

CLARKSON UNIVERSITY

**Manifold Learning and Dimensionality Reduction  
in Collective Motion**

A Dissertation

By

**Kelum D. Gajamannage**

Department of Mathematics

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy, Mathematics

**April 5, 2016**

Accepted by the Graduate School

\_\_\_\_\_,

Date,

\_\_\_\_\_

Dean of the Graduate School

The undersigned have examined the thesis/dissertation titled “**Manifold Learning and Dimensionality Reduction in Collective Motion**” presented by **KELUM D. GAJAMANNAGE**, a candidate for the degree of **Doctor of Philosophy (Mathematics)**, and hereby certify that it is worthy for acceptance.

---

Date

---

Erik Bollt (Advisor)

---

Date

---

Joseph Skufca

---

Date

---

Daniel ben-Avraham

---

Date

---

Jie Sun

---

Date

---

Rana Parshad

# Abstract

We study mathematical aspects of the behaviors of animal groups, both to empirically reveal underlying simplicity, and to model possible mechanisms of them. Social groups consisting of multi-agents perform spatially coherent behaviors to self-defense against predation or to attain dietary benefits. As positions of agents in a group at a time can be represented as a point in a higher-dimensional space, all such points describing the behavior create a hidden manifold in the higher-dimensional space. Dimensional Reduction (DR) methods revealing the underlying manifold sometimes misinterpret the topology of the embedding. In fact, multi-agent systems may undergo abrupt changes defined as phase transitions, due to natural perturbations. A phase transition splits the underlying manifold into two sub-manifolds with distinct dimensionalities and often immobilizes DR methods. However, behavioral coherence is costly, as it requires agents to change their activities such as eating and resting in a communal time rather than their ideal time. Segregation of a group into subgroups may reduce the cost if subgroups are homogeneous, but it may also create small subgroups that are then vulnerable to predation.

Here we first construct an alternative DR approach that demands a two-dimensional embedding which topologically summarizes the higher-dimensional data. Then, we devise a method to detect phase transitions in a multi-agent system. We also adopt a distinct metric that is capable of computing changes in a multi-agent group, and characterizing alternation of manifolds. Finally, we model an efficient segregation of a group by optimizing a cost function.

# Acknowledgments

The person who has always inspired me to be a good man, enlightened my life in education by cherishing all my needs, and converted his all of his energy and effort towards my success, is my father. Foremost, I would express my sincere gratitude for him. I would convey my heartiest gratitude for my mother for her endless faith and devotion.

All the stress had come along with late night and weekend working, traveling far-away for conferences, I would thanks a lot my wife Dilhani and our daughter Shenya. Their support, encouragement, quiet patience, and unwavering love were undeniably the bedrock upon which the past few years of my life have been built. Tolerance of my occasional vulgar moods is a testament of their unyielding devotion and love.

Most notably, I would be thankful for my Ph.D. advisor Prof. Erik Bollt for his patience, motivation, enthusiasm, and immense knowledge. His guidance has helped me in all the time of doing research and writing. I could not have imagined of having a perfect advisor for my Ph.D. study. I would also thankful for my research collaborators, Prof. Maurizio Porfiri, Prof. Mason Porter, Prof. Marian Dawkins, and Prof. Sachit Butail, for the given contribution towards the success of my research career.

Then I would thank for Professors, Joseph Skufca, Daniel ben-Avraham, Jie Sun and Rana Parshad for being in my thesis committee and providing me valuable comments and suggestions. I also acknowledge all the faculty in the Department of Mathematics at Clarkson University for their association which has helped me to grasp the knowledge in subjects as well as research.

I appreciate the financial support received by the National Science Foundation. I would also thank to the Department of Mathematics for providing me a teaching assistantship till I complete my Ph.D.

I appreciate of being in the *Chaos Lab* with my friends Ranil, Tian, Lingchen, Chris, Sula, Amila, Abdul, Ryan, Matt, and Sean.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xxii</b>
<b>List of Symbols</b>	<b>xxiii</b>
<b>List of Abbreviations</b>	<b>xxvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Dimensionality reduction . . . . .	4
1.2 Phase transitions in multi-agent systems . . . . .	11
1.3 Alternation of collective motion . . . . .	14
1.4 Efficient segregation of herds . . . . .	16
<b>2 Dimensionality Reduction by Principal Manifolds</b>	<b>21</b>

2.0.1	Review of Isomap . . . . .	25
2.1	Constructing the algorithm . . . . .	27
2.1.1	Clustering the raw data . . . . .	28
2.1.2	Smoothing the manifold . . . . .	32
2.1.3	Embedding . . . . .	35
2.1.4	Mapping between the manifold and raw data . . . . .	38
2.2	Examples . . . . .	40
2.2.1	Paraboloid . . . . .	40
2.2.2	A noisy swiss-roll . . . . .	42
2.2.3	Collective motion: simulation of predator mobbing . . . . .	44
2.3	Performance analysis . . . . .	49
2.3.1	Error analysis . . . . .	49
2.3.2	Computational complexity . . . . .	54
<b>3</b>	<b>Detecting Phase Transitions using Manifold's Curvature</b>	<b>57</b>
3.1	Method of detecting a phase transition . . . . .	60
3.1.1	Approximating the curvature of a curve . . . . .	61
3.1.2	Approximating the curvature of a manifold . . . . .	64
3.1.3	Transition via local data distribution on the manifold . . . . .	69
3.2	Algorithm for detecting phase transitions . . . . .	72
3.3	Group behavioral examples . . . . .	74
3.3.1	A simulation of the Vicsek model . . . . .	74
3.3.2	A crowd of human . . . . .	78

3.3.3	A flock of birds . . . . .	79
3.3.4	A school of fish . . . . .	79
<b>4</b>	<b>Characterizing Alternating Manifolds</b>	<b>83</b>
4.1	Construction of the metric . . . . .	84
4.1.1	Mapping of agents and computation of speed . . . . .	85
4.1.2	Computation of coordination . . . . .	89
4.1.3	Computation of structure . . . . .	90
4.1.4	The metric . . . . .	91
4.2	Examples simulated from the Vicsek model . . . . .	92
4.2.1	Switching speed . . . . .	94
4.2.2	Switching coordination . . . . .	96
4.2.3	Switching number of clusters . . . . .	97
<b>5</b>	<b>An Efficient Segregation of a Herd of Cows</b>	<b>102</b>
5.1	Temporal evolution and change of states of cows . . . . .	103
5.2	Cost module determining the segregation of herds . . . . .	106
5.2.1	Synchronization Module . . . . .	109
5.2.2	Risk Module . . . . .	110
5.2.3	Cost function . . . . .	112
5.2.4	Optimization . . . . .	112
5.3	Cost module and temporal evolution . . . . .	113
5.4	Example . . . . .	113
5.4.1	General segregation . . . . .	113

5.4.2	Inter-sexual segregation . . . . .	117
<b>6</b>	<b>Concluding Remarks and Future Work</b>	<b>121</b>
6.1	Dimensionality reduction by principal manifolds . . . . .	122
6.1.1	Future work . . . . .	125
6.2	Detecting phase transitions . . . . .	126
6.2.1	Future work . . . . .	127
6.3	Characterizing alternating manifolds . . . . .	128
6.3.1	Future work . . . . .	130
6.4	An efficient segregation of a herd of cows . . . . .	130
6.4.1	Future work . . . . .	133



# List of Figures

1.1	Rich collective behavior in (a) human, (b) fish, (c) birds, (e) molds, (f) ants, and (g) cows. . . . .	2
1.2	A flock of birds with three agents. While the length of an arrow relates to the speed of the corresponding agent, orientation indicates the direction of the motion. At the beginning, (a) all the agents evolve in the same speed to the same direction, and then they (b) evolve with different speeds and directions. . . . .	3
1.3	One dimensional (green) embedding of a particle moving in three dimensional spiral (orange). While red points represent the start and end of the trajectory, blue represent intermediate positions. Here, $t_1$ , $t_2$ , $t_3$ , and $t_4$ are four arbitrary time steps. . . . .	6

- 1.4 Using Isomap to create a two-dimensional embedding of a simulation of collective behavior. (a) Predator mobbing of twenty agents moving on a translating circular trajectory on a plane (enclosing a predator moving at constant speed at a  $45^\circ$  angle), axes  $y_{1,i}$ ,  $y_{2,i}$  generally represent coordinates for the  $i$ -th agent. (b) Scaled residual variance of candidate low-dimensional embeddings produced by Isomap using different nearest neighbor values  $\alpha$  (green-circle, brown-square, and black-triangle), and (c) two-dimensional representation of the data for six nearest neighbors (black-triangle). Green and blue crosses mark the start and end points of the trajectories, respectively. . . . . 10
- 1.5 An abrupt change in a crowd behavior where a (a) walking crowd suddenly starts (b) running [101]. (c) Scaled residual variance with respect to the dimensionality, where the dimensionality of the underlying manifold is given by an elbow, is obtained by running Isomap upon frames in each phase with six nearest neighbors. Embedding dimensionalities of sub-manifolds representing walking (blue circle) and running (red square) of the crowd are three and four, respectively. . . . . 13

1.6	A simulation of self-propelled particles in two dimensions. Changes in coordination can be seen as switching of embedding manifolds of different dimensions. The top row shows snapshots of self-propelled particles as they move with low, high, and then low noise again. The bottom row displays the normalized Isomap residual variance for each instance (a, b, c) and the whole motion (d). The location of elbow in each plot indicates that low noise instances are low-dimensional (3), high noise instance results into a high-dimensional embedding (6). The embedding dimensionality for the whole motion is four. . . . .	15
1.7	Cycle of states eating ( $\mathcal{E}$ ), lying down ( $\mathcal{R}$ ), and standing ( $\mathcal{S}$ ) of cows. .	20
2.1	The map $\Phi$ embeds three dimensional (a) swiss-roll data onto a two dimensions rectangle. As the map $\Phi$ is homeomorphic, the raw data can be retrieved if the map $\Phi^{-1}$ is applied to the embedding data. Points are colored to highlight the relative configuration. . . . .	25
2.2	Isomap residual variance plot of the dataset presented in Figure 2.1(a). The dimensionality of the underlying manifold is two as indicated by the elbow. . . . .	27

