Autonomous Robots manuscript No.
(will be inserted by the editor)

Synthetic Data Generation for Classification via Uni-Modal Cluster Interpolation

Eric J. Coyle · Rodney Roberts · Emmanuel G. Collins, Jr. · Adrian Barbu

Received: date / Accepted: date

Abstract The observations used to classify data from real systems often vary as a result of changing operating conditions (e.g. velocity, load, temperature, etc.). Hence, to create accurate classification algorithms for these systems, observations from a large number of operating conditions must be used in algorithm training. This can be an arduous, expensive, and even dangerous task. Treating an operating condition as an inherently metric continuous variable (e.g. velocity, load or temperature) and recognizing that observations at a single operating condition can be viewed as a data cluster enables formulation of interpolation techniques. This paper presents a method that uses data clusters at operating conditions where data has been collected to estimate data clusters at other operating conditions, enabling classification. The mathematical tools that are key to the proposed data cluster interpolation method are Catmull-Rom splines, the Schur decomposition, singular value decomposition, and a special matrix interpolation function. The ability of this method to accurately estimate distribution, orientation and location in the feature space is then shown through three benchmark problems involving 2D feature vectors. The proposed method is applied to empirical data involving vibration-based terrain classification for an autonomous robot using a feature vector of dimension 300, to show that these estimated data clusters are more effective for classification purposes than known data clusters that correspond to different operating conditions. Ultimately, it is concluded that although collecting real data is ideal, these estimated data clusters can improve classification accuracy when it is inconvenient or difficult to collect additional data.

Keywords Interpolation · Singular Value Decomposition · Terrain Classification · Data Clusters · Pattern Classification

Eric J. Coyle
Department of Mechanical Engineering
Embry-Riddle Aeronautical University
Daytona Beach, Florida 32114, USA
Tel.: (386) 226-7441
E-mail: coylee1@erau.edu

Rodney G. Roberts
Department of Electrical Engineering
Center for Intelligent Systems, Control, and Robotics (CISCOR)
FAMU-FSU College of Engineering
Tallahassee, Florida 32310, USA
Tel.: (850) 410-6458
E-mail: rroberts@eng.fsu.edu

Emmanuel G. Collins, Jr.
Department of Mechanical Engineering
Center for Intelligent Systems, Control, and Robotics (CISCOR)
FAMU-FSU College of Engineering
Tallahassee, Florida 32310, USA
Tel.: (850) 410-6373
E-mail: ecollins@eng.fsu.edu

Adrian Barbu
Department of Statistics
Florida State University
Tallahassee, Florida 32310, USA
Tel.: (850) 290-5202
E-mail: abarbu@stat.fsu.edu

1 Introduction

Interpolation between data points in a n-dimensional space is a common mathematical problem for which several techniques have been developed. These techniques include linear interpolation, polynomial interpolation, spline interpolation and Gaussian processes (**???**). However, situations can occur where it it necessary to interpolate between two or more data clusters instead of individual data points. These situations often entail systems where the outputs or measurements vary with the system operating condition $\alpha \in \mathbb{R}^q$. Before potential applications of such a technique are addressed, the interpolation problem is mathematically defined.

It is assumed that a continuous set of operating conditions is given by $\alpha(\gamma)$, where $\alpha : \mathbb{R}^+ \to \mathbb{R}^q$, $\mathbb{R}^+$ is the set of nonnegative real numbers and γ is scalar. A data cluster of size m associated with an operating condition $\alpha(\gamma)$ is given by $\{x_1(\gamma), x_2(\gamma), \ldots, x_m(\gamma)\}$ or by the columns of the associated matrix,

$$X_\gamma = X(\gamma) \triangleq \left[x_1(\gamma)\ x_2(\gamma) \cdots x_m(\gamma)\right]. \qquad (1)$$

Below, X_γ is always used to denote a data cluster, where γ is a means to order the data clusters. However, it is assumed that in general for $\gamma_i \neq \gamma_j$, the k^{th} column of X_{γ_i} is not correlated with the k^{th} column of X_{γ_j}, i.e., $x_k(\gamma_i)$ does not necessarily have any correspondence with $x_k(\gamma_j)$. It is further assumed that the distribution of these data clusters is uni-modal.

Now, for an ordered set

$$\mathbf{\Gamma} \triangleq \{\gamma_1, \gamma_2, \cdots, \gamma_\ell\}, \ \ \gamma_1 < \gamma_2 < \cdots \gamma_\ell \qquad (2)$$

it is assumed that the corresponding set of data clusters,

$$\mathbf{X} \triangleq \{X_{\gamma_1}, X_{\gamma_2}, \ldots, X_{\gamma_\ell}\} \qquad (3)$$

is known and that the data clusters have the same number of points, i.e.,

$$\dim(X_{\gamma_i}) = \dim(X_{\gamma_j}) \text{ for all } i \text{ and } j. \qquad (4)$$

The interpolation problem is to find an estimate of X_γ for $\gamma \notin \mathbf{\Gamma}$.

The assumption that all data clusters have the same number of points is satisfied by truncating larger data clusters to the size of the smallest data cluster or by adding extra samples to the smaller data clusters through additional experiments. Alternatively, extra samples for the smaller data clusters can be artificially created using the Cholesky Decomposition without changing the covariance of the data cluster (**?**). Also note that the operating condition $\alpha(\gamma)$ can depend on several variables, but the solution derived here will require that the set of known operating conditions can be written in terms of the single scalar variable γ.

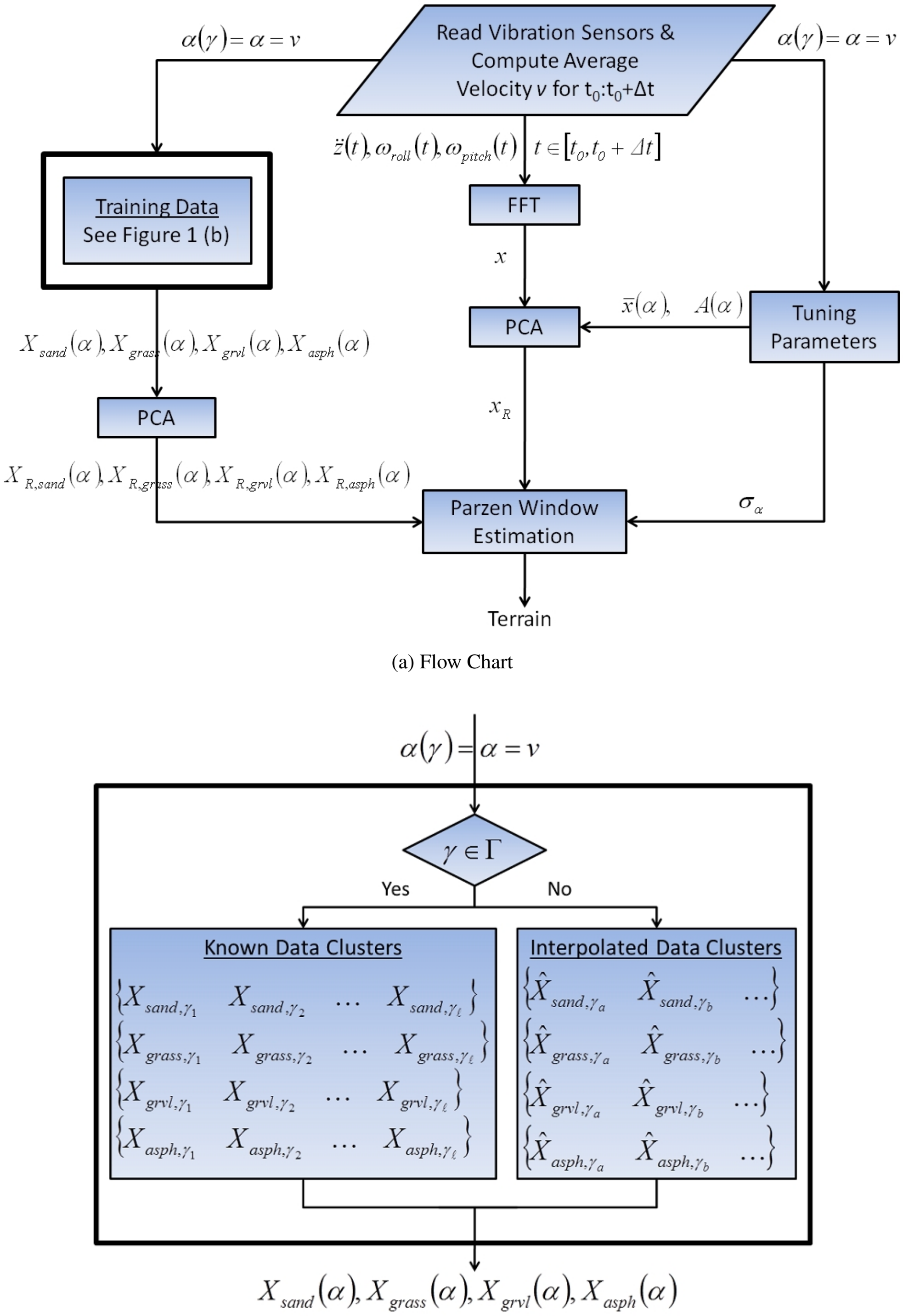
1.1 Applications of data cluster Interpolation

Now that the problem of data cluster interpolation is clearly defined, it is possible to discuss real world systems that fit this description. The application that directly motivated this interpolation research involves reaction-based terrain classification, where the traversed terrain is inferred from proprioceptive sensors. This classification problem is burdened by exhaustive data collection due to the changes in vehicle response when the operating conditions of vehicle speed v and vehicle load w (**???**) are changed. This leads to $\alpha = [v\ w]$. Once multiple observations are collected for a given α, these observations are organized into a training set, i.e., data cluster X, which is stored and used for classification. In order to fully describe the possible outcomes of the system, it becomes necessary to determine a data cluster X for a large number of α, corresponding to the speeds v and loads w that are expected in real-world operation of the vehicle. Typically, the operating conditions are in the set described by: $\underline{v} < v < \overline{v}$ and $\underline{w} < w < \overline{w}$, where $\underline{v}$ and $\underline{w}$ are lower bounds and $\overline{v}$ and $\overline{w}$ are upper bounds determined by the expected operation of the vehicle. High classification accuracy can be achieved as long as data clusters are obtained for the nodes appearing in a sufficiently fine griding of the 2-D rectangle defined by the above inequalities. However, generating these data clusters requires running numerous experiments, which is a burdensome process. Although recent works (**??**) have shown that two different model-based approaches can minimize the speed and load dependency, accurate high fidelity vehicle models are difficult to determine. Instead, it has been suggested that interpolating between a smaller number of data clusters, corresponding to strategically chosen (v, w) can substantially decrease the amount of empirical data collected (**?**). A classification scheme that accomplishes this task is given in Figure 1. This scheme will be applied to the proposed method and is described in detail in Section 3.

As data collection is exhaustive task in many machine learning applications, it is expected that additional machine learning applications may benefit from the presented methodology. These tasks are expected to include pattern classification problems and potentially regression tasks. However, it should be noted that the method presented here assumes a uni-modal distribution, which may or may not be true for classification and regression problems. As the researchers do not have access to data for other tasks, the presented methods are only applied to terrain classification and generated datasets.

