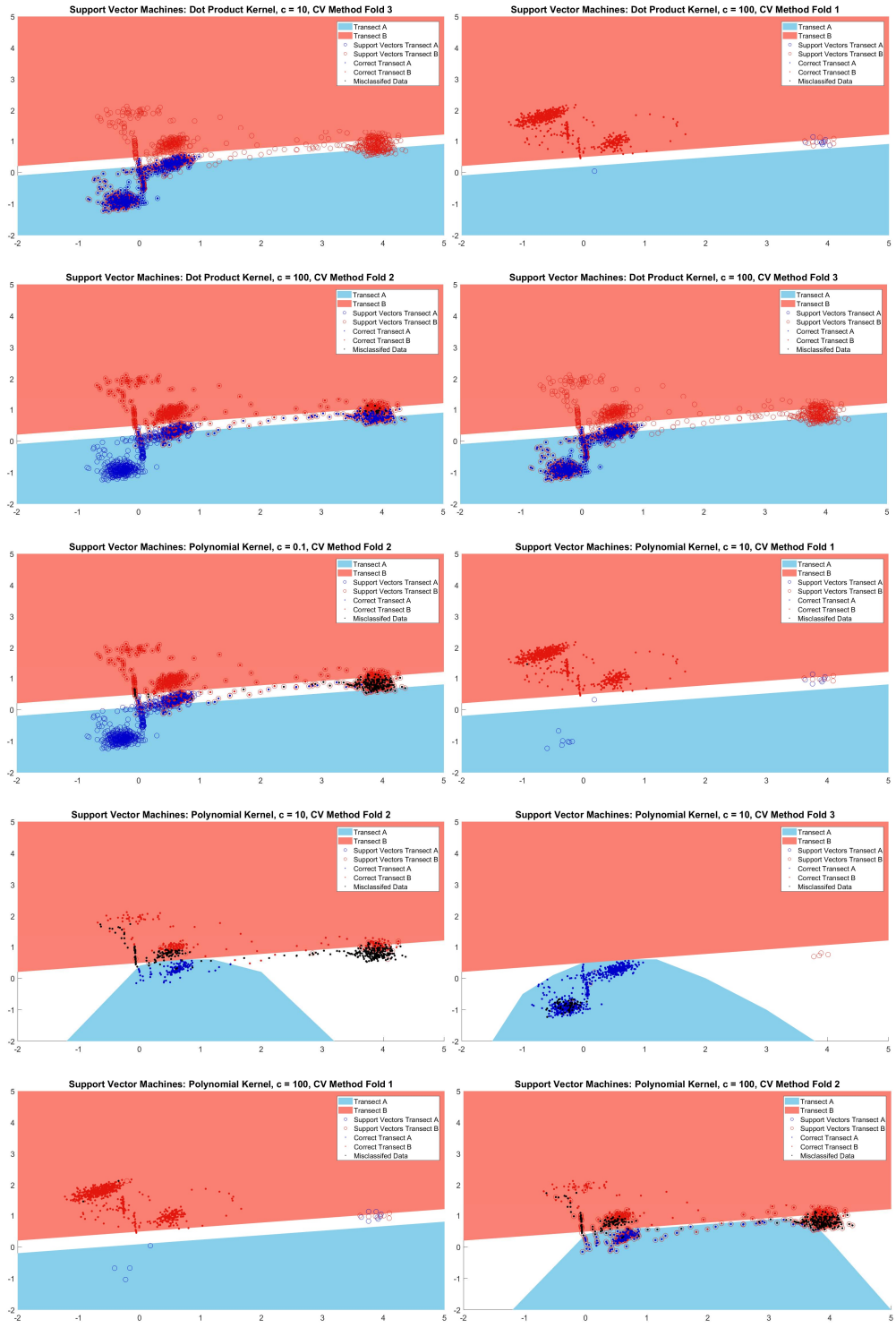
Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 2
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