2.3	Clustering three dimensional swiss-roll data into slices. (a) Assigning points, $\mathbf{y}^{(t)}$ 's, between hyperplanes thought the points $\mathbf{a}_{j-1}$ and $\mathbf{a}_j$ into the cluster $C_j^1$ by using the angles $\mathbf{y}^{(t)}\mathbf{a}_j\mathbf{q}_1$ and $\mathbf{y}^{(t)}\mathbf{a}_{j-1}\mathbf{q}_1$ as explained in Inequalities (2.9). (b) Using a single reference point (black square) located far from the data points, the dataset can be sliced into non-overlapping sections which are perpendicular to the line joining the centroid (gray asterisk) and the reference point, then slices are used to draw a grid pattern using splines. (c) A hole in the data will result in multiple sub-clusters (shown in different colors red and blue) within a cluster. .	30
2.4	Approximating clusters with smoothing splines. (a) A cubic smoothing spline is fit onto two-dimensional data, $\mathcal{D} = \{\mathbf{y}^{(t)}   t = 1, \dots, T\}$ for some $T$ , where $\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}]$ and $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}]^T$ , using different values of the smoothing parameter $p$ . (b) Smoothing splines fit clusters of a swiss-roll dataset from a single reference point. . . . .	34
2.5	Embedding the data into two-dimensional coordinates. (a) The intersection point (red) between two splines lies midway on the shortest distance between them. (b) Intersection points (red) for a swiss-roll, and (c) the two-dimensional coordinate of a point (triangle) is the geodesic distance along the intersecting splines (grey) to the axis splines (blue) from an origin (circle). . . . .	35

2.6	Embedding a higher-dimensional point $\mathbf{z}$ onto the manifold using distances along the axis splines from the origin $\mathbf{O}$ to the closest intersection $\tilde{\mathbf{z}}$ and the local distance from $\tilde{\mathbf{z}}$ to $\mathbf{z}$ . Local distances are computed by projecting the distance $\tilde{\mathbf{z}} - \mathbf{z}$ onto the unit tangent vectors $\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1$ and $\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2$ .	38
2.7	The mapping of the point $\mathbf{x}$ on the underlying manifold to the original high-dimensional space is done by finding the projections $\gamma_1, \gamma_2$ locally and propagating the same into the high dimensions. Local approximation can be created using adjacent splines $\{k, k + 1\}$ , and $\{l, l + 1\}$ and their intersections $\{\mathbf{i}^{k,l}, \mathbf{i}^{k,l+1}, \mathbf{i}^{k+1,l}\}$ .	39
2.8	Embedding of raw data from (a) a three-dimensional paraboloid with 2000 points is attained after (b) smoothing the data by using cubic smoothing splines (black). The red points herein show the virtual intersections of splines. (c) Two dimensional embedding manifold of the data along with the intrinsic coordinates. Points are colored to highlight the relative configuration.	42
2.9	Two-dimensional embeddings of the paraboloid by MDS (a), Isomap (b), Diffusion maps (c), KPCA (d), LSE (e), LLE (f), HLLE (g), and LE (h). Points are colored to highlight the relative configuration with the raw data in Figure 2.8(a).	43

2.10	Two-dimensional embedding of the noisy swiss-roll with 2500 points. (a) Raw data of the noisy swiss-roll is clustered and (b) smoothened using splines (black) in order to obtain the principal manifold. The red points herein show the virtual intersections of splines. (c) Two dimensional embedding with intrinsic coordinates denoting the distance from the origin of the manifold coordinate. Points are colored to highlight the relative configuration. . . . .	45
2.11	Two-dimensional embeddings of the noisy swiss-roll by MDS (a), Isomap (b), Diffusion maps (c), KPCA (d), LSE (e), LLE (f), HLLE (g), and LE (h). Points are colored to highlight the relative configuration with the raw data in Figure 2.10(a). . . . .	45
2.12	Embedding a forty dimensional dataset of an idealized simulation of predator mobbing where twenty particles move around a translating circle. A comparison of individual axes of the embedding produced by our method and those produced using Isomap shows that our method performs distinctly better in terms of preserving the time signature as well as range of the axes. The nearest neighbor parameter used for Isomap is equal to 10. . . . .	48
2.13	Variation of $\Gamma$ with respect to $p$ and the corresponding linear fit for (a) a noise free swiss-roll, and (b) a noise free paraboloid. $\Gamma$ is computed using Equation (2.24) between raw and embedded data. . . . .	51

2.14	Variation of $\Gamma$ with respect to noise ( $\epsilon$ ) and the corresponding quadratic fit $\Gamma$ is computed between adjacency matrices for the raw data and embedded data by Equation (2.24). . . . .	53
2.15	Variation of $\Gamma$ with respect to number of points ( $T$ ) and the corresponding exponential fit with $p = 0.9$ and $\epsilon = 0.2$ . $\Gamma$ is computed between adjacency matrices for the raw data and embedded data by Equation (2.24). . . . .	53
3.1	A cartoon of two manifolds (blue and red) and the locus of singularity (orange) of the human crowd example in Section 1.2. Snapshots sampled at time $t_1$ and $t_2$ representing the first and second phases of the motion embed as points $x^{(t_1)}$ and $x^{(t_2)}$ in the corresponding manifold, respectively.	58
3.2	(a) A sombrero-hat of 2000 points having brim (green), crown (blue), and locus (red). (b) The vertical section at $x_2 = 0$ of the manifold indicates a high curvature at red points. (c) Isomap residual plots showing embedding dimensionalities by elbows reveal that the dimensionalities of two sub-manifolds (blue and green) are two while the dimensionality of the locus (red) is three. . . . .	60
3.3	(a) Superimposing a neighborhood of the curve $C$ at point $\mathbf{y}$ with an arc $\mathbf{y}_1\mathbf{y}\mathbf{y}_2$ which subtend an small angle of $2\Theta$ at the origin $\mathbf{O}$ of a translating circle having a radius $r_c$ . (b) Circular sector $\mathbf{y}_1\mathbf{y}\mathbf{y}_2$ which is rotated such that the blue arrow represents the horizontal axis of the coordinate system. . . . .	61

3.4	Computation of principal sections $\Pi_{\mathbf{x}}^{(1)}$ and $\Pi_{\mathbf{x}}^{(2)}$ , and principal directions $\mathbf{v}_{\mathbf{x}}^{(1)}$ and $\mathbf{v}_{\mathbf{x}}^{(2)}$ , of a two-dimensional manifold $\mathcal{M}^2$ at the point $\mathbf{x}$ . The principal sections are mutually orthogonal as they are constructed from a mutual orthogonal tangential basis. . . . .	65
3.5	Two dimensional saddle surface $\mathcal{M}^2$ , described by the Equation (3.22) for $x_1, x_2 \in \mathbb{U}[-2, 2]$ . . . . .	67
3.6	Local distribution of data in the neighborhood $\mathcal{N}_{\mathbf{x}}$ on a two dimensional manifold $\mathcal{M}^2$ . Principal sections $\Pi_{\mathbf{x}}^{(1)}$ and $\Pi_{\mathbf{x}}^{(2)}$ are created by using the shape operator around $\mathbf{x}$ , and curves $C_{\mathbf{x}}^{(1)}$ and $C_{\mathbf{x}}^{(2)}$ are produced by intersecting $\Pi_{\mathbf{x}}^{(1)}$ and $\Pi_{\mathbf{x}}^{(2)}$ with $\mathcal{M}^2$ . . . . .	70
3.7	A particle swarm simulated from the Vicsek model that evolves at four consecutive time steps indicated by $t - 1, t, t + 1$ , and $t + 2$ . Gray color ellipses show neighbors. As the red particle joins with the blue particles at the $t$ -th time step and becomes neighbors, the red particle's orientation at the current time step is determined as the average direction of the blue particles in the previous time step. This interaction of the orientations is shown by colored arrows. . . . .	75



3.8	Detecting transitions in a particle swarm simulated using the Vicsek model with alternating noise levels. (a) Distribution of $(\sigma_4/\sigma_1)_t$ versus frame numbers. Therein, the range of frames representing each sub-manifold is marked by a left-right arrow and the frame containing each transition is marked by a red circle along with the frame number associated. (b) Descending order of 20 largest transitions including frame numbers of three largest transitions. (c) Isomap residual variance versus dimensionality of each sub-manifold. Therein, while the numbers next to the plot gives the underlying dimensionalities, colors refer to the corresponding sub-manifolds in (a) . . . . .	77
3.9	Detecting a transition between phases of walking and running in a crowd of human. (a) Distribution of $(\sigma_3/\sigma_1)_t$ versus frame numbers. Therein, left-right arrows and the red circle represent ranges of frames in each sub-manifold and the frame at the transition, respectively, while the snapshots show instances in each phase. (b) Plot of descending order of 20 largest transitions representing the largest transition in red along with the frame number. . . . .	78
3.10	Detecting a transition in a bird flock between phases sitting and flying. (a) Distribution of $(\sigma_3/\sigma_1)_t$ shows ranges of frames representing two sub-manifolds by left-right arrows and instances of phases by snapshots. (b) Descending order of 20 largest transitions. The frame at the largest transition is represented by red in Figures (a) and (b). . . . .	80

3.11	Detecting transitions in a school of fish. (a) Distribution of $(\sigma_6/\sigma_1)_n$ with left-right arrows showing ranges of frames in sub-manifolds, and red dots showing frames at transitions. (b) Snapshots of the school before (left) and after (right) each transition. (c) Descending order of 20 largest transitions consisting four largest transitions marked in red with their frame numbers. . . . .	80
4.1	Mapping of positions of agents between two consecutive time steps. Same colored square and circle pair refers to the same agent. Squares represent agents' positions at the $t$ -th time step and circles represent agents' positions at the $(t + 1)$ -th time step. . . . .	86
4.2	(Same as Figure 3.7) A particle swarm simulated from the Vicsek model that evolves at four consecutive time steps indicated by $t - 1$ , $t$ , $t + 1$ , and $t + 2$ . Gray color ellipses show neighbors. Since the red particle joins with the blue particles at the $t$ -th time step and becomes neighbors, the red particle's orientation at the current time step is determined as the average direction of the blue particles in the previous time step. This similarity of the orientations is shown by yellow arrows. . . . .	93

- 4.3 (a) Distribution of  $\mathcal{X}^{(t)}$  versus time. Snapshots in each section show that the structures and velocities of the agents. Therein, the length of the lines measure the speeds and the directions quantify the orientations of motion. Normalized Isomap residual variance versus dimensionality plots for each section are presented in the top row. For any two given configurations  $\mathbf{y}^{(t_1)}$  and  $\mathbf{y}^{(t_2)}$  where  $t_1, t_2 \in 1, \dots, T$ , and any arbitrary  $t_0 \in 1, \dots, T$ , if the distance metric image (b) shows the same color at two points  $(t_0, t_1)$  and  $(t_0, t_2)$  on the image, then those configurations lie on the same manifold, or on different manifolds otherwise. . . . . 95
- 4.4 (a) Distribution of  $\mathcal{X}^{(t)}$  versus time with snapshots showing the structures along with velocities of the agents. Therein, the lengths measure their speeds and arrows quantify their orientations. Normalized Isomap residual variance plots for each section (red, green, yellow) are shown in the top row. For any two given configurations  $\mathbf{y}^{(t_1)}$  and  $\mathbf{y}^{(t_2)}$  where  $t_1, t_2 \in 1, \dots, T$ , and any arbitrary  $t_0 \in 1, \dots, T$ , if the distance metric image (b) gives the same color for pixels at two points  $(t_0, t_1)$  and  $(t_0, t_2)$  on the image, then those configurations are in the same manifold, or in different manifolds otherwise. . . . . 97
- 4.5 Trajectories of agents,  $\mathbf{y}_i^{(t)} = [y_{1,i}^{(t)}, y_{2,i}^{(t)}]^T$ , in two connected components (sub-groups), with red and blue curves depicting the trajectories of their centroids. . . . . 98