Data cluster interpolation is not directly addressed in the pattern classification literature. The problem can be addressed by parameter estimation when instead of discrete classes the features are shared among samples of continuous parameters, such as object detection (**?**). It is possible learn a regressor for the conditional probability $P(X|\alpha)$ (**?**), but this would innately prevent the use of terrain classification approaches that do not explicitly use distributions for identification such as K-nearest neighbor and singular value decomposition.

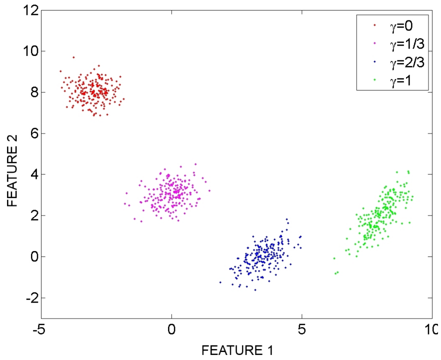
The rest of this paper, which is an extension of the research presented in (**?**), is organized as follows. Section 2, which presents the main results, first describes an approach that can be used to interpolate the data cluster mean $\bar{x}_\gamma$, followed by a method for interpolating the covariance matrix C_γ and an approach for estimating the orthogonal matrix V_γ that appears in the singular value decomposition (SVD) of



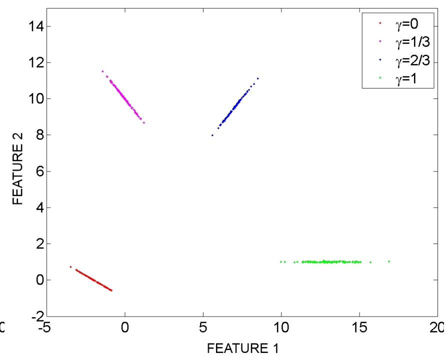
(a) Flow Chart

(b) Training Data Block

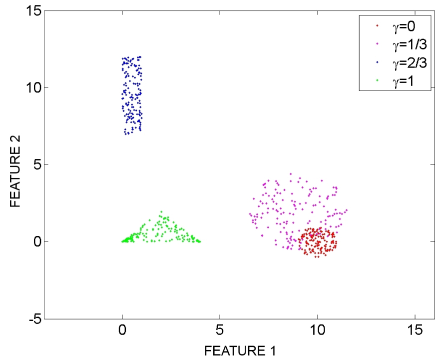
Fig. 1: Flow Chart for Vibration-based Terrain Classification



(a) Gaussian distributions



(b) Small singular values



(c) Distributions of various shapes

Fig. 2: Benchmark interpolation problems (distributions moving from left to right)

a shifted data cluster; together they enable data cluster interpolation. As problems exists for which this covariance interpolation method yields unsatisfactory results, Section 2.3 then describes how to utilize the earlier mean interpolation along with a method of interpolating all of the matrices comprising the SVD of the covariance matrix to perform data cluster interpolation. This methodology is termed here as singular value decomposition interpolation (SVDI). Subsection 2.5.2 evaluates the interpolation algorithms using the three benchmark problems shown in Figure 2, corresponding to the following: (a) data clusters with Gaussian distributions, (b) data clusters whose covariance matrices have one nearly zero singular value, and (c) data clusters of different shapes. Section 3 then evaluates the proposed method on the application that motivated this work using empirical vibration data from a mobile robotic system involving a feature vector in $\mathbb{R}^{300}$. Finally, Section 4 gives concluding remarks.

2 Main Result

Gaussian data clusters are completely characterized by the distribution mean $\bar{x}$, covariance matrix C and orthogonal scatter matrix V which is part of the SVD of data cluster $X = U\Sigma V^T$. Therefore, if the distribution of a data cluster at operating condition $\alpha(\gamma)$, denoted X_γ, can be reasonably approximated with a Gaussian distribution, it is possible to determine an estimate of X_γ, called $\hat{X}_\gamma$, simply by estimating $\bar{x}_\gamma$, C_γ, and V_γ^T, which are respectively the mean vector, covariance matrix and orthogonal scatter matrix of X_γ. In more general terms, this idea is analogous to describing a data cluster's distribution using two characteristics, the cloud location in n-dimensional space and the dispersion of points, while the orthogonal scatter matrix V_γ^T describes a specific set of samples from this distribution. Using this thesis, for $\gamma \notin \mathbf{\Gamma}$ Subsection 2.1 presents a methodology for mean interpolation and Subsection 2.2 presents a methodology for interpolating covariance matrices. Since this approach is seen to have limitations, Subsection 2.3 develops an interpolation method that independently interpolates matrices U, Σ, and V which appear in the SVD of the data cluster shifted so that its mean is zero.

2.1 Mean Interpolation Using Spline Techniques

The mean of the data cluster X_γ given by (1) is defined by

$$\bar{x}_\gamma \triangleq \frac{1}{m}\sum_{i=1}^{m} x_i(\gamma). \tag{5}$$

If $\gamma \in \mathbf{\Gamma}$, then $\bar{x}_\gamma$ is known. Hence, for $\gamma \notin \mathbf{\Gamma}$ finding an estimate $\hat{\bar{x}}_\gamma$ of $\bar{x}_\gamma$ is a vector interpolation problem based on a set of control points $\left\{\bar{x}_{\gamma_1}, \bar{x}_{\gamma_2}, \ldots, \bar{x}_{\gamma_\ell}\right\}$. If it is assumed that $\gamma_k < \gamma < \gamma_{k+1}$, then for linear interpolation only two control points are needed, $\bar{x}_{\gamma_k}$ and $\bar{x}_{\gamma_{k+1}}$. However, it is desired to enforce C^1 continuity, or more generally, smooth the curve connecting the control points, which requires the use of $\bar{x}_{\gamma_{k-1}}$, $\bar{x}_{\gamma_k}$, $\bar{x}_{\gamma_{k+1}}$ and $\bar{x}_{\gamma_{k+2}}$. Note that γ_{k-1} is undefined when $k = 1$, such that $\gamma_1 < \gamma < \gamma_2$. In this case, it is assumed $\gamma_{k-1} = \gamma_k$. Similarly, when $k+1 = \ell$, such that $\gamma_{\ell-1} < \gamma < \gamma_\ell$, it is assumed that $\gamma_{k+2} = \gamma_\ell$.

Using the four control points $\bar{x}_{\gamma_{k-1}}$, $\bar{x}_{\gamma_k}$, $\bar{x}_{\gamma_{k+1}}$ and $\bar{x}_{\gamma_{k+2}}$, the cubic Hermite spline (?) provides the estimate,

$$\begin{aligned}\hat{\bar{x}}_\gamma = (2t^3 - 3t^2 + 1)\bar{x}_{\gamma_k} + (t^3 - 2t^2 + t)\rho_k + \\ (-2t^3 + 3t^2)\bar{x}_{\gamma_{k+1}} + (t^3 - t^2)\rho_{k+1},\end{aligned} \tag{6}$$

where $t \in (0, 1)$ describes the location of $\hat{\bar{x}}_\gamma$ along the path between the control points $\bar{x}_{\gamma_k}$ and $\bar{x}_{\gamma_{k+1}}$ and is given by

$$t = \frac{\gamma - \gamma_k}{\gamma_{k+1} - \gamma_k}, \tag{7}$$

and ρ_k is the tangent vector used to determine the spline direction of departure from the control point $\bar{x}_{\gamma_k}$. Although there are several definitions of ρ_k, one of the more common definitions is that of the Cardinal spline (?), which defines ρ_k as

$$\rho_k = (1 - c)\frac{\bar{x}_{\gamma_{k+1}} - \bar{x}_{\gamma_{k-1}}}{\gamma_{k+1} - \gamma_{k-1}}, \tag{8}$$

where c is known as the tension parameter that controls the length of the tangent vector. If $c = 0$, then ρ_k reduces to that of the *Catmull-Rom spline*,

$$\rho_k = \frac{\bar{x}_{\gamma_{k+1}} - \bar{x}_{\gamma_{k-1}}}{\gamma_{k+1} - \gamma_{k-1}}, \tag{9}$$

which is one of the more popular choices of the tangent vector used in cubic Hermite splines. The reason for this is that the tangents are continuous over the entire curve (i.e. the curve is C^1 continuous), the curve is guaranteed to pass through the control points, and it is considered computationally efficient. For these same reasons, the Catmull-Rom spline is considered an appropriate choice of interpolation method for estimating $\hat{\bar{x}}_\gamma$.

2.2 Interpolation Based on the Covariance Matrix

The covariance matrix C_γ corresponding to the data cluster $X_\gamma \in \mathbb{R}^{n\times m}$ is given by

$$C_\gamma = \frac{1}{n-1}\tilde{X}_\gamma \tilde{X}_\gamma^T, \tag{10}$$

where

$$\tilde{X}_\gamma = X_\gamma - \beta_\gamma, \tag{11}$$

and each column of $\beta_\gamma \in \mathbb{R}^{n\times m}$ is equal to the mean $\bar{x}_\gamma$ defined by (5). Note that $\tilde{X}_\gamma$ may be viewed as the data cluster X_γ shifted so that it is centered at the origin.

The singular value decomposition (SVD) of $\tilde{X}_\gamma$ is given by

$$\tilde{X}_\gamma = U_\gamma \Sigma_\gamma V_\gamma^T, \tag{12}$$

where $U_\gamma \in \mathbb{R}^{n\times n}$ and $V_\gamma \in \mathbb{R}^{m\times m}$ are orthogonal, and $\Sigma_\gamma \in \mathbb{R}^{n\times m}$ is a diagonal matrix. The singular values of $V_\gamma \in \mathbb{R}^{m\times m}$ are $(C_{\gamma,1}, C_{\gamma,2}, \ldots, C_{\gamma,p})$, where $p = \min(m, n)$. Since the rank $r \leq p$ of $\tilde{X}_\gamma$ is equal to the rank of Σ_γ, it can be shown that

$$\begin{aligned}&C_{\gamma,1} \geq C_{\gamma,2} \geq \cdots C_{\gamma,r} > 0 \text{ and} \\ &C_{\gamma,r+1} = C_{\gamma,r+2} = \cdots = C_{\gamma_p} = 0.\end{aligned} \tag{13}$$