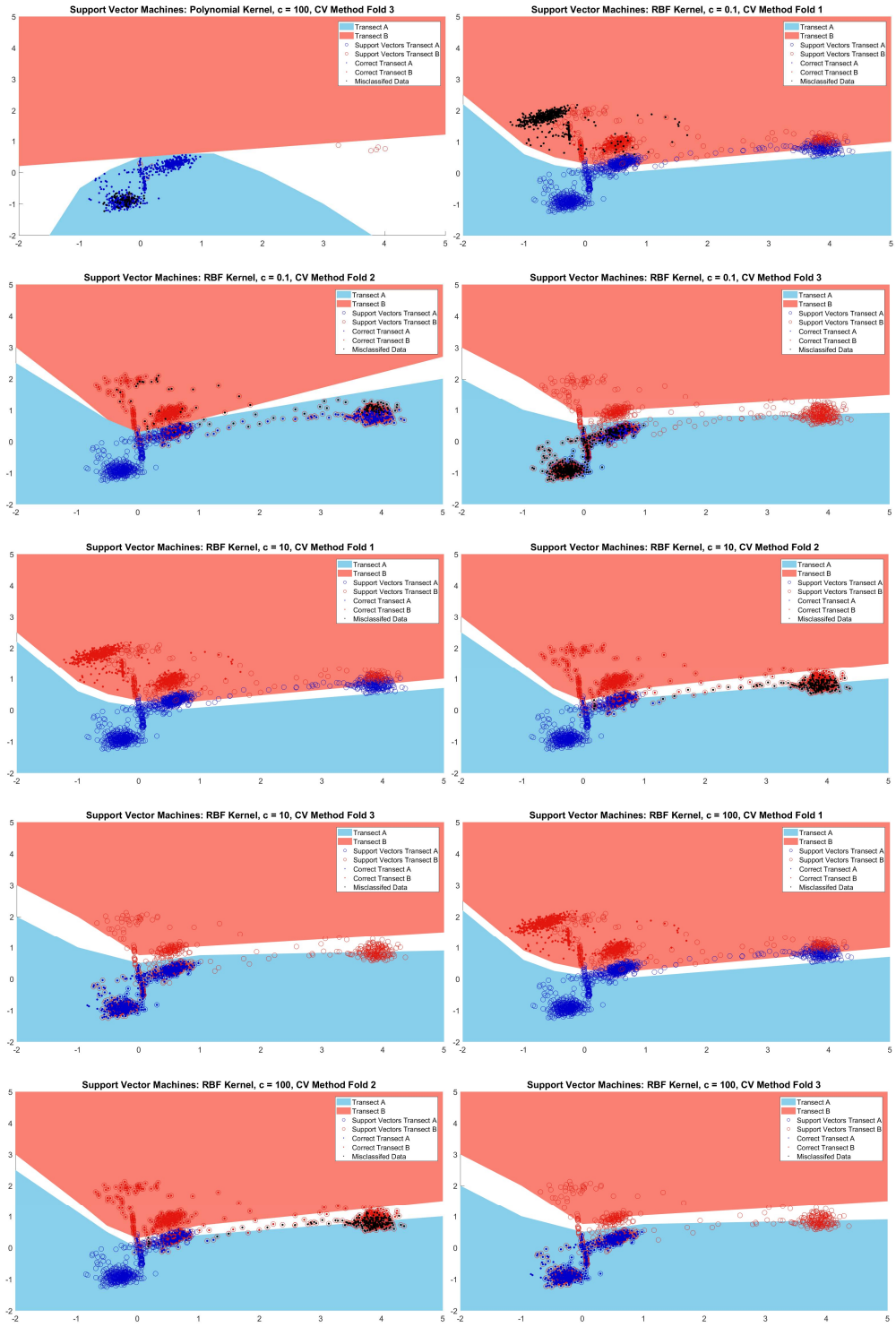

Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 3
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data