4.6	(a) Distribution of $\mathcal{X}^{(t)}$ versus time with snapshots giving the structures and velocities. Therein, the length of lines measure their speeds and arrows quantify the their orientations. The top row of plots shows normalized Isomap residual variance versus dimensionality for each section (red, green, yellow). For any two given configurations $\mathbf{y}^{(t_1)}$ and $\mathbf{y}^{(t_2)}$ where $t_1, t_2 \in 1, \dots, T$ , and any arbitrary $t_0 \in 1, \dots, T$ , if the distance metric image (b) gives the same color for pixels at two points $(t_0, t_1)$ and $(t_0, t_2)$ on the image, then those configurations are in the same manifold, or in different manifolds otherwise. . . . .	100
5.1	Switching condition of an arbitrary cow [128]. Edges of the square in the left panel represent boundaries given by Equation (5.6). Right panel shows three states $\{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$ where $\mathcal{S} = \mathcal{S}_x \cup \mathcal{S}_y$ , and the potential switching between states by arrows. Same color of arrows between panels refers to the same dynamics. . . . .	106
5.2	Switching of states [128], $\{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$ where $\mathcal{S} = \mathcal{S}_x \cup \mathcal{S}_y$ , corresponding to Table 5.1. Here, an example, “1a” refers to the “case a” of the “case 1” in the table. . . . .	108
5.3	(a) Predation risk as an inverse exponential function of group size. The red curve segment shows significant risk and blue shows safe. (b) Risk of segregating a herd of 20 cows ( $n$ ) into two groups of sizes $n_1^{(t)}$ and $n_2^{(t)} = n - n_1^{(t)}$ , with $\tau = 1/30$ and $n_s = 10$ (yellow triangles), 20 (green squares), and 30 (brown circles). . . . .	111

5.4	Process diagram of the algorithm. Algorithm inputs initial conditions for variables hunger, fatigue, risk, and grouping adjacency matrix. Different parameters input by ES, SM, RM, and CF are explained in Sections 5.1, 5.2.1, 5.2.2, and 5.2.4, respectively. Algorithm validates the grouping adjacency matrix in each time steps with the new grouping information obtained by optimizing the CF. Algorithm outputs groups providing the efficient segregation at each time step and stops at the stopping criteria which is the maximum number of time steps. . . . .	114
5.5	Total cost (red squares) at $t = 20$ and its components: hunger (blue triangles), fatigue (green diamonds), and risk (yellow circles) versus the index representing the combinations of group sizes as given in Table 5.2. Total cost and its components for different convex parameters: (a) $\gamma_1 = .33$ and $\gamma_2 = .33$ ; (b) $\gamma_1 = .5$ and $\gamma_2 = .25$ ; and (c) $\gamma_1 = .25$ and $\gamma_2 = .5$ . . . . .	116
5.6	Grouping dynamics and related cost over time. (a) Occupancy, $N_{1,2,3}^{(20)}$ , of the first six cows in three groups where each one represented in a different color. (b) The number of groups into which the herd is split is determined by (c) the total cost. . . . .	117
5.7	Dynamics of a herd of 10 cows in two groups and related cost over time. (a) First group containing five cows are colored in red, while the second group containing the other five cows colored in yellow. (b) Number of mismatched cows between groups in the herd. (c) The cost that determine the cows' occupancy in each group. . . . .	120

# List of Tables

2.1	Default parameters implemented in Matlab Toolbox for DR [134] are used in the paraboloid and noisy swiss-roll examples. . . . .	40
5.1	Iterative scheme for the temporal evolution given by Equations (5.1), (5.2), (5.3), and (5.5). The Poincaré section describing the switching between cases is presented in Figure 5.2. . . . .	107
5.2	All plausible group sizes, if a herd of 12 cows is segregated into a maximum of three groups. . . . .	115

# List of Symbols

Symbol	Description
$\mathcal{M}^m$	A manifold with $m$ dimensions
$\mathcal{L}$	Locus of singularities
$\Phi$	Map between manifolds
$\Phi_f, \Phi_g, \Phi_h$	Maps between particles/agents
$\mathcal{D}$	Input data
$\mathcal{D}_m$	Matrix form of input data
$d$	Dimension of the input data
$n$	Total number of agents
$N$	Set of all agents
$\mathcal{Y}^{(t)}$	A point on the manifold sampled at the $t$ -th time step
$\mathbf{y}_i^{(t)}$	Position vector of the $i$ -th particle/agent at the $t$ -th time step
$\mu$	Mean (centroid) of the input dataset
$e$	Dimension of the output embedding
$\mathbf{x}$	A point in the embedding space
$\mathbf{z}$	An external input point for the embedding

Symbols	Description
$\mathbf{O}$	Origin of the embedding manifold
$O$	Order of the complexity
$t$	Index for time
$T$	Total number of time steps/points in the input dataset
$\mathcal{T}_t$	Transition at the $t$ -th time step
$(\mathbf{v}_1, \sigma_1), (\mathbf{v}_2, \sigma_2)$	First two Principal Components (PCs) of the input dataset, where $\mathbf{v}_i$ is the $d$ -dimensional eigenvector and $\sigma_i$ is the eigenvalue of the $i$ -th PC for $i = 1, 2$
$\mathbf{q}_{1,2}$	Reference points 1 and 2, respectively
$\mathcal{C}_j^{1,2}$	The $j$ -th cluster for the reference points 1 and 2, respectively
$n_{\mathcal{C}}^{1,2}$	Number of clusters for the reference points 1 and 2, respectively
$\mathcal{S}_j^{1,2}$	The $j$ -th cubic spline for the reference points 1 and 2, respectively
$\mathbf{i}^{k,l}$	The intersection point in $d$ -dimensional space between splines $k$ and $l$
$(\tilde{\mathbf{s}}_{\tilde{\mathbf{z}}}^1, \tilde{\mathbf{s}}_{\tilde{\mathbf{z}}}^2)$	Tangents to splines at the intersection point $\tilde{\mathbf{z}}$
$\mathcal{G}_j^{1,2}$	The $j$ -th geodesic for reference points 1 and 2, respectively
$m_s$	Mesh size
$p$	Smoothing parameter in cubic splines
$\alpha$	Number of neighbors in the nearest neighbor search
$r_c$	Radius of a circle
$r$	Pearson correlation coefficient
$\kappa$	Curvature of a curve/manifold
$\epsilon$	Noise parameter



Symbol	Description
$\rho$	Density of data
$\omega$	Number of revolutions
$\Delta$	Metric distance
$\Sigma_C$	Covariance matrix
$C$	A curve
$A$	Distance matrix of a neighborhood graph
$A^{(t)}$	Adjacency matrix corresponds to neighbors at the $t$ -th time step
$\Gamma$	Embedding error
$f, g$	Functions
$\gamma_1, \gamma_2$	Scalars
$\theta, \phi, \Theta$	Angles
$\lambda_1, \lambda_2, \dots$	Eigenvalues
$i, j, k, l, m$	Indices
$\sigma$	Standard deviation
$\mathcal{N}_{\mathbf{x}}$	A neighborhood at the point $\mathbf{x}$
$\mathcal{S}_{\mathbf{x}}$	Shape operator at the point $\mathbf{x}$
$N_{\mathbf{x}}$	Unit normal at point $\mathbf{x}$ on the manifold
$\mathbf{u}_{\mathbf{x}}^{(j)}, \hat{\mathbf{u}}_{\mathbf{x}}^{(j)}$	The $j$ -th tangent and unit tangent on the manifold at the point $\mathbf{x}$
$\Pi_{\mathbf{x}}^{(j)}$	The $j$ -th principal section at the point $\mathbf{x}$
$\kappa(\mathbf{v}_{\mathbf{x}}^{(j)})$ and $\mathbf{v}_{\mathbf{x}}^{(j)}$	The $j$ -th principal curvature and the principal direction at the point $\mathbf{x}$ on the manifold
$C_{\mathbf{x}}^{(j)}$	Curve created by intersecting the $j$ -th principal section at point $\mathbf{x}$ with the manifold

Symbol	Description
$\sum_t^\alpha$	The $\alpha$ -points moving sum at the $t$ -th time step
$\mathbf{w}_i^{(t)}$	Velocity of the $i$ -th particle/agent in the $t$ -th time step
$\mathbf{w}_p^{(t)}$	Velocity of the predator at the $t$ -th time step
$w_i^{(t)}$	Speed of the $i$ -th particle/agent at the $t$ -th time step
$d_k(\mathbf{a}_1, \mathbf{a}_2)$	Geodesic distance between points $\mathbf{a}_1$ and $\mathbf{a}_2$ along the spline $k$
$d(i, j)$	Euclidean distance between nodes $i$ and $j$
$d_{\mathcal{D}}(t_1, t_2)$	Euclidean distance between points $t_1$ and $t_2$ in the data matrix $\mathcal{D}$
$\mathcal{P}^{(t)}$	Polarization of agents at the $t$ -th time step
$\gamma^{(t)}$	Rotational angle of a particle/an agent group at the $t$ -th time step
$R_i^{(t)}$	Rotation matrix of the $i$ -th agent at the $t$ -th time step

# List of Abbreviations

Abbreviation	Description
DR	Dimensionality Reduction
NDR	Nonlinear Dimensionality Reduction
PM	Principal Manifold
PCA	Principal Component Analysis
MDS	Multidimensional Scaling
KPCA	Kernel Principal Component Analysis
LLE	Local Linear Embedding
HLLE	Hessian Local Linear Embedding
LE	Laplacian Eigenmaps
CCA	Curvilinear Component Analysis
MS	Manifold Sculpting
LSE	Local Spline Embedding
ES	Evolution Scheme
SM	Synchronization Module
RM	Risk Module
CF	Cost Function
CM	Cost Module

# Chapter 1

## Introduction

Multi-agent systems such as crowds of people [66, 94, 144], schools of fish [58, 103], flocks of birds [11, 95], colonies of molds [107] and ants [108], and herds of cows [128] often synchronize their activities and exhibit discrete patterns as shown in Figure 1.1. Animals manifest synchronization as it offers a high vigilance from potential predator attacks and increases the ability to attain and defend food sources [46, 73].