Also, realize that U_γ may be viewed as a left rotation matrix that rotates Σ_γ, and V_γ^T can be viewed as a right rotation matrix that rotates $U_\gamma\Sigma_\gamma$. Substituting (12) into (10) yields the SVD of C_γ,

$$C_\gamma = U_\gamma\left(\frac{1}{n-1}\bar{\Sigma}_\gamma^2\right)U_\gamma{}^T, \tag{14}$$

where $\bar{\Sigma}_\gamma \in \mathbb{R}^{n\times n}$ is the square diagonal matrix satisfying,

$$\bar{\Sigma}_\gamma^2 = \Sigma_\gamma \Sigma_\gamma^T. \tag{15}$$

It follows from (14), (12), and (11) that an estimate $\hat{X}_\gamma$ of the data cluster X_γ can be reconstructed if the following estimates are obtained: an estimate $\hat{C}_\gamma$ of the covariance matrix C_γ, an estimate $\hat{V}_\gamma$ of the right singular vector V_γ, and an estimate $\hat{\bar{x}}_\gamma$ of the mean $\bar{x}_\gamma$ (which determines an estimate $\hat{\beta}_\gamma$ of β_γ). In particular, $\hat{X}_\gamma$ is given by

$$\hat{X}_\gamma = \hat{U}_\gamma \hat{\Sigma}_\gamma \hat{V}_\gamma^T + \hat{\beta}_\gamma, \tag{16}$$

where $\hat{U}_\gamma \in \mathbb{R}^{n\times n}$ and $\hat{\Sigma}_\gamma \in \mathbb{R}^{n\times m}$ are obtained by the SVD,

$$\hat{C}_\gamma = \hat{U}_\gamma\left(\frac{1}{n-1}\hat{\bar{\Sigma}}_\gamma^2\right)\hat{U}_\gamma^T, \quad \hat{\bar{\Sigma}}_\gamma \in \mathbb{R}^{n\times n} \tag{17}$$

and

$$\hat{\bar{\Sigma}}_\gamma^2 = \hat{\Sigma}_\gamma \hat{\Sigma}_\gamma^T. \tag{18}$$

2.2.1 Estimation of C_γ

For $\gamma \notin \mathbf{\Gamma}$ the problem of estimating $C_\gamma \in \mathbb{R}^{n\times n}$ is a matrix interpolation problem based on a set of control points $\left\{C_{\gamma_1}, C_{\gamma_2}, \ldots, C_{\gamma_\ell}\right\}$. Assuming that $\gamma_k < \gamma < \gamma_{k+1}$, an estimate $\hat{C}_\gamma$ can be obtained by the linear interpolation,

$$\hat{C}_\gamma = tC_{\gamma_k} + (1 - t)C_{\gamma_{k+1}}, \tag{19}$$

where $t \in (0, 1)$ is given by (7). Note that since C_{γ_k} and $C_{\gamma_{k+1}}$ are nonnegative definite it follows from (19) that $\hat{C}_\gamma$ is nonnegative definite, a necessity for a covariance matrix.

data cluster estimation using the covariance estimation (19) can actually lead to nonsensical results for the case $m < n$ (i.e., there are fewer points in the data clusters than the dimension of the feature space). To see this, first note that, based on the assuming as defined in (4), it follows that it is desirable to seek an estimate $\hat{X}_\gamma \in \mathbb{R}^{n\times m}$, such that $\dim(\hat{X}_\gamma) = \dim(X_{\gamma_k}) = \dim(X_{\gamma_{k+1}})$. It then follows from (16) that $\hat{X}_\gamma \in \mathbb{R}^{n\times m}$ requires $\hat{\Sigma}_\gamma \in \mathbb{R}^{n\times m}$ and that $\hat{\Sigma}_\gamma$ has $p = \min(m, n)$ singular values. Using $\hat{C}_\gamma$ from (19), the singular values of $\hat{\Sigma}_\gamma$ are obtained from (17) and are in fact the n diagonal elements of $\hat{\hat{\Sigma}}_\gamma \in \mathbb{R}^{n\times n}$. If $m < n$, then $p = m$ and hence $n > p$, i.e., there are more singular values of $\hat{\Sigma}_\gamma$ than required. Therefore, unless at least $n - p$ singular values of $\hat{\hat{\Sigma}}_\gamma$ are zero, which will not generally be the case, $\hat{\Sigma}_\gamma$ cannot be defined by (19) rendering the covariance estimation methodology invalid.

2.2.2 Estimation of V_γ

The problem of estimating the orthogonal matrix $V_\gamma \in \mathbb{R}^{m\times m}$ for $\gamma \notin \mathbf{\Gamma}$ is a matrix interpolation problem based on a set of control matrices $\left\{V_{\gamma_1}, V_{\gamma_2}, \ldots, V_{\gamma_\ell}\right\}$. The key to this estimation is given a special orthogonal matrix V, determine a matrix function $V(t)$ on $[0, 1]$ such that $V(0) = I$, $V(1) = V$, and $V(t)$ is an orthogonal matrix for each $t \in [0, 1]$. A key result in accomplishing this goal is the below canonical form for a real, orthogonal matrix which was developed with the aid of (**?**).

Theorem 1 *Let V be a real $n \times n$ orthogonal matrix. Then the eigenvalues of V are of the form 1, −1, or $e^{j\theta}$, where $\theta \in \mathbb{R}$. If the eigenvalues of V are 1 (p times), −1 (r times), and $e^{j\theta_k}$ for $k = 1, \ldots, s$, then V can be written as $V = QDQ^T$ where*

$$D = \begin{bmatrix} I_p & & & & \\ & -I_r & & & \\ & & \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} & & \\ & & & \ddots & \\ & & & & \begin{bmatrix} \cos\theta_s & -\sin\theta_s \\ \sin\theta_s & \cos\theta_s \end{bmatrix} \end{bmatrix}, \quad (20)$$

Q is a real $n \times n$ orthogonal matrix and I_k is a $k \times k$ identity matrix.

Proof See Appendix A.

Since V is a special orthogonal matrix, there must be an even (possibly zero) number of eigenvalues equal to −1. A suitable $V(t)$ is then given by

$$V(t) = QD(t)Q^T, \quad (21)$$

where the block diagonal matrix $D(t)$ is given by

$$D(t) = \mathrm{diag}(I_p, \underbrace{R(\pi t), \ldots, R(\pi t)}_{r \text{ times}}, R(\theta_1 t), \ldots, R(\theta_s t)), \quad (22)$$

and

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (23)$$

The identity matrix I_p in $D(t)$ corresponds to the p eigenvalues of U that equal 1, the $r/2$ rotation matrices $R(\pi t)$ correspond to the r eigenvalues equal to −1, and the remaining rotation matrices $R(\theta_k t)$ correspond to the s complex conjugate pairs of complex eigenvalues. Note that the decomposition matrices Q and D can be computed using the Schur method or block diagonalization techniques which derive from the Schur decomposition of Q (**?**).

It follows that for $\gamma_k < \gamma < \gamma_{k+1}$ an estimate $\hat{V}_\gamma$ can be obtained by the interpolation,

$$\hat{V}_\gamma = Q_{V_{k+1}} D_{V_{k+1}}(t) Q_{V_{k+1}}^T V_{\gamma_k}, \quad (24)$$

for $t \in (0, 1)$ given by (7), where $D_{V_{k+1}}(t)$ and $Q_{V_{k+1}}$ result from the Schur method or block diagonalization method for matrix computations (**?**) applied to $V_{\gamma_{k+1}}$, yielding the canonical form of (20) presented in Theorem 1.

The interpolation (24) has the desired properties,

$$\begin{aligned} \hat{V}_\gamma &= V_{\gamma_k} \text{ for } t = 0, \\ \hat{V}_\gamma &= V_{\gamma_{k+1}} \text{ for } t = 1, \end{aligned} \quad (25)$$

and $\hat{V}_\gamma$ is orthogonal.

The interpolation (24) is an extension of the interpolation method for 3D rotation matrices given in (**?**), which uses matrix logarithms for interpolating these rotation matrices. Another established method for interpolating between 3D rotation matrices is Spherical Linear Interpolation (SLERP), which is often used to animate rotating objects (**?**). However, extending SLERP to higher dimension orthogonal matrices is not as straightforward as the method of (**?**).

2.2.3 Summary of Interpolation Based on the Covariance

Given $\left\{X_{\gamma_1}, X_{\gamma_2}, \ldots, X_{\gamma_\ell}\right\}$ and γ satisfying $\gamma_k < \gamma < \gamma_{k+1}$, the described interpolation methods result in the estimated data cluster $\hat{X}_\gamma$ given by (16), where for t given by (7) each of the matrices appearing in the right hand side of (16) are computed as follows:

1. Each column of $\hat{\beta}_\gamma \in \mathbb{R}^{n\times n}$ is equal to the mean estimate $\hat{\bar{x}}$ given by (6) and (9).
2. $\hat{U}_\gamma$ and $\hat{\Sigma}_\gamma$ result from the SVD of $\hat{C}_\gamma$ using (17) and (18), where $\hat{C}_\gamma$ is obtained from (19).
3. Using the Schur method or block diagonalization techniques, determine $Q_{V_{k+1}}$ and $D_{V_{k+1}}(t)$. $\hat{V}_\gamma^T$ is then obtained from (24).

The remainder of the paper will refer to this method of data cluster interpolation as *Covariance Interpolation* or CovI. CovI is applied to the benchmark problems of (2) below.

2.2.4 Application of Covariance Interpolation to the Benchmark Problems

CovI is specifically designed to interpolate the changes in data cluster shape and feature space location that occur as a result of changing operating conditions $\alpha(\gamma)$, as discussed in the beginning of Section 2. When applying this interpolation technique to the three benchmark problems of Figure 2, it can be seen that CovI is effective when the data cluster shape and/or location changes significantly. However, it also becomes apparent that not all data cluster interpolation problems as defined in Section 1, are strictly based on changes in data cluster shape and/or location. Consider Figures 3, 4 and 5, which display interpolated data clusters for the benchmark problems of Figure 2 at several values of γ.

Figures 3 and 4 respectively show reasonable estimates of the data clusters for the interpolation of the Gaussian distributions and distributions of different shapes. That is, in general an estimated data cluster lies between the known data clusters and the shape of the estimated data cluster becomes that of the known data cluster corresponding to γ_k as $\gamma \to \gamma_k$. This means that when $\gamma \in \{\gamma_1, \gamma_2, \ldots, \gamma_\ell\}$ the estimated data clusters lie directly on top of the known data clusters. The red and magenta data clusters of Figure 4 also indicate the algorithm's ability to handle overlapping data clusters.

Figure 5 shows that when the distribution shape remains constant and only the data cluster orientation changes, CovI can result in interpolated data clusters of a different shape, which is undesirable. As previously mentioned, Gaussian distributions are fully defined by the mean $\bar{x}$, covariance matrix C and orthogonal scatter matrix V^T. Figure 5 shows an example of data clusters that can be closely approximated using a Gaussian distribution, but cannot be intuitively interpolated by estimating these properties through CovI. To understand this phenomena recognize that using arguments similar to those in Subsection 2.2.1 it follows that for $m \geq n$ the covariance estimate (17) results in the estimation of n singular values as desired. Suppose $\text{rank}(X_{\gamma_k}) = r_k < n$ and $\text{rank}(X_{\gamma_{k+1}}) = r_{k+1} < n$. Then it is desired that $\text{rank}(\hat{X}_\gamma)$ is in the set of integers lying between r_k and r_{k+1}, denoted here by $\mathcal{I}_k$, such that $\min \mathcal{I}_k \leq \text{rank}(\hat{X}_\gamma) \leq \max \mathcal{I}_k < n$. However, it is possible that the covariance estimate (17) yields r nonzero singular value estimates, where $r > \max \mathcal{I}_k$. As an example, if $C_{\gamma_k} = \text{diag}(1, 0)$ and $C_{\gamma_{k+1}} = \text{diag}(0, 1)$ such that $r_k = r_{k+1} = 1$, then (17) yields $r = 2$. This increase in rank occurs in Figure 5, where for $i = 1, \ldots 4$ $\text{rank}(X_{\gamma_i}) = 1$, but for $\gamma \notin \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ the interpolation always yields $\text{rank}(\hat{X}_\gamma) = 2$.