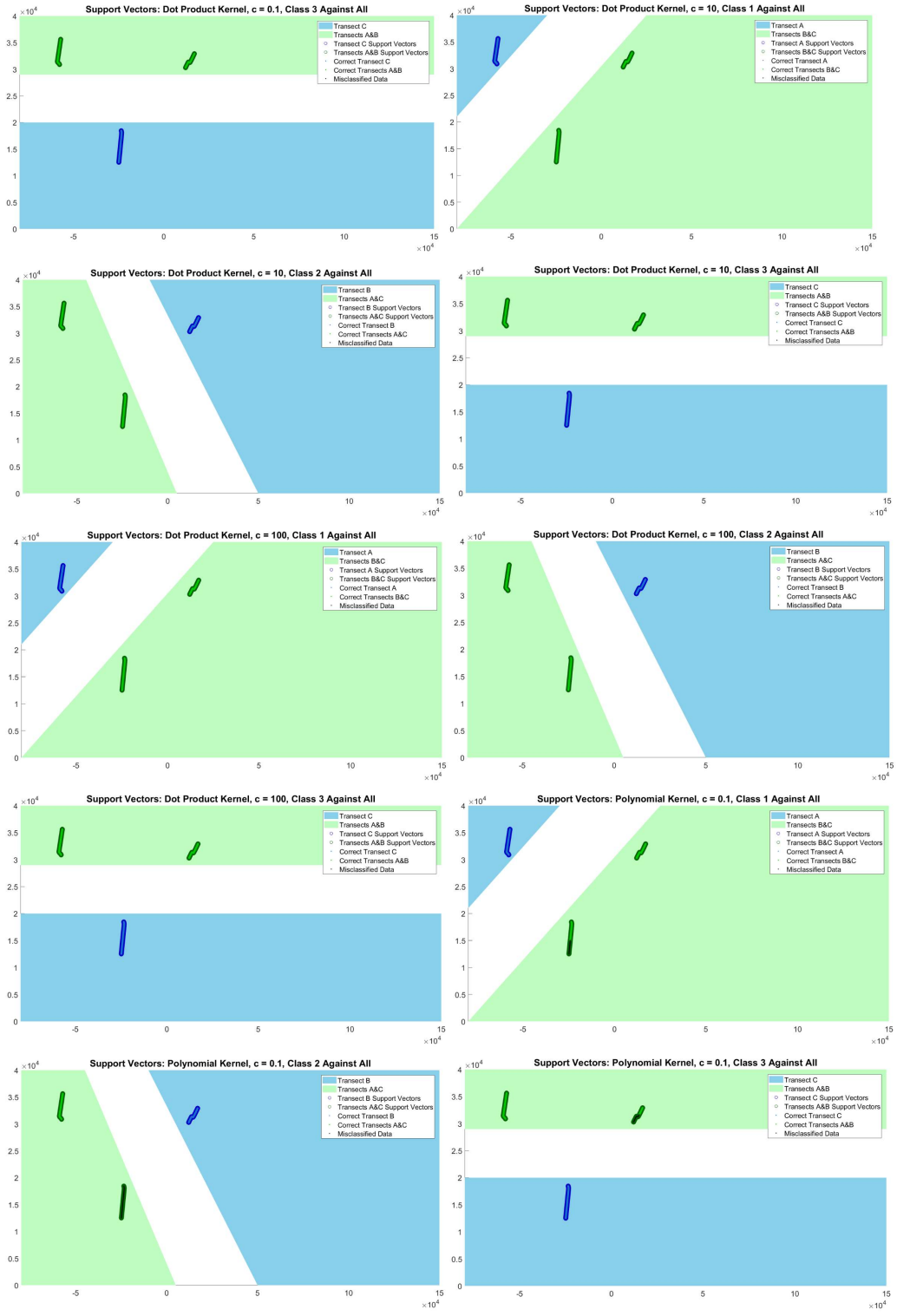
B3: SVM results using the normalized dataset