The study of multi-agent systems has become a very popular topic in the area of nonlinear dynamical systems during the past few decades as they are rich with abstract mathematical concepts. In nature, most of such systems yield a high-dimensionality as they consist of plenty of agents. For instance, consider a flock of birds with three agents such that each agent moves in a three dimensional space as in Figure 1.2. This system attains the best coordination when the agents have similar speeds and directions and requires only three variables to describe the motion. Thus, the three dimensional coordinates of the centroid of the system may be used as variables to describe the whole motion. In contrast, the flock of birds reaches the worst coordination when each

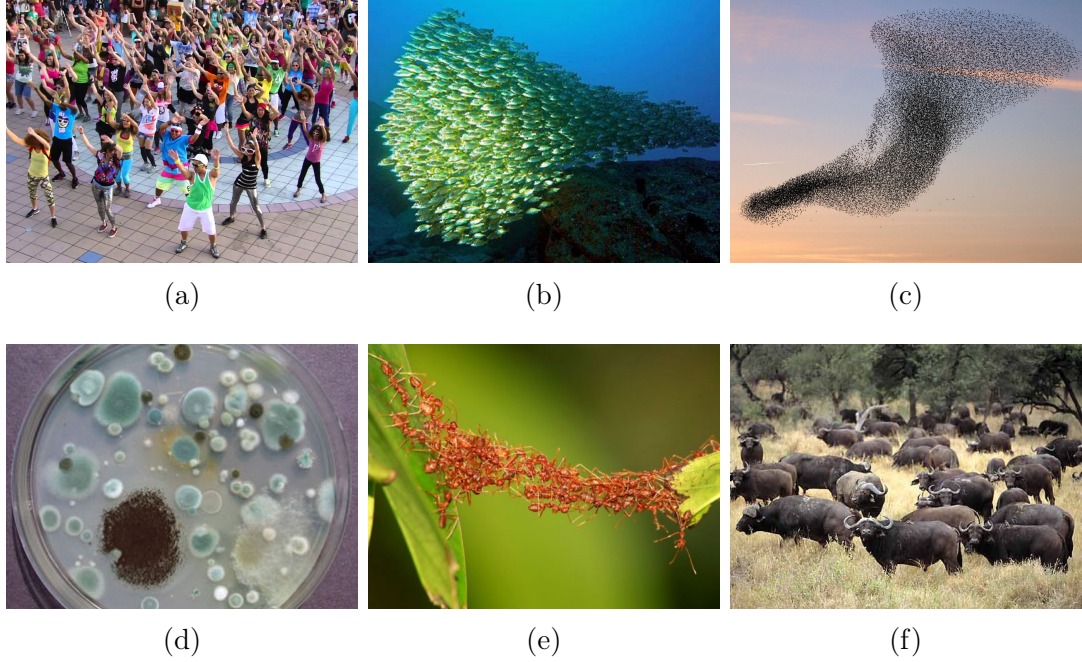


Figure 1.1: Rich collective behavior in (a) human, (b) fish, (c) birds, (e) molds, (f) ants, and (g) cows.

agent has different speeds and directions, thereby the same system is described by nine dimensions (each bird is in three dimensions for three birds).

As the number of agents increases, the dimensionality of the system increases even faster. The study of such systems requires to manipulate higher-dimensional data which is arduous even for the latest technology. Thus, revealing the best (coarse) observables those describe the dynamics of a complex system and reducing the dimensionality is vital. Even in the same system, describing the motion at a transition made by change of an activity requires complex mathematical frameworks. Sometimes, collective behavior undergoes alternation between two or more activities due to natural perturbations and makes the study even more arduous. Communal decisions in multi-agent groups are made either despotically by a dominant (or dominants) or democratically by the

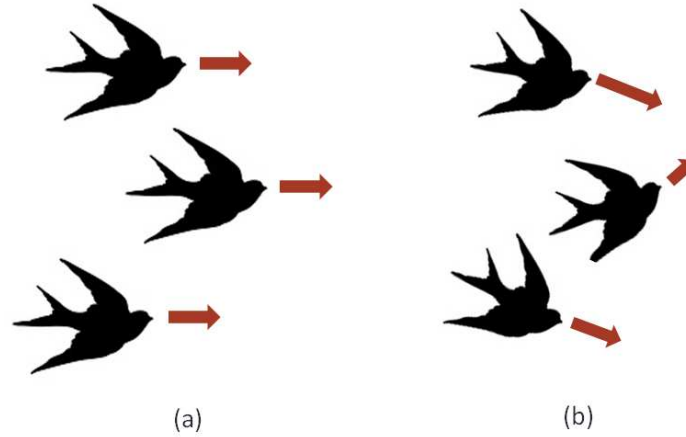


Figure 1.2: A flock of birds with three agents. While the length of an arrow relates to the speed of the corresponding agent, orientation indicates the direction of the motion. At the beginning, (a) all the agents evolve in the same speed to the same direction, and then they (b) evolve with different speeds and directions.

majority of individuals in the group [29]. Even though the agents benefit from the synchronization made by communal decisions, it requires a significant portion of agents to change their desired activity at a worst time which would incur a lack of efficiency. This low efficiency can be reestablished by segregating the multi-agent group into subgroups such that the agents' activities in each group are homogeneous.

My research presented in this thesis mainly focuses on four components:

1. Development of a novel Nonlinear Dimensionality Reduction (NDR) method using cubic smoothing splines.
2. Detection of phase transitions in collective motion using the manifold's curvature.
3. Characterization of alternating manifolds in collective motion using a metric which assesses changes of speed, coordination, and structure of agents.
4. Model an efficient segregation of a herd of cows by minimizing the cost incurred by hunger, fatigue, and predation risk.

We present introductory details including the literature review and the problem definition for each of the aforesaid component in the rest of this chapter. However, we make an individual chapter for each of those to present detailed methods and explanatory examples.

## 1.1 Dimensionality reduction

Dynamics of a multi-agent system can be studied either by tracking agents' trajectories or reducing the dimensionality of the system. As tracking of all agents' trajectories and deriving conclusions about the whole system is exhaustive due to the high-dimensionality, filming the whole system and studying the collective behavior to comprehend social aspects in a biological sense is desired. However, analyzing these datasets is still a nontrivial task due to the high volume. A desirable step that may ease the experimenter's task of locating events of interest is to identify coarse observables [50, 72, 89] and behavioral measures [88] as the group navigates through the space. In this context, Dimensionality Reduction (DR) offers a large set of tools to infer properties of such complex multi-agent dynamical systems. Thus, we use methods from machine learning to reduce the dimensionality and to identify low-dimensional embedding manifold for higher-dimensional data.

We can reveal a hidden manifold in an abstract higher-dimensional space which describes the collective motion of a group at time  $t$  such that each instant  $\mathbf{y}^{(t)}$ , of the data  $\mathcal{D}$ , describing the multi-agent system corresponds to a point  $\mathbf{x}^{(t)}$  on the manifold

$\mathcal{M}$  [1, 53]. Thus, the multi-agent system evolves according to the underlying flow

$$\Phi : \mathcal{D} \rightarrow \mathcal{M} \text{ such that } \Phi(\mathbf{y}^{(t)}) = \mathbf{x}^{(t)}. \quad (1.1)$$

The data

$$\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, T\}, \quad (1.2)$$

for some  $T$ , describing the collective motion can either be configurations or frames. For a system of  $n$  agents, a configuration is a vector of positions of agents  $\mathbf{y}^{(t)}$ , such that

$$\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}; \mathbf{y}_2^{(t)}; \dots; \mathbf{y}_n^{(t)}], \quad (1.3)$$

where  $\mathbf{y}_i^{(t)}$  is a column vector of the position of the  $i$ -th agent.

If the data is frames partitioned from a video of collective motion having the pixel dimensionality of  $m_1 \times m_2$ , then for all  $t$ , we reshape each intensity matrix of a frame into a column matrix as

$$\begin{bmatrix} y_1^{(t)} & y_{m_1+1}^{(t)} & \cdots & y_{(m_2-1)m_1+1}^{(t)} \\ y_2^{(t)} & y_{m_1+2}^{(t)} & \cdots & y_{(m_2-1)m_1+2}^{(t)} \\ \cdots & \cdots & \cdots & \cdots \\ y_{m_1}^{(t)} & y_{2m_1}^{(t)} & \cdots & y_{m_1 m_2}^{(t)} \end{bmatrix}_{m_1 \times m_2} \implies [y_1^{(t)}, y_2^{(t)}, \dots, y_{m_1+1}^{(t)}, \dots, y_{m_1 m_2}^{(t)}]^T.$$

Thus, an entry of input data  $\mathcal{D}$ , in this case is

$$\mathbf{y}^{(t)} = [y_1^{(t)}, \dots, y_i^{(t)}, \dots, y_{m_1 m_2}^{(t)}]^T, \quad (1.4)$$



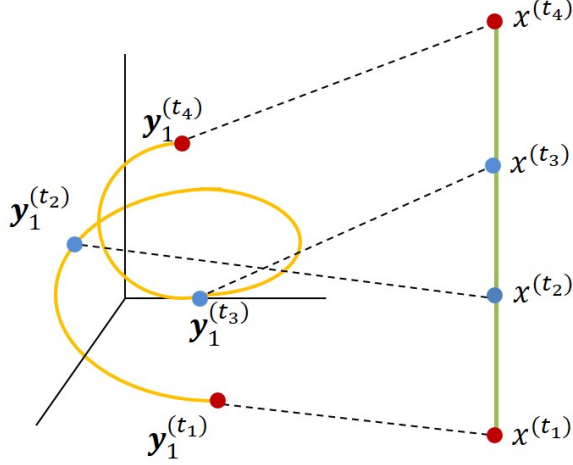


Figure 1.3: One dimensional (green) embedding of a particle moving in three dimensional spiral (orange). While red points represent the start and end of the trajectory, blue represent intermediate positions. Here,  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  are four arbitrary time steps.

where  $y_i^{(t)}$  is the intensity of the  $i$ -th pixel of the reshaped frame. As an example, consider a particle (a single-agent system) revolves in a three dimensional spiral as shown in Figure 1.3. As  $n = 1$  and the motion is three dimensional, Equation (1.3) reduces to  $\mathcal{Y}^{(t)} = [\mathbf{y}_1^{(t)}]$  where  $\mathbf{y}_1^{(t)} \in \mathbb{R}^3$  for all  $t$ . This spiral is embedded in one dimension, thus  $x^{(t)} \in \mathbb{R}$  for all  $t$ .

Traditional DR methods based on linear techniques such as Principal Components Analysis (PCA) have been shown to possess limited accuracy when input data is nonlinear and complex [135]. Dimensionality reduction entails finding the axes of maximum variability [71] or retaining the distances between points [32]. Multi Dimensional Scaling (MDS) with Euclidean metric is another DR method which attains low-dimensional representation by retaining the pairwise distance of points in low-dimensional representations [32]. However, Euclidean distance calculates the shortest distance between two points on a manifold instead of the genuine manifold distance, which may lead to

a difficulty of inferring low-dimensional embeddings. The isometric mapping algorithm (Isomap) resolves the problem associated with MDS by preserving the pairwise geodesic distance between points [130]; it has recently been used to analyze group properties in collective motion such as the level of coordination and fragmentation [1, 7, 34]. Within Isomap, however, short-circuiting [117] created by faulty connections in the neighborhood graph, manifold non-convexity [35, 131] and holes in the data [82] can degrade the faithfulness of the reconstructed embedding manifold.