Despite the above limitations, if the change in orientation is small, i.e. the distance between the space generated by the eigenvectors of U_{γ_k} and the space generated by the eigenvectors of $U_{\gamma_{k+1}}$ is small, which can be assumed as $\gamma_k \to \gamma_{k+1}$, then CovI may be an appropriate method of data cluster interpolation. This is evidenced by the data cluster estimate for $\gamma = 0.2$ shown in Figure 5b, which interpolates between the data clusters at $\gamma_1 = 0$ and $\gamma_2 = 1/3$, which have similar orientations.

Recall that subsection 2.2.1 revealed that CovI will fail for $m < n$. When combined with the discussion of Figure 5, which revealed that CovI may not preserve the data cluster shape when $m \geq n$ and significant orientation changes occur, it is evident that CovI has two important deficiencies. These deficiencies led to the development of an alternative method, based on interpolating all of the matrices appearing in the SVDs of the normalized data cluster $\tilde{X}_\gamma$.

2.3 Interpolation Based on Singular Value Decomposition

CovI is based on estimating the covariance matrix $\hat{C}_\gamma$ and using (17) and (18) to extract the singular value matrix $\hat{\Sigma}_\gamma$ along with $\hat{U}_\gamma$. The problems with this method ultimately result from its lack of control over the singular values contained in $\hat{\Sigma}_\gamma$. Hence, this subsection develops an interpolation approach that achieves the desired control by independently estimating the SVD matrices $\hat{U}_\gamma$, $\hat{\Sigma}_\gamma$, and $\hat{V}_\gamma$, appearing in (16) (i.e., $\hat{X}_\gamma = \hat{U}_\gamma \hat{\Sigma}_\gamma \hat{V}_\gamma^T + \hat{\beta}_\gamma$). An estimate of $\hat{V}_\gamma$ has already been presented in Subsection 2.2.2. It remains to present methodologies for computing $\hat{U}_\gamma$ and Σ_γ.

2.4 Estimation of $\hat{U}_\gamma$

Since both U_γ and V_γ are orthogonal matrices, the methodology for computing the estimate $\hat{U}_\gamma$ is essentially identical to that for $\hat{V}_\gamma$. Hence, based on the results of Subsection 2.2.2,

$$\hat{U}_\gamma = Q_{U_{k+1}} D_{U_{k+1}}(t) Q_{U_{k+1}}^T U_{\gamma_k}, \tag{26}$$

for $t \in (0, 1)$ given by (7), where $D_{U_{k+1}}(t)$ and $Q_{U_{k+1}}$ result from writing $U_{\gamma_{k+1}}$ in the canonical form presented in Theorem 1. The properties of (26) are identical to those of (24), which are discussed in Subsection 2.2.2.

It should be noted however that matrices U_γ and V_γ^T of a SVD $\tilde{X}_\gamma = U_\gamma \Sigma_\gamma V_\gamma^T$ are not unique due to the non-uniqueness of the signs of the columns of U_γ and V_γ. For example, $\tilde{X}_\gamma = (-U_\gamma)\Sigma_\gamma(-V_\gamma^T)$ is also a valid SVD. This means, that when individually estimating U_γ and V_γ, it is important to intelligently choose which SVD of X_{γ_k} and $X_{\gamma_{k+1}}$ will be used in estimating U_γ and V_γ, respectively using (26) and (24). Consider that if, γ_k and γ_{k+1} are close together, the angle between the *ith* column of U_{γ_k} and the *ith* column of $U_{\gamma_{k+1}}$ will be small. Therefore, it is assumed that the best choice of U_{γ_k} and $U_{\gamma_{k+1}}$ will minimize this angle. For this reason Algorithm 1, which reduces the angle between the *ith* column of U_{γ_k} and the *ith* column of $U_{\gamma_{k+1}}$ for $i = [1, \min(m, n)]$, is presented. Once this algorithm has

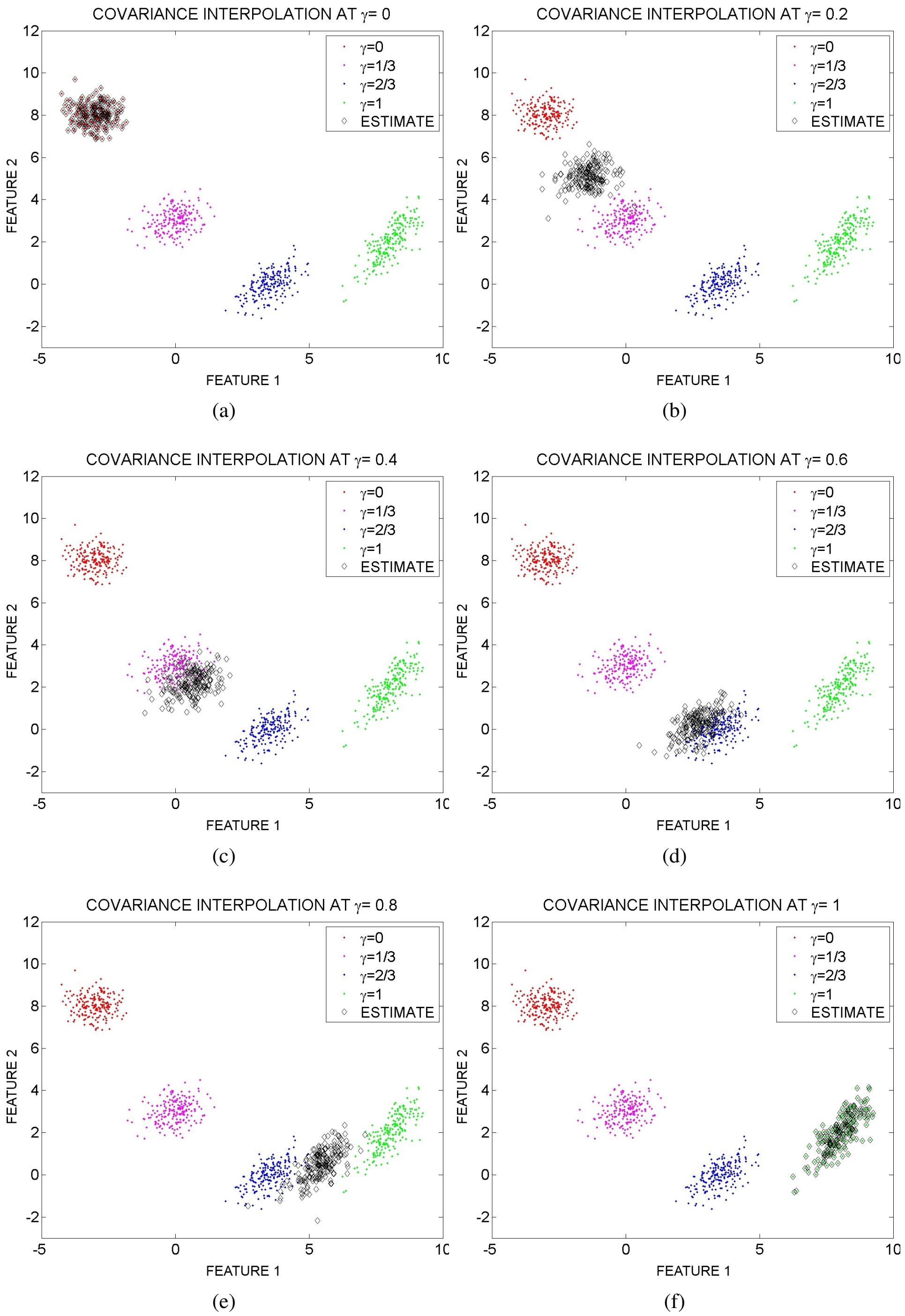


Fig. 3: Covariance Interpolation Method Results for the Benchmark Problem Involving Gaussian Distributions

concluded, $\hat{U}_\gamma$ and $\hat{V}_\gamma^T$ can then be estimated using (26) and (24).

2.5 Estimation of $\hat{\Sigma}_\gamma$

Recall from (12) that for $\gamma \in \mathbf{\Gamma}$ $\Sigma_\gamma \in \mathbb{R}^{n\times m}$ is a diagonal matrix with p singular values. Hence, for $\gamma \in (\gamma_k,\ \gamma_{k+1})$ the estimate $\hat{S}_\gamma$ can be achieved by using either linear interpolation of the p diagonal elements of Σ_{γ_k} and $\Sigma_{\gamma_{k+1}}$ or cubic

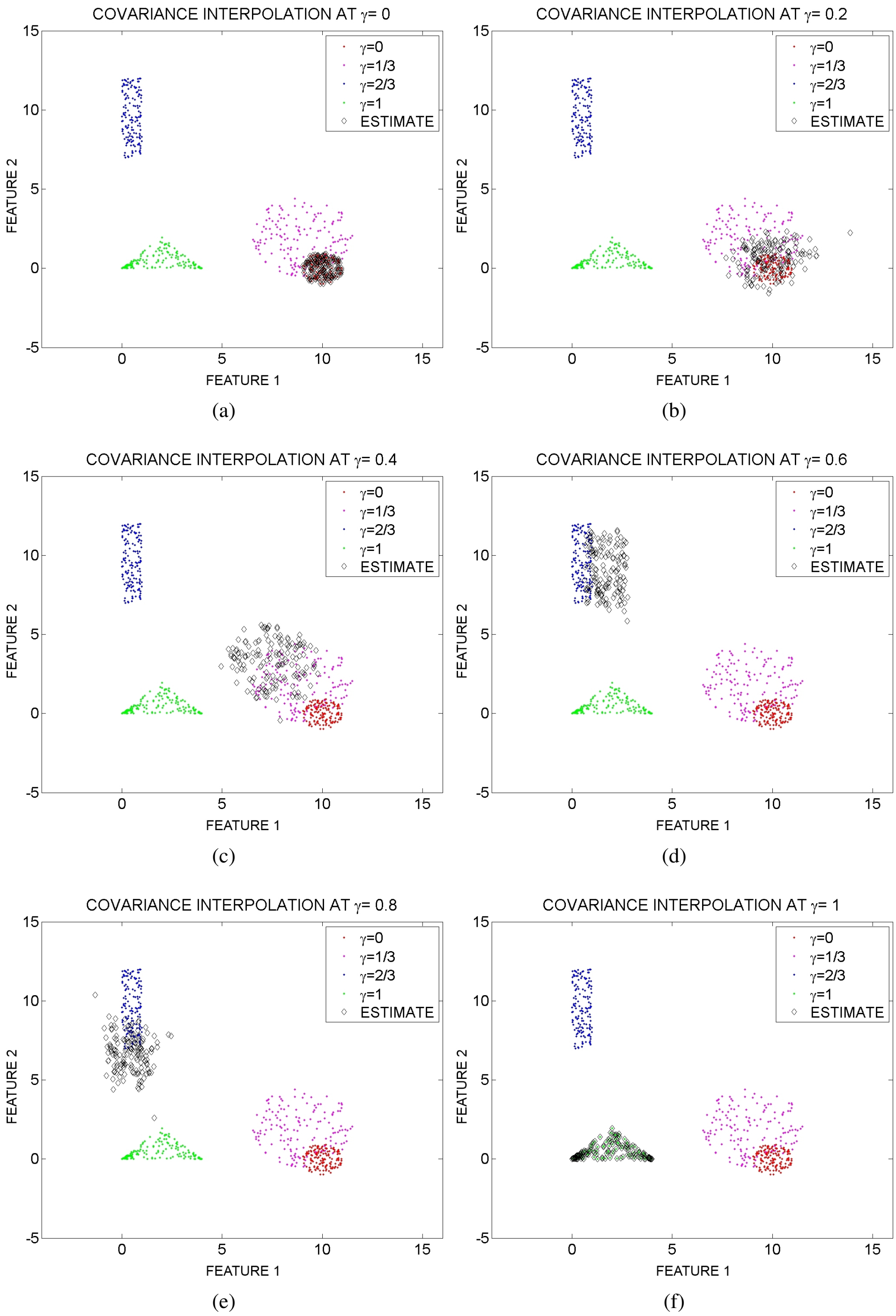


Fig. 4: Covariance Interpolation Method Results for the Benchmark Problem Involving Distributions of Different Shapes

spline interpolation using the p diagonal elements of $\Sigma_{\gamma_{k-1}}$, Σ_{γ_k}, $\Sigma_{\gamma_{k+1}}$, and $\Sigma_{\gamma_{k+2}}$.