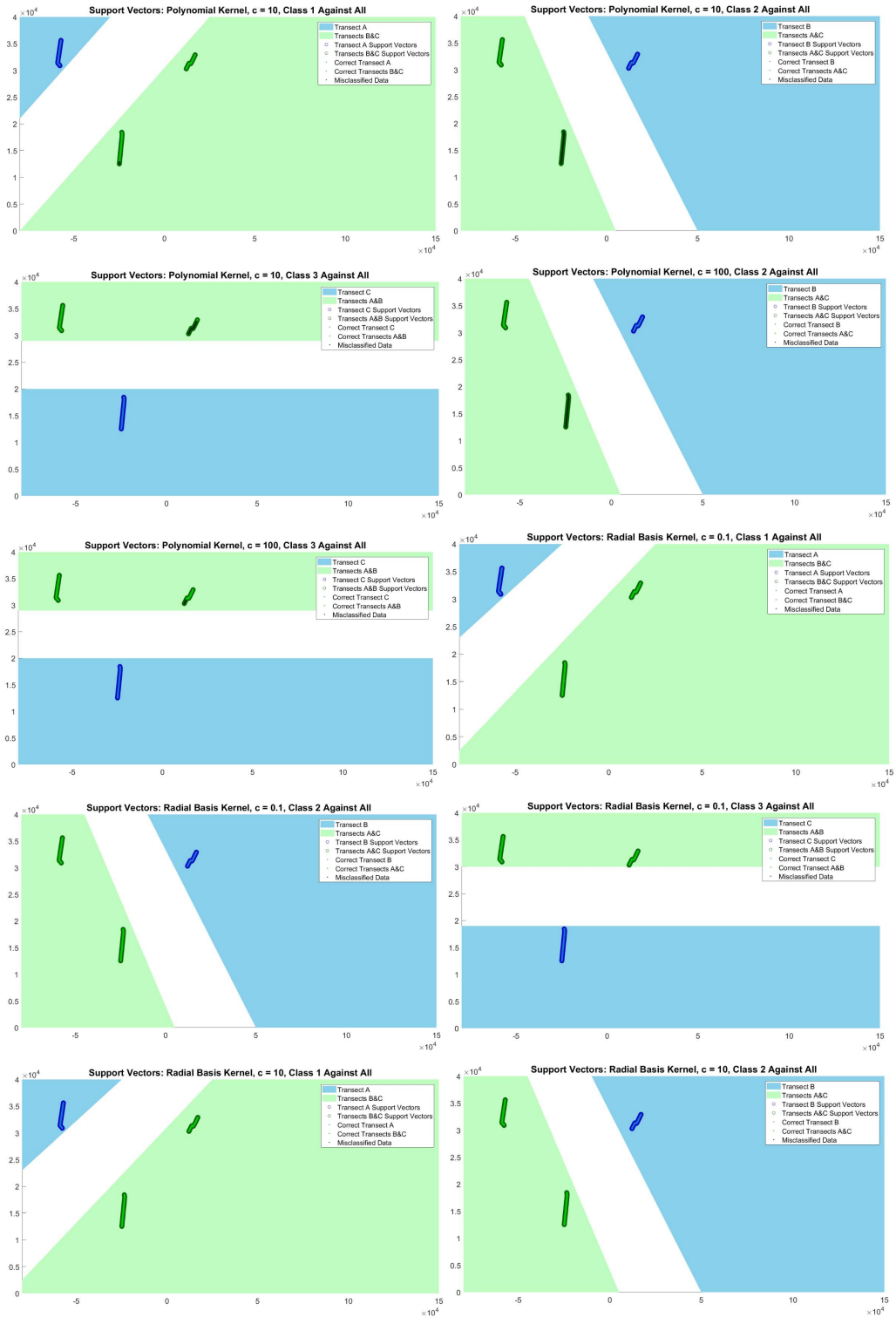


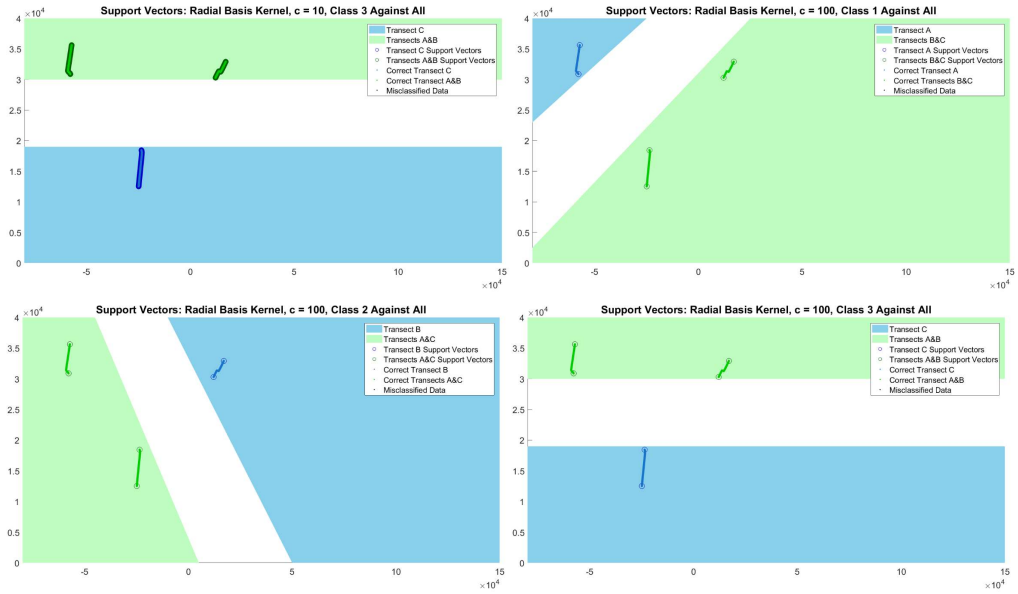

Support Vector Machines: Dot Product Kernel, c = 10, CV Method Fold 3
Support Vector Machines: Dot Product Kernel, c = 100, CV Method Fold 1
Support Vector Machines: Dot Product Kernel, c = 100, CV Method Fold 2
Support Vector Machines: Dot Product Kernel, c = 100, CV Method Fold 3
Support Vector Machines: Polynomial Kernel, c = 0.1, CV Method Fold 2
Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 1
Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 2
Support Vector Machines: Polynomial Kernel, c = 10, CV Method Fold 3
Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 1
Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 2
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data


Support Vector Machines: Polynomial Kernel, c = 100, CV Method Fold 3
Support Vector Machines: RBF Kernel, c = 0.1, CV Method Fold 1
Support Vector Machines: RBF Kernel, c = 0.1, CV Method Fold 2
Support Vector Machines: RBF Kernel, c = 0.1, CV Method Fold 3
Support Vector Machines: RBF Kernel, c = 10, CV Method Fold 1
Support Vector Machines: RBF Kernel, c = 10, CV Method Fold 2
Support Vector Machines: RBF Kernel, c = 10, CV Method Fold 3
Support Vector Machines: RBF Kernel, c = 100, CV Method Fold 1
Support Vector Machines: RBF Kernel, c = 100, CV Method Fold 2
Support Vector Machines: RBF Kernel, c = 100, CV Method Fold 3
Transect A
Transect B
Support Vectors Transect A
Support Vectors Transect B
Correct Transect A
Correct Transect B
Misclassifed Data

Appendix C: Support Vectors for multiclass re-substitution method.




Support Vectors: Polynomial Kernel, c = 10, Class 1 Against All
Transect A
Transects B&C
Transect A Support Vectors
Transects B&C Support Vectors
Correct Transect A
Correct Transects B&C
Misclassified Data