Diffusion maps [26] have also been shown to successfully identify coarse observables in collective phenomena [8] that would otherwise require hit-and-trial guesswork [50]. Beyond Isomap and diffusion maps, the potential of other NDR methods to study collective motion is largely untested. For example, Kernel PCA (KPCA) requires the computation of the eigenvectors of the kernel matrix instead of the eigenvectors of the covariance matrix of the data [122], but this is computationally expensive [135]. Local Linear Embedding (LLE) embeds high-dimensional data through global minimization of local linear reconstruction errors [135]. Hessian LLE (HLLE) minimizes the curviness of the higher-dimensional manifold by assuming that the low-dimensional embedding is locally isometric [44]. Laplacian Eigenmaps (LE) perform a weighted minimization (instead of global minimization as in LLE) of distance between each point and its given nearest neighbors to embed high-dimensional data [13].

Iterative NDR approaches have also been recently developed in order to bypass spectral decomposition which is common in most of NDR methods [56]. Curvilinear Component Analysis (CCA) employs a self-organized neural network to perform two tasks, namely, vector quantization of submanifolds in the input space and nonlinear

projection of quantized vectors onto a low-dimensional space [37]. This method minimizes the distance between the input and output spaces. Manifold Sculpting (MS) transforms data by balancing two opposing heuristics: first, scaling information out of unwanted dimensions, and second, preserving local structure in the data. The MS method which is robust to sampling issues, iteratively reduces the dimensionality by using a cost function that simulates the relationship among points in a local neighborhoods [56]. The Local Spline Embedding (LSE) is another NDR method which embeds the data points using splines that map each local coordinate into a global coordinate of the underlying manifold by minimizing the reconstruction error of the objective function [141]. This method reduces the dimensionality by solving an eigenvalue problem while the local geometry is exploited by the tangential projection of data. Local spline embedding assumes that the data is not only unaffected by noise or outliers, but also sampled from a smooth manifold which ensures the existence of a smooth low-dimensional embedding.

Due to the global perspective of all these methods, none of them provides sufficient control over the mapping from the original high-dimensional dataset to the low-dimensional representation, limiting the analysis in the embedding space. In other words, the low-dimensional coordinates are not immediately perceived as useful, whereby one must correlate the axes of the embedding manifold with selected functions of known observables to deduce their physical meaning [50, 130]. In this context, a desirable feature of DR that we emphasize here is the regularity in the spatial structure and range of points on the embedding space, despite the presence of noise.

With regard to datasets of collective behavior, nonlinear methods have limited use

for detailed analysis at the level of the embedding space. This is primarily because the majority of these methods collapse the data onto a higher-dimensional space, whose coordinates are not guaranteed to be linear functions of known system variables [79]. In an idealized simulation of predator induced mobbing [43], a form of collective motion where a group of animals forages around a moving predator, two degrees of freedom are obvious, namely the translation of the group and the rotation about the predator (center of the translating circle). This two-dimensional manifold is not immediately perceived by Isomap, even for the idealized scenario presented in Figure 1.4, where a group of 20 individuals ( $n$ ) simulated in 2000 time steps ( $T$ ), input as data

$$\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, 2000\}, \quad (1.5)$$

of configurations

$$\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}; \mathbf{y}_2^{(t)}; \dots; \mathbf{y}_{20}^{(t)}], \quad (1.6)$$

where  $\mathbf{y}_i^{(t)} = [y_{1,i}^{(t)}, y_{2,i}^{(t)}]^T \in \mathbb{R}^2$  for all  $i$ , rotate about a predator moving at a constant speed about a line bisecting the first quadrant (Figure 1.4(a)).

Isomap running on the data is unable to locate a distinct elbow in the residual variance versus dimensionality curve, notwithstanding substantial tweaking of the parameters. Thus, the inferred dimensionality is always 1 (Figure 1.4(b)). For a two-dimensional embedding (Figure 1.4(c)), visual comparison of the relative scale of the axes indicates that the horizontal axis represents a greater translation than the vertical axis. It is likely that the horizontal axis captures the motion of the group along the translating circle. The vertical axis could instead be associated with either the

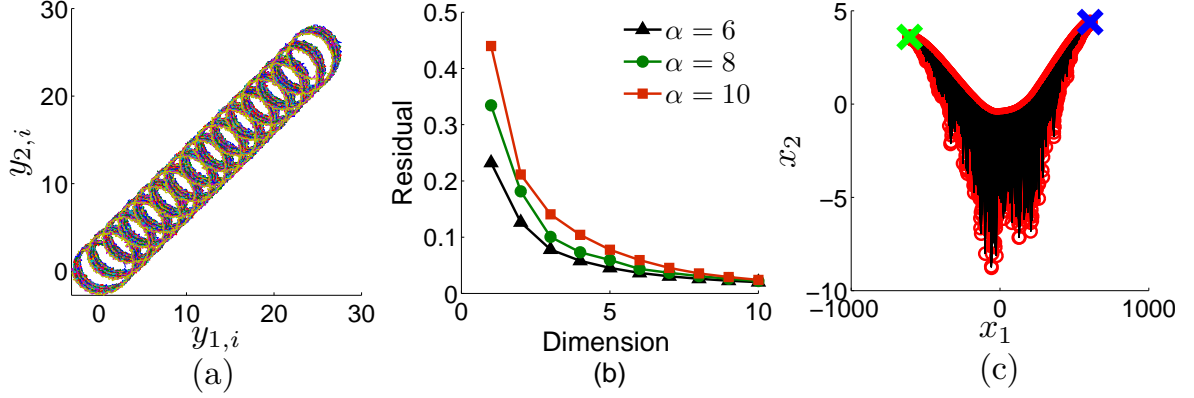


Figure 1.4: Using Isomap to create a two-dimensional embedding of a simulation of collective behavior. (a) Predator mobbing of twenty agents moving on a translating circular trajectory on a plane (enclosing a predator moving at constant speed at a  $45^\circ$  angle), axes  $y_{1,i}$ ,  $y_{2,i}$  generally represent coordinates for the  $i$ -th agent. (b) Scaled residual variance of candidate low-dimensional embeddings produced by Isomap using different nearest neighbor values  $\alpha$  (green-circle, brown-square, and black-triangle), and (c) two-dimensional representation of the data for six nearest neighbors (black-triangle). Green and blue crosses mark the start and end points of the trajectories, respectively.

motion about the center of the circle or noise, which is independent and identically distributed at each time step. The ambiguity in determining the meaning of such direction indicates a drawback of Isomap in providing meaningful interpretations of the low-dimensional coordinates.

We propose an alternative approach to NDR in Chapter 2 that directly work on the input data in the higher-dimensional space [10, 62], and doesn't require heavy matrix computations or orthogonalization like most of other DR methods. The new method enhances data regularity as it computes embedding coordinates using the geodesic distance like Isomap rather than using Euclidean distance. As this method constructs the manifold using cubic smoothing splines, the smoothnesses of the manifold is strongly preserved. The method undergoes three steps: clustering the raw data, smoothing the

manifold, and embedding of higher-dimensional data. The clustering step partitions the data into two sets of non-overlapping segments called slices as in [10], using two reference points. Smoothing step first creates a neighborhood graph within each cluster using the range search in [51], with a specific distance parameter. Then, it locates the longest geodesic using the Dijkstra’s algorithm [42, 81] in each neighborhood graph.

In [62], one-dimensional curves are used to represent the high-dimensional data in which points on the curve coincide with the average of raw data projected on itself. Similarly, a cubic smoothing spline is used to fit to the longest geodesic. The second set of slices and corresponding splines are created in a similar way resulting a two-dimensional grid structure known as the PM surface. The grid structure made by two sets of cubic smoothing splines serves as the intrinsic coordinate system on the manifold. In the embedding step, the virtual intersection points between pairs of cubic smoothing splines, one from each reference point, are constructed. Any point in the input space is projected on the PM in terms of the intersection points and their respective distances along the cubic splines. We demonstrate stepwise computations of the method using a three dimensional dataset called a swiss-roll. Then we validate the method on three different datasets, including an instance of collective motion similar to that in Figure 1.4.

## 1.2 Phase transitions in multi-agent systems

In multi-agent animal groups, the response to a perturbation — internal or external — is often manifested in the form of changes in group speed, coordination, or structure

[12, 23, 31, 64, 90, 96, 137]. Such changes are witnessed in schools of fish and flocks of birds under attack [83, 102, 133], foraging animal groups [30, 38], and crowds of human exposed to alarm situations leading to panic [65, 123]. Based on recent efforts demonstrating that the collective motion is associated with a low-dimensional embedding [1, 21, 34, 36, 53], we expect that such behavioral changes should be manifested in variation of the topology of an underlying manifold.

A phase transition, defined as an abrupt change of the behavior in a multi-agent system, shifts the system from one state to another [39, 125] which is topologically represented as an existence of two distinct manifolds separated at a locus of singularity where the transition physically exists. A manifold containing singularity may immobilizes DR methods and reveals unfaithful embeddings. Thus, detection of such transitions, especially in humans [4], is an ongoing research problem [6].

As an instance to demonstrate the aforesaid pitfall, we partition 120 frames ( $T$ ) from a video of a human crowd containing a transition at the 61-th time step from walk (Figure 1.5(a)) to run (Figure 1.5(b)) presented in Figure 1.5. Here, the data

$$\mathcal{D} = \{\mathcal{Y}^{(t)} | t = 1, \dots, 120\}, \quad (1.7)$$

consists of frames in the format

$$\mathcal{Y}^{(t)} = \left[ y_1^{(t)}, y_2^{(t)}, \dots, y_{m_1 m_2}^{(t)} \right]^T, \quad (1.8)$$

where  $m_1 \times m_2$  is the pixel dimension of a frame and  $y_i^{(t)}$  is the intensity of the  $i$ -th

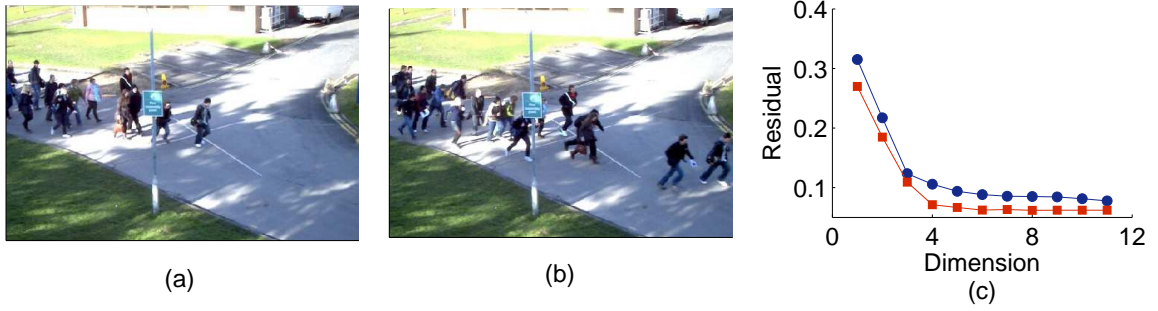


Figure 1.5: An abrupt change in a crowd behavior where a (a) walking crowd suddenly starts (b) running [101]. (c) Scaled residual variance with respect to the dimensionality, where the dimensionality of the underlying manifold is given by an elbow, is obtained by running Isomap upon frames in each phase with six nearest neighbors. Embedding dimensionalities of sub-manifolds representing walking (blue circle) and running (red square) of the crowd are three and four, respectively.

pixel of the reshaped frame at the  $t$ -th time step as explained in Equation (1.4). As we know the frame number at the phase transitions, we partition the frames into two phases and run Isomap, available in [130], at each phase separately. We observe that each phase is embedded in a different space (Figure 1.5(c)), since their dimensionalities are different.