For $\gamma \in \mathbf{\Gamma}$ define $\sigma_\gamma \in \mathbb{R}^p$ such that the i^{th} element corresponds to the i^{th} singular value of Σ_γ. An estimate $\hat{\sigma}_\gamma$ determines the singular values of $\hat{\Sigma}_\gamma$, the estimate of the singular value matrix . For t defined by (7) linear interpolation yields

$$\hat{\sigma}_\gamma = (1-t)\sigma_{\gamma_k} + t\sigma_{\gamma_{k+1}}, \tag{27}$$

and Cardinal spline interpolation yields

$$\begin{aligned}\hat{\sigma}_\gamma = (2t^3 - 3t^2 + 1)\sigma_{\gamma_k} + (t^3 - 2t^2 + t)\lambda_k + \\ (-2t^3 + 3t^2)\sigma_{\gamma_{k+1}} + (t^3 - t^2)\lambda_{k+1},\end{aligned} \tag{28}$$

where

$$\lambda_k = (1-c)\frac{\sigma_{\gamma_{k+1}} - \sigma_{\gamma_{k-1}}}{\gamma_{k+1} - \gamma_{k-1}}, \tag{29}$$

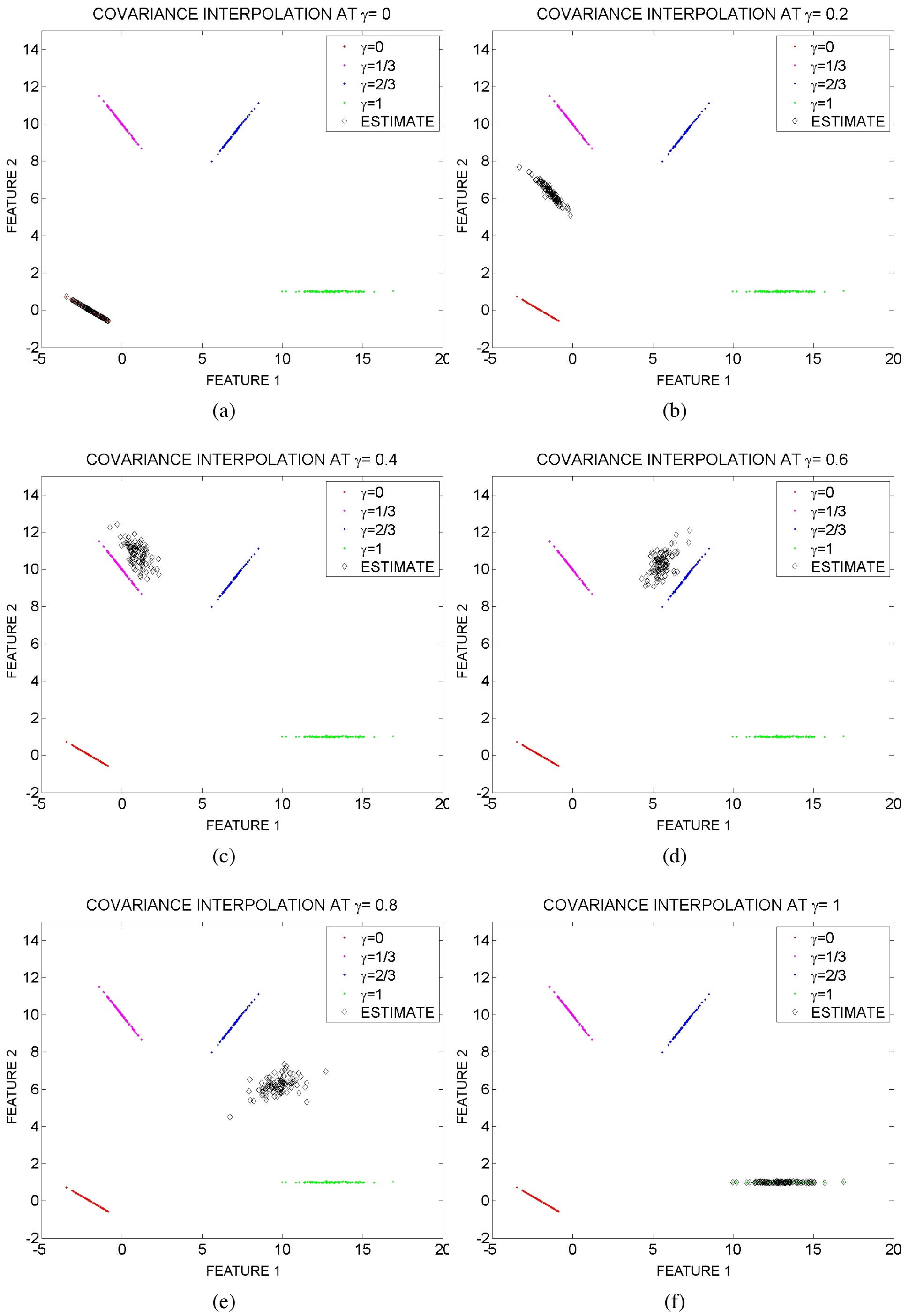


Fig. 5: Covariance Interpolation Method Results for the Benchmark Problem Involving Distributions with Small Singular Values

Algorithm 1 Finding singular value decompositions of X_{γ_k} and $X_{\gamma_{k+1}}$ that reduces the angular change between $U_{\gamma_k}(:,i)$, the *ith* column of U_{γ_k} and $U_{\gamma_{k+1}}(:,i)$, the *ith* column of $U_{\gamma_{k+1}}$

```
svd(X̃_{γ_k}) = U_{γ_k} Σ_{γ_k} V_{γ_k}^T
svd(X̃_{γ_{k+1}}) = U_{γ_{k+1}} Σ_{γ_{k+1}} V_{γ_{k+1}}^T
while i ≤ min(m, n) do
    α = cos^{-1}( U_{γ_k}(:,i)^T U_{γ_{k+1}}(:,i) / (|U_{γ_k}(:,i)||U_{γ_{k+1}}(:,i)|) )
    if α > π/2 then
        U_{γ_k}(:,i) = −U_{γ_k}(:,i)
        V_{γ_k}(:,i) = −V_{γ_k}(:,i)
    end if
end while
return U_{γ_k}, U_{γ_{k+1}}, Σ_{γ_k}, Σ_{γ_{k+1}}, V_{γ_k}^T, V_{γ_{k+1}}^T
```

and c is the tension parameter. It is imperative that each element of $\hat{\sigma}_\gamma$ is nonnegative. Also, if the number of nonzero elements of σ_{γ_k} and $\sigma_{\gamma_{k+1}}$ are respectively r_k and r_{k+1} then r, the number of nonzero elements of $\hat{\sigma}_\gamma$, should be in the range that spans $[r_k,\ r_{k+1}]$. These two properties are ensured by (27) but can only be ensured by (29) as $c \to 1$. Hence, in this research (27) is used to estimate $\hat{\Sigma}_\gamma$.

2.5.1 *Summary of Interpolation Based on SVD*

Ultimately, given $\left\{X_{\gamma_1}, X_{\gamma_2}, \ldots, X_{\gamma_\ell}\right\}$ and γ satisfying $\gamma_k < \gamma < \gamma_{k+1}$, the described interpolation methods for $\hat{U}_\gamma$, $\hat{\Sigma}_\gamma$, $\hat{V}_\gamma$ and $\hat{\beta}_\gamma$ result in the estimated data cluster $\hat{X}_\gamma$ given by (16), where for t given by (7) each of the matrices appearing in the right had side of (16) are computed as follows:

1. Each column of $\hat{\beta}_\gamma \in \mathbb{R}^{n\times n}$ is equal to the mean estimate $\hat{\bar{x}}$ given by (6) and (9).
2. An SVD $X_{\gamma_k} = U_{\gamma_k}\Sigma_{\gamma_k}V_{\gamma_k}^T$ and $X_{\gamma_{k+1}} = U_{\gamma_{k+1}}\Sigma_{\gamma_{k+1}}V_{\gamma_{k+1}}^T$ are computed using Algorithm 1.
3. Using the Schur method or block diagonalization techniques, determine $Q_{U_{k+1}}$ and $D_{U_{k+1}}(t)$. $\hat{U}_\gamma$ is then obtained from (26).
4. $\hat{\Sigma}_\gamma$, which along with $\hat{U}_\gamma$ describes the covariance matrix of $\hat{X}_\gamma$ (see (17)), results from (27).
5. Using the Schur method or block diagonalization techniques, determine $Q_{V_{k+1}}$ and $D_{V_{k+1}}(t)$. $\hat{V}_\gamma^T$ is then obtained from (24).

The interpolation technique described above is termed *Singular Value Decomposition Interpolation* (SVDI). SVDI is similar to the previously described *Covariance Interpolation* method (CovI), but it is shown below that SVDI succeeds in overcoming the issues associated with CovI.

2.5.2 *Application of SVD Interpolation to the Benchmark Problems*

To evaluate SVDI, consider again the benchmark problems originally presented in Figure 2. As SVDI was formulated to solve the problem associated with distributions that change orientation instead of distribution, consider first the benchmark problem of Figure 2b where the data clusters have a near zero singular value. Applying SVDI to this benchmark problem results in Figure 6.