Support Vectors: Polynomial Kernel, c = 10, Class 2 Against All
Transect B
Transects A&C
Transect B Support Vectors
Transects A&C Support Vectors
Correct Transect B
Correct Transects A&C
Misclassified Data
Support Vectors: Polynomial Kernel, c = 10, Class 3 Against All
Transect C
Transects A&B
Transect C Support Vectors
Transects A&B Support Vectors
Correct Transect C
Correct Transects A&B
Misclassified Data
Support Vectors: Polynomial Kernel, c = 100, Class 2 Against All
Transect B
Transects A&C
Transect B Support Vectors
Transects A&C Support Vectors
Correct Transect B
Correct Transects A&C
Misclassified Data
Support Vectors: Polynomial Kernel, c = 100, Class 3 Against All
Transect C
Transects A&B
Transect C Support Vectors
Transects A&B Support Vectors
Correct Transect C
Correct Transects A&B
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 0.1, Class 1 Against All
Transect A
Transects B&C
Transect A Support Vectors
Transects B&C Support Vectors
Correct Transect A
Correct Transect B&C
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 0.1, Class 2 Against All
Transect B
Transects A&C
Transect B Support Vectors
Transects A&C Support Vectors
Correct Transect B
Correct Transects A&C
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 0.1, Class 3 Against All
Transect C
Transects A&B
Transect C Support Vectors
Transects A&B Support Vectors
Correct Transect C
Correct Transect A&B
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 10, Class 1 Against All
Transect A
Transects B&C
Transect A Support Vectors
Transects B&C Support Vectors
Correct Transect A
Correct Transects B&C
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 10, Class 2 Against All
Transect B
Transects A&C
Transect B Support Vectors
Transects A&C Support Vectors
Correct Transect B
Correct Transects A&C
Misclassified Data