The mapping in Equation (1.1) switches from one manifold to another at a phase transition. Thus, the system can be represented as two distinct sub-manifolds,  $\mathcal{M}^{(j)}$  where  $j = 1$  and 2, along with a locus singularities ( $\mathcal{L}$ ) where the sub-manifolds intersect as

$$\mathcal{L} = \bigcap \mathcal{M}^{(j)} \quad (1.9)$$

[54]. As a phase transition splits a manifold into two sub-manifolds, prior to utilize DR algorithms, the transitions in the collective behavior should be detected.

We detect transitions based on revealing high curvature on the manifold which dif-



ferentiate phases of the motion in Chapter 3. We hypothesize that a phase transition is manifested in the form of change in local curvature which can be detected by ratios of singular values. We formalize this concept in two dimensions by proving a relation between curvature and singular value ratio, and then extend it to higher dimensions. We justify that the same transition can also be detected by analyzing the ratio of the smallest singular value to the largest singular value computed locally upon a neighborhood of points on the manifold. Based on the distribution of a moving sum of absolute moving difference of the singular value ratios, we detect transitions and order their magnitudes.

### 1.3 Alternation of collective motion

Another salient scenario is that the collective motion sometimes alternates between behavioral patterns through transitions. Thus, the dimensionality of the manifold underlying the motion is also alternates. As an illustrative example of alternation between high and low coordination in a multi-agent group, we simulate 50 self-propelled particles ( $n$ ) in 180 time steps ( $T$ ) using the Vicsek model [136] by alternatively imposing low and high noise to their individual dynamics in three distinct phases. Figure 1.6 shows the resulting variation in the coordination of agents. Following [1, 21, 34, 36], an established DR algorithm called Isomap [130] is applied to data

$$\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, 180\}, \quad (1.10)$$

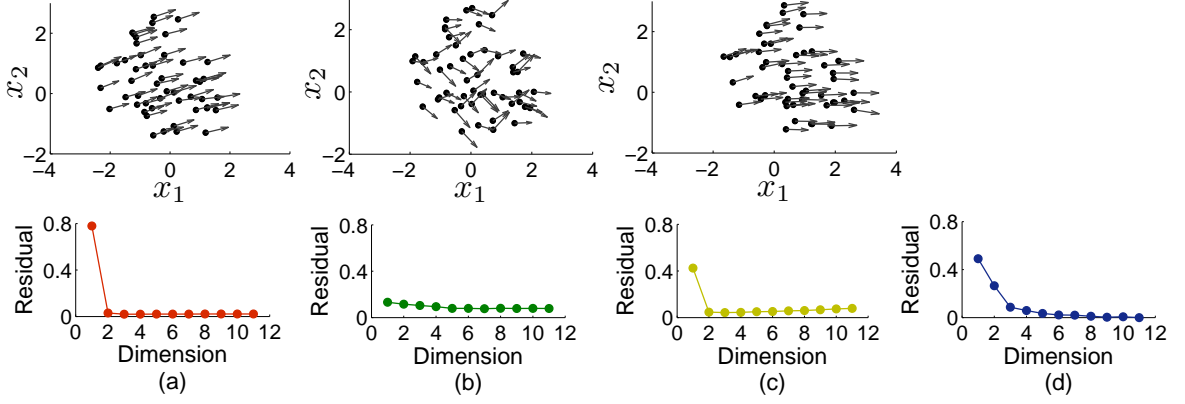


Figure 1.6: A simulation of self-propelled particles in two dimensions. Changes in coordination can be seen as switching of embedding manifolds of different dimensions. The top row shows snapshots of self-propelled particles as they move with low, high, and then low noise again. The bottom row displays the normalized Isomap residual variance for each instance (a, b, c) and the whole motion (d). The location of elbow in each plot indicates that low noise instances are low-dimensional (3), high noise instance results into a high-dimensional embedding (6). The embedding dimensionality for the whole motion is four.

of configurations

$$\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}; \mathbf{y}_2^{(t)}; \dots; \mathbf{y}_{50}^{(t)}], \quad (1.11)$$

where  $\mathbf{y}_i^{(t)} \in \mathbb{R}^2$  for all  $i$  (Equation (1.3)), obtained from each phase as well as the whole dataset comprising all three phases to infer the dimensionality of the underlying embedding. The residual of the reconstruction error shows that the first and last instances (low noise) embed on manifolds with lower-dimensionality than the middle instance (high noise) [1]. However, when the same algorithm is applied to the full dataset, we find an embedding dimensionality that is neither an indicator of high nor low group coordination. This example suggests that group behavior is manifested in distinct embedding manifolds, and a naive implementation of DR could result in a loss of information about critical changes in collective motion [1].

We also propose a method to study higher-dimensional alternating manifolds in response to an internal or external perturbation to a group of agents in Chapter 4. Our approach shares similarities with the coarse-grained analysis of stochasticity-induced switching in one-dimensional model of collective motion [72]. Therein, a single coarse observable, the average nearest-neighbor distance is used to distinguish between the stationary and mobile states in the collective motion. We characterize group motion by using a single coarse observable which computes the distance between sub-manifolds corresponding to distinct group activities. Specifically, we use agent positions in each configuration to create a single weighted metric of group speed, orientation, and structure. The positions of agents between consecutive time steps are mapped using a nearest-neighbor search in position and velocity which suffices to characterize the group motion.

## 1.4 Efficient segregation of herds

Animals in a multi-agent group gain many advantages from grouping and synchronizing together including greater vigilance from more pairs of eyes, coordinated defense against predators, and increased ability to find and defend food sources [46, 73]. However, living in large groups also carries disadvantages, such as increased risk of disease and parasitism [5, 20], having food stolen [40], and interference with movement [132]. Specially, perfect synchronization requires animals to change their activities in a communal time rather than their ideal time, which would then encounters low efficiency. Even when the segregation of the group into homogeneous subgroups enhances the

efficiency, it may also create small subgroups which are then vulnerable to predation [33, 63]. The balance between synchrony and risk of predation varies with individuals' occupancy in groups and can be described by a Cost Function (CF), formulated as a convex combination of cost components contributed through synchrony and risk.

As groups of animals become very large, the cost gain through synchrony tend to exceeds that of risk as a significant number of individuals change their desirable activities like eating or resting to agree with the majority. At some stage, the group can become sufficiently large that it can segregate into two groups and give more benefit to its members than the overloaded single group [124]. Thus, there is a point at which the CF should be optimized to determine the best segregation of the group. However, an optimum group size is not necessarily stable [124]. Even when a group is already at its optimum, extra individuals will still benefit from joining the group. A stable group size is therefore likely to be consistently larger than an optimum group size [124, 127].

Sometimes, the context even be more complex as individuals within a group differ in many ways that relate to their fitness. For example, if males and females differ in their nutritional needs and yet all benefit from staying with the group, some individuals may have to interrupt valuable feeding time to keep up with the herd when it moves off [28, 114]. Individuals may stay within a group, however they pay a cost by synchronizing activities with others and being forced to switch between eating, resting, or moving at a communal time rather than the time that is optimal for them as individuals [45]. Alternatively, the group may segregate into sub-groups consisting individuals with more similar switching times such as all-males and all-females or juveniles and adults where the cost of synchronization is consequently lower [27, 45, 87, 106, 116]. The cost

of synchronization will depend both on the range of different types of animals in the group and on the particular activities involved, so that some animals such as baboons will break up into sub-groups for foraging, particularly when food is scarce, and come together into larger groups for sleeping [120].

Degree of social activity segregation between two classes has been assessed using a model and the performance was tested using data on inter-sexual social segregation in red deer [28]. However, even in the same class (e.g. male), activity synchronization can vary significantly, and it can depend on the age, body mass, or health of animals. Thus, class-based segregation can create weak groups such that agents are homogeneous across many classes. Moreover, segregation between classes is seasonal sometimes, as an example, inter-sexual social segregation is not observed during the mating season in nature.

The ideal time that the switching of an activity occur had been modeled quasirormally in [45], and the grouping benefit was revealed as linearly increasing with the group size. This work hypothesizes that segregation never occurs due to synchronization cost and the different levels of synchronization always beneficial. It is also expected that a herd attempts to assure the synchrony by either dispersing or staying together. We assume that large groups encounter a higher synchronization cost while small groups increase predation risk [33, 63]. Therefore, the perfect group size is neither too large nor too small.

Communal decisions in herds are made either despotically by a dominant (or dominants) or democratically by the majority of individuals in the herd [29]. The synchronization cost has also been modeled to compare despotic and democratic groups in

[29], and revealed that the cost for despotic groups are higher than that of democratic groups.

Specifically, cows (*Bos taurus*) make regular decisions over staying with their herd or split from their herd due to their two-stage feeding process that involves first grazing (standing up) and then ruminating (largely lying down). Lying down and ruminating can occupy up to 65% of a cow’s day [70, 74]. Both grazing and lying (including ruminating) are essential to successful digestion of grass [118], but they have to be terminated if the herd decides to move to another area, which can happen about 15–20 times a day [70]. Figure 1.7 shows the cycle of states as we observe often in free range cows. Individual cows have similar, but not identical, needs for lying and grazing [74, 110], so that keeping up with the herd each time it moves can incur a considerable cost in interrupted grazing or lying times. This cost can include reduced growth rate in young cattle [91, 98], and physiological and behavioral symptoms of “stress” deprived of adequate opportunities for lying down [49, 93].