Figure 6 clearly shows an improvement over the estimated data clusters given by CovI (see Figure 5), by allowing for changes in orientation. At all γ, SVDI yields an estimated data cluster with a near zero singular value, which is the same characteristic displayed in the known data clusters for this system. Additionally, the estimated data clusters always approach the known data clusters when $\gamma \to \gamma_k$ and lie directly on top of known data clusters when $\gamma \in \mathbf{\Gamma}$. Although Figure 6 shows that SVDI is effective in cases where the data cluster orientation changes, SVDI should also be able to determine reasonable data cluster estimates when the data cluster shape changes instead of orientation as in the benchmarks of Figures 2a and 2c, corresponding respectively to data clusters with Gaussian distributions and data clusters of different shapes.

Consider Figures 7 and 8, which are respectively the result of applying SVDI to the Figure 2a and Figure 2c benchmarks. Upon inspection, it can be seen that Figure 7 and 8 are highly similar to the corresponding result for the CovI method shown in Figures 3 and 4, with barely any visually recognizable differences. However, there are sometimes small differences in orientation or shape when comparing estimated data clusters from CovI and SVDI. This small change in orientation can be seen by comparing Figure 8b and Figure 4b while a small change in shape can be seen by comparing Figure 7b and Figure 3b. It is believed that these subtle differences occur as a result of SVDI independently interpolating values that describe the data cluster orientation and shape, while CovI couples orientation and shape estimation into an estimate of the covariance matrix.

3 Interpolation of data clusters in Vibration-Based Terrain Classification

Although SVDI has proven effective on all of the benchmark problems, the true test of its effectiveness will be seen in estimates of data clusters for physical systems. Recall from the discussion of reaction-based terrain classification in Subsection 1.1 that the problem motivating this work has speed dependency. Here, SVDI will be tested on this problem using vibration data from the ATRV-Jr mobile robot, shown in Figure 10.

In vibration-based terrain classification the frequency response magnitudes of the vertical acceleration $\ddot{z}(t)$, roll rate $\omega_{roll}(t)$ and pitch rate $\omega_{pitch}(t)$, which are the vibration signals, during a time interval $[t_0, t_0 + \Delta t]$ compose the feature vector x. This is computed using a Fast Fourier Transform

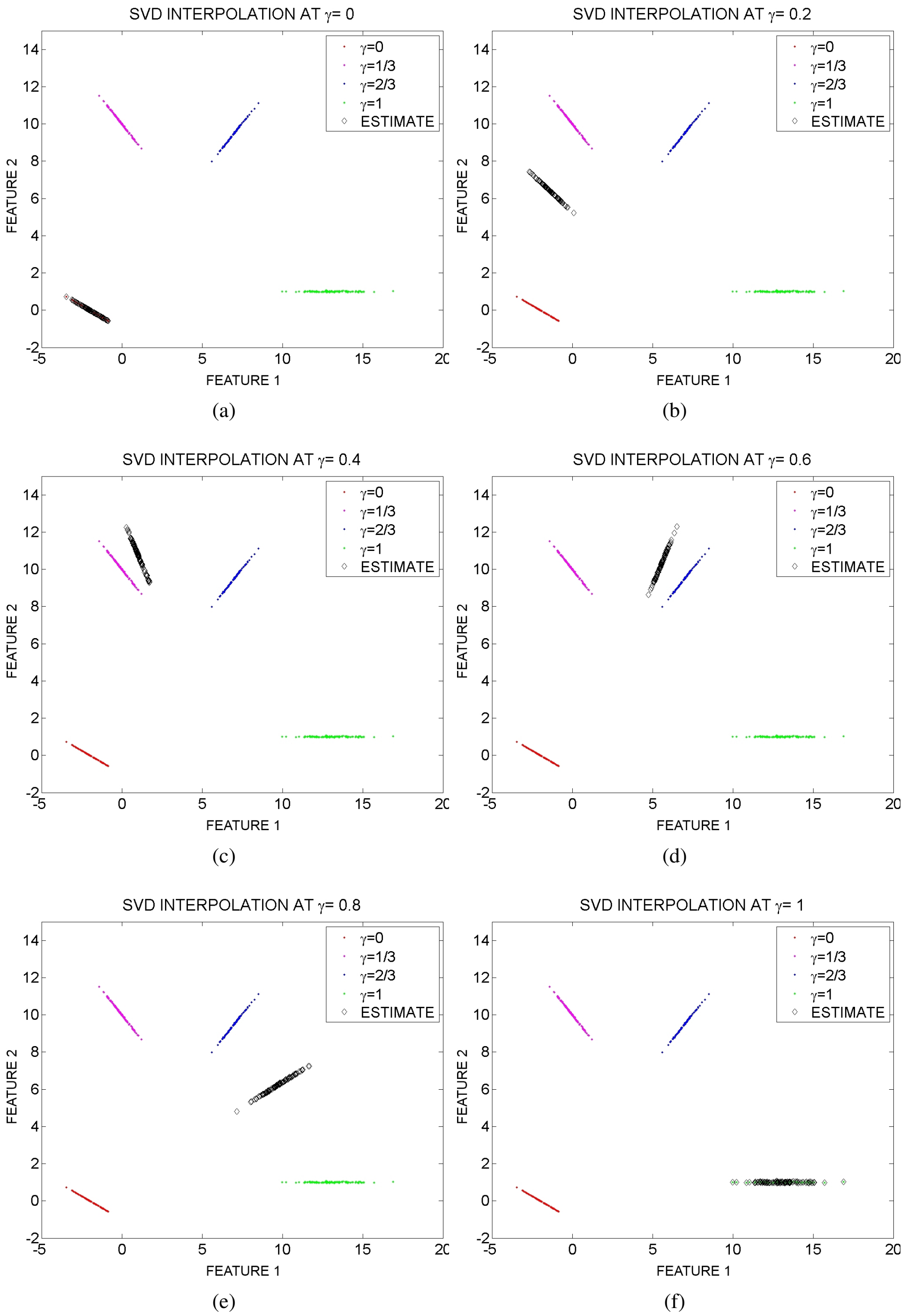


Fig. 6: Singular Value Decomposition Interpolation Method Results for the Benchmark Problem Involving Distributions with Small Singular Values

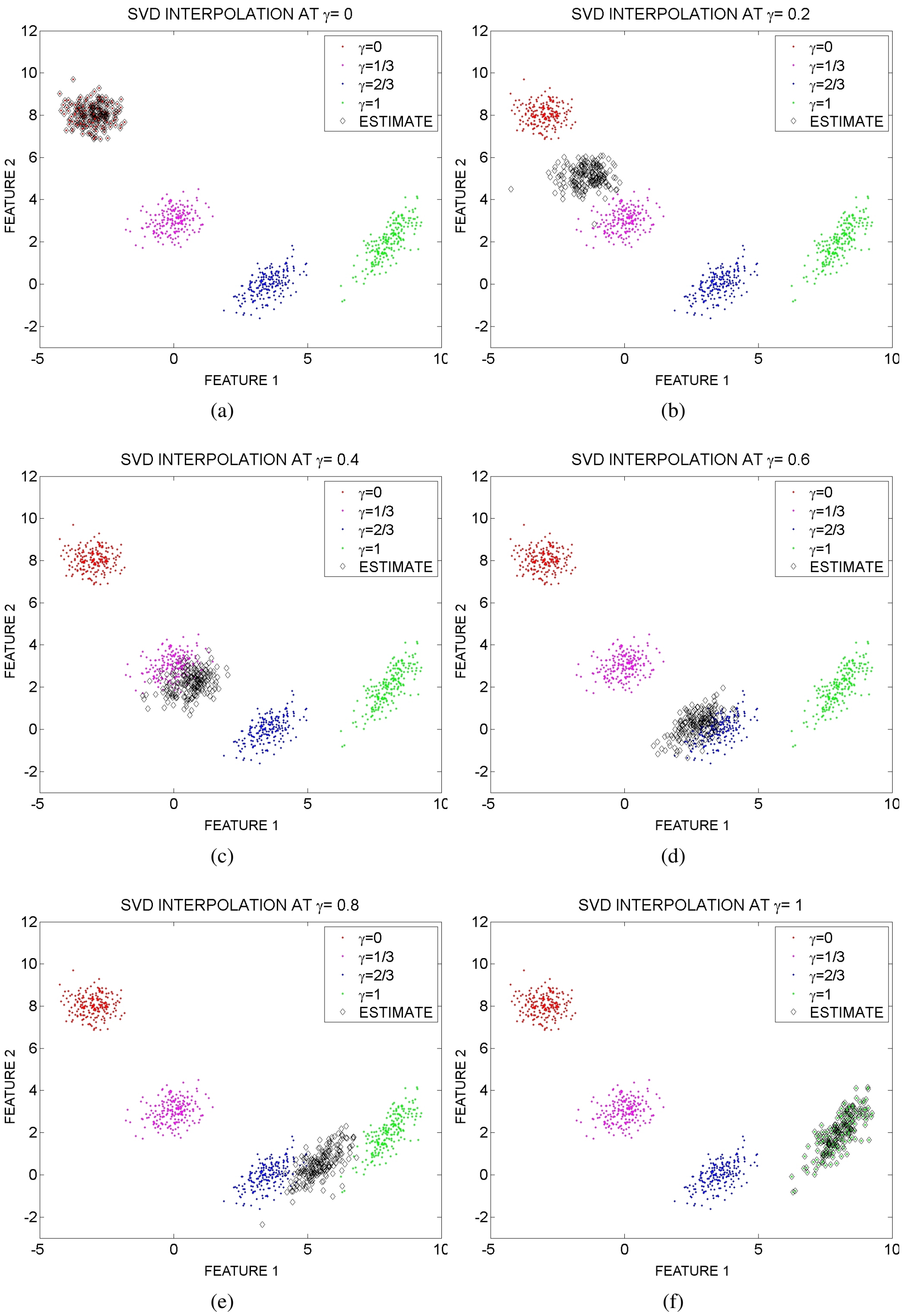


Fig. 7: Singular Value Decomposition Interpolation Method Results for the Benchmark Problem Involving Gaussian Distributions