Support Vectors: Radial Basis Kernel, c = 10, Class 3 Against All
Transect C
Transects A&B
Transect C Support Vectors
Transects A&B Support Vectors
Correct Transect C
Correct Transect A&B
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 100, Class 1 Against All
Transect A
Transects B&C
Transect A Support Vectors
Transects B&C Support Vectors
Correct Transect A
Correct Transects B&C
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 100, Class 2 Against All
Transect B
Transects A&C
Transect B Support Vectors
Transects A&C Support Vectors
Correct Transect B
Correct Transects A&C
Misclassified Data
Support Vectors: Radial Basis Kernel, c = 100, Class 3 Against All
Transect C
Transects A&B
Transect C Support Vectors
Transects A&B Support Vectors
Correct Transect C
Correct Transect A&B
Misclassified Data

Acknowledgements

The authors would like to thank Diane Fribance for the ADCP data collection. This study was funded by the National Science Foundation (NSF) Research Experiences for Undergraduates program under award AGS 1560210. Computing is done on the CCU Cyberinfrastructure supported in part by the NSF Major Research Instrument program under award AGS 1624068.

References

Bro, R. and A. K. Smilde (2014), Principal component analysis. Analytical Methods, 6, 2812-2831, http://pubs.rsc.org/en/content/articlepdf/2014/AY/C3AY41907J

Flagg, C.N., G. Schwartze, E. Gottlieb, and T. Rossby (1998), Operating an Acoustic Doppler Current Profiler aboard a Container Vessel. J. Atmos. Oceanic Technol., 15, 257–271, https://doi.org/10.1175/1520-0426(1998)015<0257:OAADCP>2.0.CO;2

Goebel, M. and L. Gruenwald (1999), A survey of data mining and knowledge discovery software tools, ACM SIGKDD Explorations Newsletter 1(1), 20-33.

Hand, D. (2007), Principles of Data Mining, Drug Safety 30(7), 621-622.

Healey, C.G. (1998), On the Use of Perceptual Cues and Data Mining for Effective Visualization of Scientific Datasets.

Huang, Y., L. Kao, and F. E. Sandes (2007), Predicting ocean salinity and temperature variation using data mining and fuzzy inference. International Journal of Fuzzy Systems, 9(3), 143-151.

Kostaschuk, R., J. Best, P. Villard, J. Peakall, and M. Franklin (2005), Measuring flow velocity and sediment transport with an acoustic Doppler current profiler. Geomorphology 68(1-2), 25-37.

Lorke, A., D. F. McGinnis, P. Spaak, and A. Wuest (2004), Acoustic observations of zooplankton in lakes using a Doppler current profiler. Freshwater Biology 49(10), 1280-1292.

Mountrakis, G., J. Im, and C. Ogole (2011), Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 66(3), 247-259.

New, A. L. (1992) Factors affecting the quality of shipboard acoustic Doppler current profiler data. Deep Sea Research Part A. Oceanographic Research Papers 39(11-12), 1985-1996.

Steinbach, M., P. Tan, V. Kumar, S. Klooster, and C. Potter, (2002). Data mining for the discovery of ocean climate indices.

Teledyne RD Instruments (2011), Acoustic Doppler Current Profiler Principles of Operation: A Practical Primer.

Tong, S. and D. Koller (2001), Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 2, 45-66, http://www.jmlr.org/papers/volume2/tong01a/tong01a.pdf

Ursella, L. and M. Gacic (2001), Use of the Acoustic Doppler Current Profiler (ADCP) in the study of the circulation of the Adriatic Sea. Annales Geophysicae, European Geosciences Union, 19 (9), 1183-1193, https://hal.archives-ouvertes.fr/hal-00316908/

Vanhatalo, E., M. Kulahci, and B. Bergquist (2017), On the structure of dynamic principal component analysis used in statistical process monitoring. Chemometrics and Intelligent Laboratory Systems 167(1), 1-11.