We model an efficient segregation of a herd of cows by minimizing a CF quantifying their hunger, fatigue, and predation risk in Chapter 5. We always rely on a perfect synchronization, thus we model segregation of a herd into groups, fission-fusion of groups and switch of agents between groups such that groups are highly homogeneous with respect to their hunger and fatigue. Moreover, we rely on democracy in our study when they take communal decisions as it ensures high efficiency. We define cows’ grazing demand as *hunger*, while the demand for lying as *fatigue*. We compute their hunger and fatigue at each time step using the Evolution Scheme (ES) in [128] and quantify synchronization cost based on cows’ hunger and fatigue. We assess the

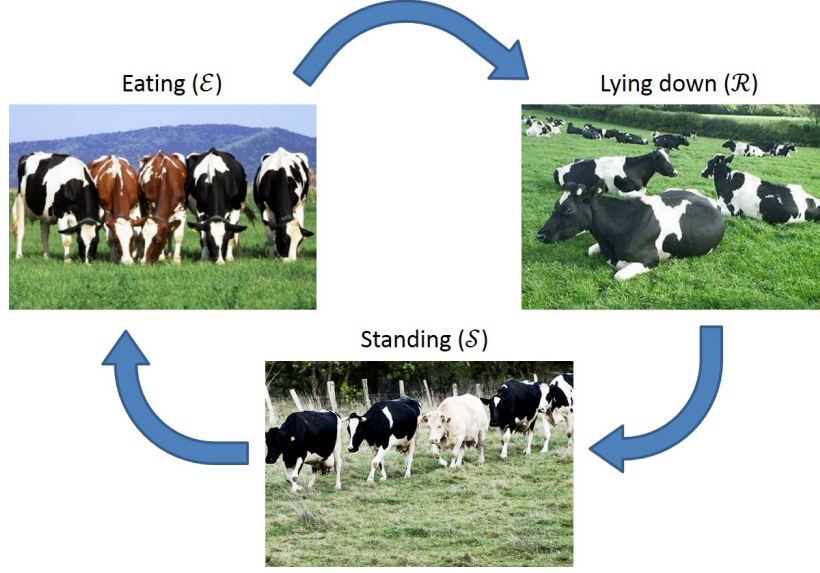


Figure 1.7: Cycle of states eating ( $\mathcal{E}$ ), lying down ( $\mathcal{R}$ ), and standing ( $\mathcal{S}$ ) of cows.

cost incurred by the hunger (or fatigue) as the mean of standard deviations of hunger (or fatigue) computed in each group and the cost from risk is modeled as inverse exponentially of the group size. The CF is defined as a convex combination of costs associate with hunger, fatigue, and risk. During each time step, we minimize this CF over all plausible groups created from a given herd with known maximum allowance of groups and determine the efficient segregation.

## Chapter 2

# Dimensionality Reduction by Principal Manifolds

*Dimensionality Reduction* (DR) is defined as a transformation of higher-dimensional data into a meaningful representation of lower dimension without losing any information [135]. Dimensionality reduction evaluates the *coarse observables* and then reveals the underlying *manifold*  $\mathcal{M}$ , hidden in the higher-dimensional data  $\mathcal{D}$ , along with the underlying map

$$\Phi : \mathcal{D} \rightarrow \mathcal{M} \text{ such that } \Phi(\mathbf{y}^{(t)}) = \mathbf{x}^{(t)}. \quad (2.1)$$

This map is also known as a *topological embedding* of the data  $\mathcal{D}$  onto the manifold  $\mathcal{M}$ . As  $\Phi$  is a topological embedding, it also provides the inverse mapping  $\Phi^{-1}$  which maps points on the manifold with the points in the higher-dimensional space. Let assume a  $d$ -dimensional input dataset is embedded onto an  $e$ -dimensional space such that  $e \ll d$ . Thus, the map from a point  $\mathbf{x}^{(t)} \in \mathbb{R}^e$  in the embedding space to a point



$\mathcal{Y}^{(t)} \in \mathbb{R}^d$  in the original dataset is

$$\Phi^{-1} : \mathbb{R}^e \rightarrow \mathbb{R}^d \text{ such that } \Phi^{-1}(\mathbf{x}^{(t)}) = \mathcal{Y}^{(t)}. \quad (2.2)$$

**Definition 2.0.1. Dimensionality reduction [135].** A transformation of higher-dimensional data into a meaningful representation of lower dimension which preserves certain properties.

**Definition 2.0.2. Coarse observables [135].** The best observables (dimensions) those describe the dynamics of a complex system.

**Definition 2.0.3. Topological space [92].** A topology on a set  $X$  is a collection  $\mathcal{T}$  of subsets of  $X$  possessing the following properties:

- $\emptyset$  and  $X$  are in  $\mathcal{T}$ .
- The union of elements of any subcollection of  $\mathcal{T}$  is in  $\mathcal{T}$ .
- The intersection of the elements of any finite subcollection of  $\mathcal{T}$  is in  $\mathcal{T}$ .

Then,  $(X, \mathcal{T})$  is called a topological space.

**Definition 2.0.4. Homeomorphic [92].** A function  $h : (X, \mathcal{T}_X) \rightarrow (Y, \mathcal{T}_Y)$  between two topological spaces  $(X, \mathcal{T}_X)$  and  $(Y, \mathcal{T}_Y)$  is called a homeomorphism, if it possesses the following properties:

- $h$  is bijective.
- $h$  is continuous.

- The inverse function  $h^{-1}$  is continuous.

Two topological spaces  $(X, \mathcal{T}_X)$  and  $(Y, \mathcal{T}_Y)$  are homeomorphic, if there is a homeomorphism.

**Definition 2.0.5. Hausdorff space [92].** A topological space  $(X, \mathcal{T})$  is called a Hausdorff space, if for each pair  $x_1$  and  $x_2$  of distinct points of  $X$ , there exist disjoint neighborhoods  $\mathcal{N}_1$  and  $\mathcal{N}_2$  in  $\mathcal{T}$ , respectively.

**Definition 2.0.6. Countable basis [92].** A space  $X$  is said to have a countable basis at  $x$ , if there is a countable collection  $\mathcal{B}$  of neighborhoods of  $x$  such that each neighborhood of  $x$  contains at least one element of  $\mathcal{B}$ .

**Definition 2.0.7. An  $m$ -dimensional manifold [92].** An  $m$ -dimensional manifold is a Hausdorff space  $(X, \mathcal{T})$  with a countable basis such that each point  $x \in X$  has a neighborhood that is homeomorphic with an open subset of  $m$ -dimensional Euclidean space  $(\mathbb{R}^m)$ .

**Definition 2.0.8. Subspace topology [92].** Let  $(X, \mathcal{T})$  be a topological space and  $Y$  is a subset of  $X$ . The collection

$$\mathcal{T}_Y := \{Y \cap U | U \in \mathcal{T}\} \quad (2.3)$$

is a topology on  $Y$ , called the subspace topology.

**Definition 2.0.9. Topological Embedding [92].** Let  $f : (X, \mathcal{T}_X) \rightarrow (Y, \mathcal{T}_Y)$  is an injective continuous map, where  $(X, \mathcal{T}_X)$  and  $(Y, \mathcal{T}_Y)$  are topological spaces. If  $f' : (X, \mathcal{T}_X) \rightarrow (f(X), \mathcal{T}_{f(X)})$  be a homeomorphism and  $\mathcal{T}_{f(X)}$  is subspace topology

inherent from  $Y$ , then  $f : (X, \mathcal{T}_X) \rightarrow (Y, \mathcal{T}_Y)$  is a topological embedding (or simply, an embedding).

As given in Equations (1.2), (1.3), and (1.4), the data,

$$\mathcal{D} = \{\mathcal{Y}^{(t)} | t = 1, \dots, T\} \quad (2.4)$$

for some  $T$ , describing the *collective motion* can either be configurations or frames. For a multi-agent system of  $n$  agents, a configuration is a vector of positions of agents  $\mathcal{Y}^{(t)}$ , such that

$$\mathcal{Y}^{(t)} = [\mathbf{y}_1^{(t)}; \mathbf{y}_2^{(t)}; \dots; \mathbf{y}_n^{(t)}], \quad (2.5)$$

where  $\mathbf{y}_i^{(t)}$  is a column vector of the position of the  $i$ -th agent. If the data is frames partitioned from a video of collective motion having pixel dimensionality  $m_1 \times m_2$ , then for all  $t$ ,

$$\mathcal{Y}^{(t)} = [y_1^{(t)}, \dots, y_i^{(t)}, \dots, y_{m_1 m_2}^{(t)}]^T, \quad (2.6)$$

where  $y_i^{(t)}$  is the intensity of the  $i$ -th pixel of the column matrix formed by reshaping the  $t$ -th frame.

**Definition 2.0.10. *Collective motion* [137].** A spontaneous emergence of ordered movement in a system compound of a large number of self-propelled particles/agents.

Embedding of a three dimensional swiss-roll is presented in Figure 2.1 to demonstrate the concept. This example presents a single-agent system evolving over time. Thus, Equation (2.5) reduces to  $\mathcal{Y}^{(t)} = [\mathbf{y}_1^{(t)}]$ , where  $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}, y_{3,1}^{(t)}]^T \in \mathbb{R}^3$ , for all  $t$ . This swiss-roll embeds in two dimension, thus  $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}] \in \mathbb{R}^2$  for all  $t$ .

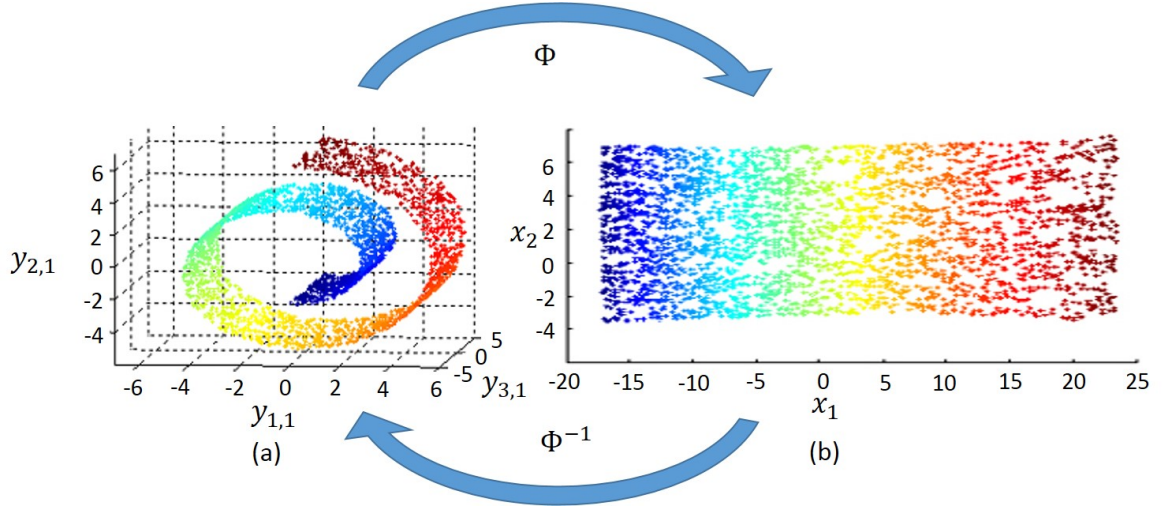


Figure 2.1: The map  $\Phi$  embeds three dimensional (a) swiss-roll data onto a two dimensions rectangle. As the map  $\Phi$  is homeomorphic, the raw data can be retrieved if the map  $\Phi^{-1}$  is applied to the embedding data. Points are colored to highlight the relative configuration.