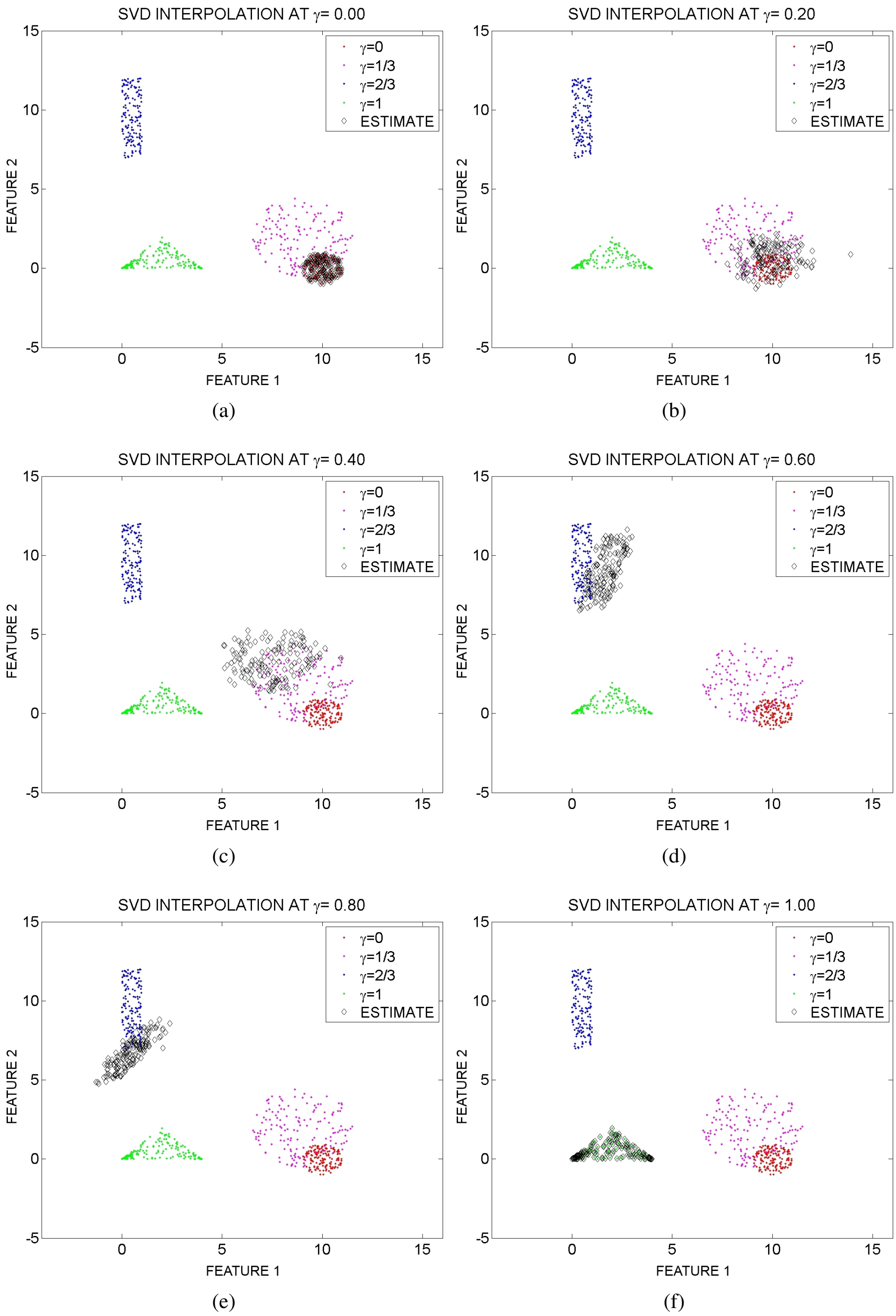


Fig. 8: Singular Value Decomposition Interpolation Method Results for the Benchmark Problem Involving Distributions of Different Shapes

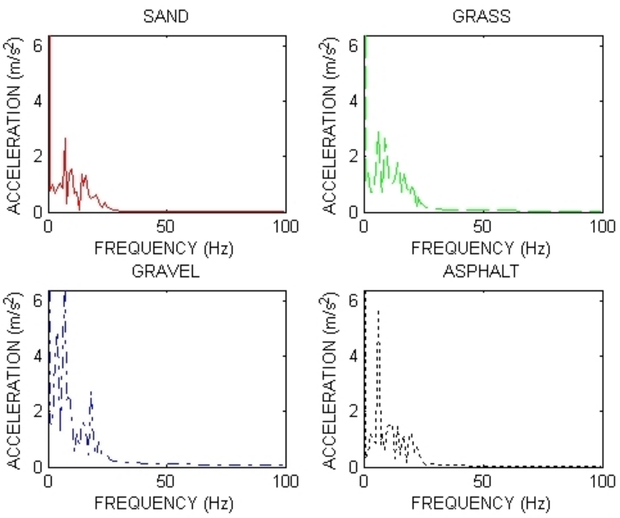


Fig. 9: Examples of the Vertical Acceleration Frequency Response Magnitudes for Sand, Grass, Gravel and Asphalt

(a) ATRVJr

(b) DMU

Fig. 10: The ATRV-Jr mobile robot and Dynamic Measurment Unit (DMU)

(FFT) and results in a feature vector x defined by

$$x = \left[|\ddot{z}(j\omega)|, |\omega_{roll}(j\omega)|, |\omega_{pitch}(j\omega)|\right]$$

and examples of the differences of each terrain's vertical acceleration $|\ddot{z}(j\omega)|$ are shown in Figure 9.

Data from the DMU mounted on the ATRV-Jr was collected at a rate of 200 Hz while driving over four terrains (sand, grass, gravel and asphalt) and at speeds of 0.4, 0.5, 0.6, 0.8, 1.0, 1.2, and 1.4 m/s. The data was then cut into 1 second intervals, which resulted in a 300 dimensional feature vector corresponding to the 1 second interval ($|\ddot{z}(j\omega)|$, $|\omega_{pitch}(j\omega)|$ and $|\omega_{roll}(j\omega)|$ each have 100 elements below the 100 Hz Nyquist frequency). In total, 100 feature vectors were collected at each of the considered combinations of speed and terrain. The classification scheme that was used for evaluating SVDI is presented in Figure 1 and described below.

Speed is the only operating condition that varied during these experiments, meaning $\alpha(\gamma) = v$, where v is the vehicle speed. This leads to the classification scheme of Figure 1 where an observation x is classified using either "Known data clusters" from each terrain, or "Interpolated data clusters" from each terrain. The set of known data clusters are given the label $\left\{X_{terrain,\gamma_1}, X_{terrain,\gamma_2}, \ldots, X_{terrain,\gamma_\ell}\right\}$ and correspond to $\gamma \in \mathbf{\Gamma}$, while the interpolated data clusters are labeled $\left\{\hat{X}_{terrain,\gamma_a}, \hat{X}_{terrain,\gamma_b}, \ldots\right\}$ and correspond to $\gamma \notin \mathbf{\Gamma}$. The tuning parameters are determined using a leave one out training routine at each γ considered in the "Known data clusters" or "Interpolated data clusters" block, yielding a different set of tuning parameters for each possible speed $\alpha(\gamma)$. These training parameters are the average feature vector $\bar{x}(\alpha)$, the transformation matrix $A(\alpha)$ that transforms each $X_{terrain}(\alpha)$ or $\hat{X}_{terrain}(\alpha)$ into the reduced dimension data cluster $X_{R,terrain}(\alpha)$ or $\hat{X}_{R,terrain}(\alpha)$, and the width of the Parzen window $C(\alpha)$. An observation x is then classified using the tuning parameters and data clusters that correspond to the same α as x. All classification results presented in this section correspond to 10-fold cross-validation for determining these training parameters to avoid over-fitting.

When testing SVDI on the ATRV-Jr data, the "Known data clusters" correspond to $\alpha(0) = 0.4$ m/s, $\alpha(0.2) = 0.6$ m/s, $\alpha(0.6) = 1.0$ m/s and $\alpha(1.0) = 1.4$ m/s and the set of known data clusters for a given *terrain* are denoted $\mathbf{X}_{terrain} = \{X_{terrain,0}, X_{terrain,0.2}, X_{terrain,0.6}, X_{terrain,1.0}\}$. Using SVDI and known data clusters $\mathbf{X}_{terrain}$, the "Interpolated data clusters" are then computed at $\gamma_a = 0.1$, $\gamma_b = 0.4$ and $\gamma_c = 0.8$ corresponding to $\alpha(\gamma_a) = 0.5$ m/s, $\alpha(\gamma_b) = 0.8$ m/s and $\alpha(\gamma_c) = 1.2$ m/s for each terrain. The empirical data at $\alpha = 0.5$ m/s, $\alpha = 0.8$ m/s and $\alpha = 1.2$ m/s is then classified using the classification scheme of Figure 1 in order to show the effectiveness of SVDI (this is labeled as benchmark results in Figure 11). The same empirical data is also classified using this algorithm, when the "Interpolated data clusters" block corresponds to an empty set, forcing the x to be classified using "Known data clusters", that do not correspond to the same α as x (the known data clusters at a speed closest to α are used). This second classification result represents the best classification accuracy that can be achieved without interpolation or collecting additional samples. As a means of comparison, benchmark classification results are also presented for the case where additional experiments are conducted. This corresponds to "Known data clusters" $\mathbf{X}_{terrain} = \{X_{terrain,0.1}, X_{terrain,0.4}, X_{terrain,0.8}\}$ and where the "Interpolated data cluster" block is an empty set. All three sets of classification results are given in Figure 11. Note that CovI could not be applied to this problem since there are more features in a single observation than there are samples in the known data clusters ($m = 100 < n = 300$).

As seen in Figure 11, SVDI outperforms what can be achieved without interpolation, in terms of classification accuracy, at all of the considered speeds. The difference ranges from a 2% improvement at 0.5 m/s to an 8% improvement at 1.2 m/s. Overall, SVDI is shown to be 5% more effective than what can be achieved without interpolation. However,

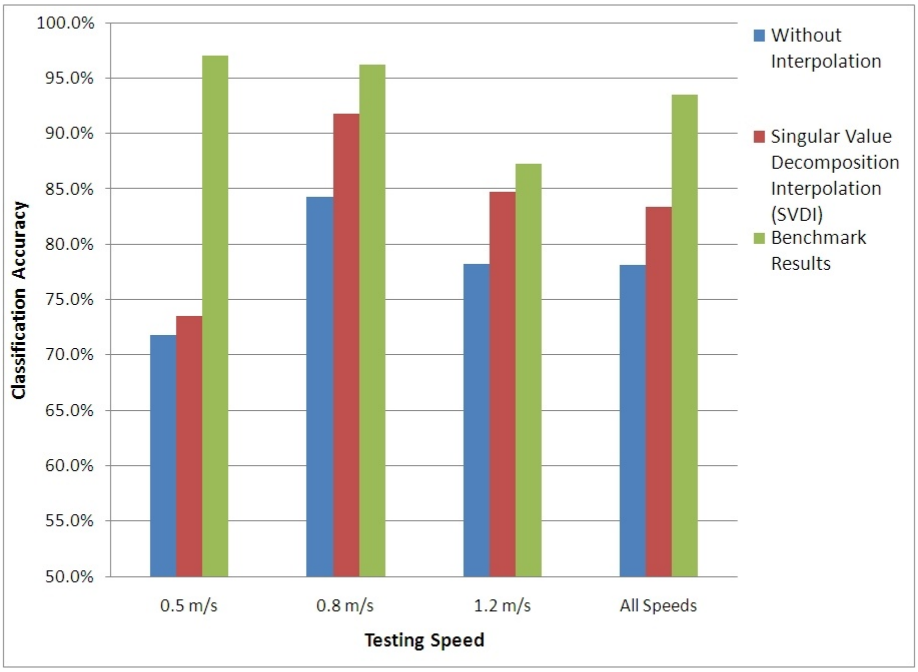


Fig. 11: Accuracy Results for Vibration-Based Terrain Classification on the ATRV-Jr Mobile Robot

from Figure 11 it is obvious that for the best possible results, the vibration-based terrain classification algorithm needs to be tested and trained with empirical data at the same speeds. Ultimately, this means that if training data does not exist at the same speed where testing occurs, it is better to use SVDI to estimate data clusters used for algorithm training, than to simply classify the test samples using algorithms based on neighboring data clusters.

4 Conclusion

This paper presents a method for estimating data clusters termed *Singular Value Decomposition Interpolation* (SVDI) that uses splines, singular value decomposition, Schur decomposition and a special matrix interpolation function. The method is designed to determine data cluster estimates that can be used in machine learning algorithms when real data cannot be collected. Using three benchmark problems, the SVDI method is shown to yield intuitive data cluster estimates with acceptable distribution, orientation and location in the feature space. Furthermore, SVDI is shown to be beneficial when applied to a real machine learning application using empirical data and compares favorably to known techniques. Although the evidence in this paper shows that interpolated data clusters are not as effective as real data, interpolated data clusters are shown here to be more effective than known data clusters at neighboring operating conditions.

Future work on SVDI, should eliminate the requirement that the data cluster estimate and known data clusters all have an equal number of samples, as this can cause some data to be ignored. Additionally, it is desired to implement this technique on several real systems other than the vibration-based terrain classification system presented here. Although, the current form of SVDI smooths the path of the data cluster mean, the path of individual points can be discontinuous, since only X_{γ_k} and $X_{\gamma_{k+1}}$ are used to estimate U_γ, Σ_γ, and V_γ^T. However, it is believed that SVDI may be extended to consider four control matrices $\{X_{\gamma_{k-1}}, X_{\gamma_k}, X_{\gamma_{k+1}}, X_{\gamma_{k+2}}\}$ instead of only X_{γ_k} and $X_{\gamma_{k+1}}$. That is, it may be possible to estimate the U_γ, Σ_γ, and V_γ^T matrices using $X_{\gamma_{k-1}}$, X_{γ_k}, $X_{\gamma_{k+1}}$ and $X_{\gamma_{k+2}}$. This should allow for smoothing of the data cluster orientation and shape by smoothing the path of individual points, which is expected to improve the accuracy of the data cluster estimate.

Acknowledgements This work was prepared through collaborative participation in the Robotics Consortium which is sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2- 0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Funding for this research also provided by the National Science Foundation, Project EEC-0540865.

A Proof of Theorem 1

Referring to Theorem 1 of Section 2.2.2, which presented the canonical form of (20), it is well known that the eigenvalues of a real, orthogonal matrix have unit magnitude so the first part of the theorem concerning the form of the eigenvalues is clear. Before proving the remainder of the theorem, two simple lemmas on complex vectors are presented. The first lemma concerns an effect that scalar multiplication can have on the dot product of the real and imaginary parts of a complex vector.

Lemma 1 *Given a vector* $\mathbf{w} \in C^n$*, there exists a scalar* $z \in C$ *such that the real and imaginary parts of* $z\mathbf{w}$ *are perpendicular to each other.*

Proof For notational convenience, $z = a + jb$ will be used and $\mathbf{w} = \mathbf{u} + j\mathbf{v}$ where $a, b \in \mathcal{R}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{R}^n$. Then $z\mathbf{w} = a\mathbf{u} - b\mathbf{v} + j(b\mathbf{u} + a\mathbf{v})$, and the dot product of the real and imaginary parts of $z\mathbf{w}$ is

$$(a\mathbf{u} - b\mathbf{v}) \cdot (b\mathbf{u} + a\mathbf{v}) = a^2\mathbf{u} \cdot \mathbf{v} + ab(||\mathbf{u}||^2 - ||\mathbf{v}||^2) - b^2\mathbf{u} \cdot \mathbf{v}. \quad (30)$$

The goal is to find a and b so that this dot product is zero. If $\mathbf{u} \cdot \mathbf{v} = 0$, then $z = 1$ will suffice; otherwise set the real part $a = 1$ and choose $b \in \mathcal{R}$ so that $\mathbf{u} \cdot \mathbf{v} b^2 + (||\mathbf{v}||^2 - ||\mathbf{u}||^2)b - \mathbf{u} \cdot \mathbf{v} = 0$. This can be done precisely when the discriminant is nonnegative, i.e., when $(||\mathbf{v}||^2 - ||\mathbf{u}||^2)^2 + 4(\mathbf{u} \cdot \mathbf{v})^2 \geq 0$, which is clearly true. This yields a suitable complex scalar $z = a + jb$.

The next lemma concerns the nature of the complex eigenvectors of an orthogonal matrix when the real and imaginary parts are perpendicular to each other.

Lemma 2 *Suppose that* λ *is a non-real eigenvalue of an orthogonal matrix* U *and that* $\mathbf{w} = \mathbf{u} + j\mathbf{v}$ *is a corresponding eigenvector, where the real vectors* $\mathbf{u}$ *and* $\mathbf{v}$ *are perpendicular to each other. Then* $||\mathbf{u}|| = ||\mathbf{v}||$.

Proof The scalar λ has the form $\lambda = e^{j\theta} = \cos\theta + j\sin\theta$ where $\sin\theta \neq 0$. By equating the real parts of the expressions $U\mathbf{w} = U\mathbf{u} + jU\mathbf{v}$ and

$$U\mathbf{w} = (\cos\theta + j\sin\theta)(\mathbf{u} + j\mathbf{v}) = \cos\theta\mathbf{u} - \sin\theta\mathbf{v} + j(\sin\theta\mathbf{u} + \cos\theta\mathbf{v}) \quad (31)$$

and using the facts that $||U\mathbf{u}|| = ||\mathbf{u}||$ and $\mathbf{u} \cdot \mathbf{v} = 0$, the result $||\mathbf{u}||^2 = ||U\mathbf{u}||^2 = \cos^2\theta||\mathbf{u}||^2 + \sin^2\theta||\mathbf{u}||^2$ is obtained, which further simplifies to $\sin^2\theta(||\mathbf{u}||^2 - ||\mathbf{v}||^2) = 0$. Equating the imaginary parts yields the same expression. Since $\sin\theta \neq 0$, it follows that $||\mathbf{u}||^2 = ||\mathbf{v}||^2$, which proves the result.

Theorem 1 can now be proven.

Proof Once again, it is well known that the eigenvalues of a real, orthogonal matrix have unit magnitude, i.e., the eigenvalues have the form 1, −1, or $e^{j\theta}$. Suppose that $\mathbf{q}$ is a unit length eigenvector of U corresponding to 1 or −1. Letting Q_1 be an orthogonal matrix with $\mathbf{q}$ as its first column, then $Q_1^T U Q_1$ is an orthogonal matrix whose first column equals $[\pm 1 \quad 0 \quad \cdots \quad 0]^T$. As an orthogonal matrix with a value of ± 1 in its (1, 1) element, it follows that the remaining elements in the first row of $Q_1^T U Q_1$ must all be zero. Consequently, U can be written in the following block diagonal representation:

$$U = Q_1 \begin{bmatrix} \pm 1 & \\ & U_2 \end{bmatrix} Q_1^T. \quad (32)$$

The situation is somewhat more involved for complex eigenvalues. Since U is a real matrix, its complex eigenvalues appear as complex conjugate pairs, and the corresponding eigenvectors can also be chosen to be complex conjugates of each other. Let $\lambda = e^{j\theta} = \cos\theta + j\sin\theta$ be a non-real eigenvalue, where $\sin\theta$ is necessarily nonzero. Next, note that by Lemmas 1 and 2 the corresponding eigenvector $\mathbf{w} = \mathbf{u} + j\mathbf{v}$ can be chosen so that its real and imaginary parts have unit norm and are perpendicular to each other. It can then be shown that

$$[\mathbf{u} \quad -\mathbf{v}]^T U [\mathbf{u} \quad -\mathbf{v}] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (33)$$

Letting Q_1 be an orthogonal matrix whose first two columns are $\mathbf{u}$ and $-\mathbf{v}$, yields

$$U = Q_1 \begin{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} & \\ & U_2 \end{bmatrix} Q_1^T. \quad (34)$$

It is then possible to iteratively apply (32) and (34) to obtain the form given in (20). For example, the orthogonal matrix U_2 in (20) has the same eigenvalues as U except that the first eigenvalue of U is removed. So, if there is another real eigenvalue, then one can choose an $(n-1) \times (n-1)$ orthogonal matrix Q_2 such that $Q_2^T U_2 Q_2$ also has the same form as (32). Hence, for the orthogonal matrix $Q = Q_1 \text{diag}(1, Q_2)$,

$$U = Q \begin{bmatrix} \pm 1 & & \\ & \pm 1 & \\ & & U_3 \end{bmatrix} Q^T. \quad (35)$$

Thus, iteratively applying the above procedures results in the desired form.

References

Bartels, R. H., J. C. Beatty, and B. A. Barsky (1998). *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling* (1st ed.)., Chapter Hermite and Cubic Spline Interpolation., pp. 9–17. The Morgan Kaufmann Series in Computer Graphics. San Francisco, CA: Morgan Kaufmann.

Collins, Jr., E. G. and E. J. Coyle (2008). Vibration-based terrain classification using surface profile input frequency responses. In *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, California.

Coyle, E. J., E. G. Collins, Jr., and R. G. Roberts (2011). Speed independent terrain classification using singular value decomposition interpolation. In *Proceedings of IEEE International Conference on Robotics and Automation*, Shanghai, China. submitted for publication.

Culver, W. J. (1966). On the existence and uniqueness of the real logarithm of a matrix. *Proceedings of American Mathimatical Society 17*, 1146–1151.

Davies, P. I. and N. J. Higham (2003). A Schur-Parlett algorithm for computing matrix functions. *SIAM Journal of Matrix Analysis Applications 25*, 464–485.

Duda, R. O., P. E. Hart, and D. G. Stork (2001). *Pattern Classification* (2 ed.). New York: Wiley.

DuPont, E. M., E. G. Collins, Jr., E. J. Coyle, and R. G. Roberts (2008). Terrain classification using vibration sensors: theory and methods. In *New Research on Mobile Robotics*. NOVA.

DuPont, E. M., C. A. Moore, E. G. Collins, Jr., and E. J. Coyle (2008). Frequency response method for online terrain identification in unmanned ground vehicles. *Autonomous Robots 24*(4), 337–347.

DuPont, E. M., C. A. Moore, and R. G. Roberts (2008). Terrain classification for mobile robots traveling at various speeds an eigenspace manifold approach. In *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, California.

Feiveson, A. H. (1966). The generation of a random sample-covariance matrix. Technical Report NASA-TN-D-3207, NASA.

Hildebrand, F. B. (1987). *Introduction to Numerical Analysis: 2nd Edition*. New York, NY, USA: Dover Publications, Inc.

Park, F. C. and B. Ravani (1997). Smooth invariant interpolation of rotations. *ACM Transactions on Graphics 16*(3), 277–295.

Rasmussen, C. E. and C. K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Shoemake, K. (1985). Animating rotation with quaternion curves. *SIGGRAPH Computer Graphics 19*(3), 245–254.

Ward, C. C. and K. Iagnemma (2008). Speed-independent vibration-based terrain classification for passenger vehicles. *Vehicle System Dynamics 00*(0), 1–19.

Weiss, C., H. Fröhlich, and A. Zell (2006). Vibration-based terrain classification using support vector machines. In *Proceedings of* *the International Conference on Intelligent Robots and Systems*, Beijing, China.

Yuan, Q., A. Thangali, V. Ablavsky, and S. Sclaroff (2007). Parameter sensitive detectors. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1–6.