In this thesis, we use a well established NDR method named as Isomap [130] to compare our method and sometimes to compute the dimensionality of underlying manifolds. Thus, we provide a detailed review about Isomap in Section 2.0.1.

### 2.0.1 Review of Isomap

Isomap is a widely using NDR method available in [130]. This is a simple and a robust method to reveal lower-dimensional embedding of a higher-dimensional data. It preserves the neighborhood distance and approximates intrinsic dimensionality of the embedding using geodesic distance induced by a neighborhood graph. Thus, Isomap is an extension of the classic Multi-Dimensional Scaling (MDS) method [75, 76], which used Euclidean metric to approximate neighborhood distance. Isomap defines geodesic distance between two nodes as the shortest path computed using the Dijkstra's algo-

rithm [42, 81]. The Isomap method is presented as Algorithm 1.

---

**Algorithm 1** *Isomap*: This algorithm inputs data  $\mathcal{D}$  and the parameter  $\alpha$  for number of nearest neighbors. Algorithm outputs  $d$ -dimensional embedding coordinates.

---

- 1: **procedure** ISOMAP( $\mathcal{D}, \alpha$ )
  - 2:     Search  $\alpha$  nearest neighbors for each point in the data  $\mathcal{D}$ .
  - 3:     Construct a neighborhood graph by adjoining neighbors together.
  - 4:     Compute lengths of edges using the Euclidean distance.
  - 5:     Compute the shortest path between each pair of nodes (points) using Dijkstra's algorithm [42, 81], and denote lengths of all paths by  $\mathcal{D}_G$ .
  - MDS**: *Computes lower-dimensional embedding.*
  - 6:     Apply the double centering to the matrix  $\mathcal{D}_G$ : For  $n$  total points, if the double centering matrix is  $\tau(\mathcal{D}_G)$ , then  $\tau(\mathcal{D}_G) = -H\mathcal{D}_S H/2$ , where  $\mathcal{D}_S$  is the matrix of squared distances,  $(\mathcal{D}_S)_{k_1, k_2} = ((\mathcal{D}_G)_{k_1, k_2})^2$  for  $k_1, k_2 = 1, \dots, n$ , and the centering matrix  $H = I - [1, \dots, 1][1, \dots, 1]^T/n$  [60].
  - 7:     Let  $\lambda_j$  be the  $j$ -th eigenvalue (in the descending order) of the matrix  $\tau(\mathcal{D}_G)$ , and let  $\mathbf{v}_j$  be the  $j$ -th eigenvector. Then, the  $d$ -dimensional embedding coordinates are first  $d$  entries of  $\sqrt{\lambda_j}\mathbf{v}_j$ .
  - 8: **end procedure**
- 

Isomap residual variance between data and embedding is used to analyze the accuracy of the embedding. A input dataset is embedded into all the dimensions 1–11 one at a time and then the residual variances are computed. The residual variance is computed using  $1 - R^2(\mathcal{D}_y, \mathcal{D}_x)$ , where  $\mathcal{D}_y$  is a matrix representing the distance between points in the neighborhood graph and  $\mathcal{D}_x$  is the matrix of Euclidean distances between points in the Isomap's low-dimensional embedding.

Figure 2.2 shows the residual variance versus the dimension of the data presented in Figure 2.1(a). Therein, one dimensional embedding encounters the largest error while the embedding dimensions 2–11 encounter similar residual variances. Even though all the embedding dimensions higher than two entails the same amount of residual, it is always important to consider the smallest dimensionality as the embedding dimensionality when the error is saturated. This is because, the embedding which is low in

dimension ease the subsequent analysis of the manifold. Thus, the best embedding dimension of this data set is two, as it is simple and addition of more dimensions doesn't improve the embedding.

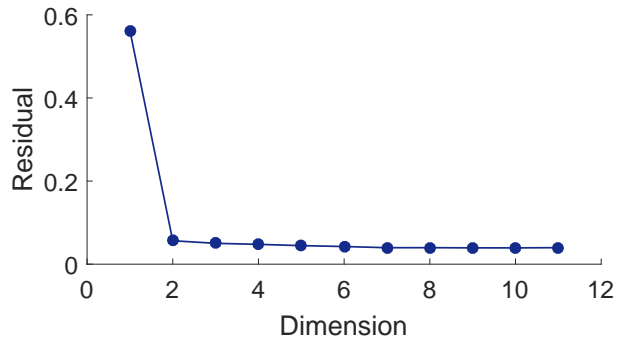


Figure 2.2: Isomap residual variance plot of the dataset presented in Figure 2.1(a). The dimensionality of the underlying manifold is two as indicated by the elbow.

Section 2.1 provides the detailed DR algorithm. Differently from other DR approaches the proposed method is based on a Principal Manifold (PM) of the raw dataset, obtained by constructing a series of cubic smoothing splines in slices of data that are partitioned using locally orthogonal hyperplanes. The resulting PM finding approach summarizes the data in the form of a two-dimensional embedding, that is  $e = 2$  in Map (2.2).

## 2.1 Constructing the algorithm

Here we propose an alternative approach to NDR, one that does not require heavy matrix computations or orthogonalization, involves working directly on raw data in the high-dimensional space [10, 62]. The new method relies on geodesic distance rather

than Euclidean distance and emphasizes manifold regularity. Our approach is based on a spline representation of data that allows us to control the expected manifold regularity. Typically, this entails conditioning the data so that the lower dimensions are revealed. For example, in [10], raw data is successively clustered through a series of lumping and splitting operations until a faithful classification of points is obtained.

In [62], one-dimensional parameterized curves are used to summarize the high-dimensional data by using a property called self-consistency, in which points on the final curve coincide with the average of raw data projected on itself. In a similar vein, we construct a PM of the high-dimensional data using cubic smoothing splines. Summarizing the data using splines to construct a PM of the data has two advantages: (i) it respects the data regularity by enabling access to every point of the dataset, and (ii) it gives direct control over the amount of noise rejection in extracting the true embedding through the smoothing parameter.

We demonstrate stepwise computations of the method using a three dimensional dataset called a swiss-roll as shown in Figure 2.1(a). As we mentioned at the beginning of this chapter, swiss-roll dataset represents positions of a temporally evolving single-agent system input as  $\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, T\}$  for some  $T$ , where  $\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}]$  and  $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}, y_{3,1}^{(t)}]^T$ .

### 2.1.1 Clustering the raw data

The clustering makes two sets of slices of the data such that slices between sets are mutually orthogonal. Input data  $\mathcal{D}$  is described by the Equation (1.2) such that each data entry  $\mathbf{y}^{(t)}$  is a  $d$ -dimensional point in the higher-dimensional space. Therein,  $T$

is the total number of time steps that the data is sampled from. As each time step generates a higher-dimensional data point, the underlying manifold consists of  $T$  total points.

The clustering begins by partitioning the data into non-overlapping clusters. The partitioning is performed by creating hyperplanes orthogonal to the straight line joining an external reference point  $\mathbf{q} \in \mathbb{R}^d$  through the mean  $\boldsymbol{\mu} \in \mathbb{R}^d$  to some point  $\tilde{\mathbf{q}} \in \mathbb{R}^d$ . Although any reference point may be selected, we use Principal Components (PCs) to make the clustering procedure efficient in data-representation. In particular, Principal Component Analysis (PCA) of the data matrix

$$\mathcal{D}_m = [\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(T)}] \quad (2.7)$$

is performed to obtain two largest principal components  $(\mathbf{v}_1, \sigma_1)$  and  $(\mathbf{v}_2, \sigma_2)$ , where  $\mathbf{v}_i$  is the  $d$ -dimensional coefficients and  $\sigma_i$  is the eigenvalue of the  $i$ -th PC. The first reference point is chosen such that  $\mathbf{q}_1 = \boldsymbol{\mu} + 3\sigma_1\mathbf{v}_1$ . To cover the full range of the dataset, the line joining the points  $\mathbf{q}_1$  and the point  $\tilde{\mathbf{q}}_1 = \boldsymbol{\mu} - 3\sigma_1\mathbf{v}_1$  is divided into  $n_c^1$  equal parts using the ratio formula in a  $d$ -dimensional space [104]. The  $j$ -th point  $\mathbf{a}_j \in \mathbb{R}^d$  on this line is

$$\mathbf{a}_j = \boldsymbol{\mu} + \frac{3\sigma_i(2j - n_c^i)\mathbf{v}_i}{n_c^i}; \quad j = 0, \dots, n_c^i \quad (2.8)$$

where  $i = 1$  for this case of dealing with the first reference point.

All points between hyperplanes through points  $\mathbf{a}_{j-1}$  and  $\mathbf{a}_j$  are assigned to the



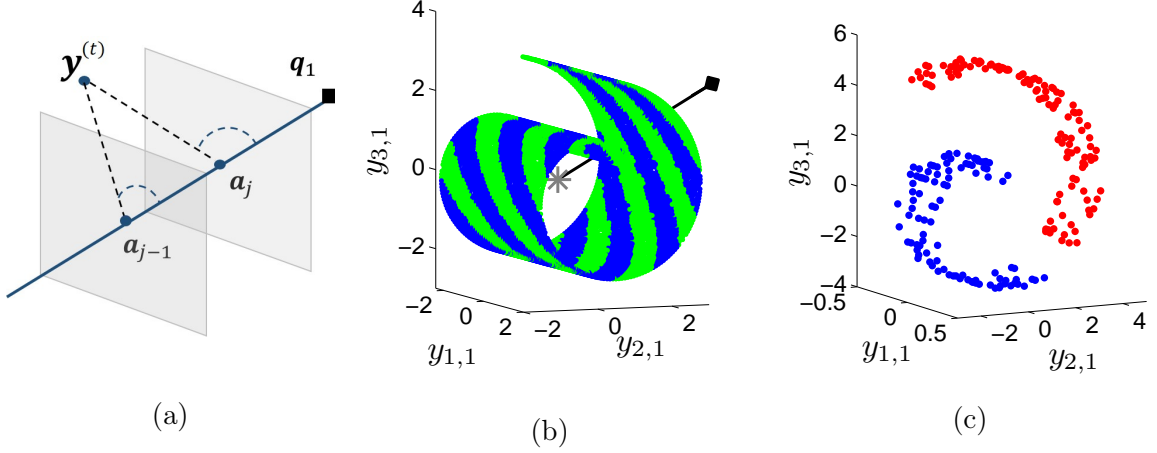


Figure 2.3: Clustering three dimensional swiss-roll data into slices. (a) Assigning points,  $\mathbf{y}^{(t)}$ 's, between hyperplanes through the points  $\mathbf{a}_{j-1}$  and  $\mathbf{a}_j$  into the cluster  $C_j^1$  by using the angles  $\mathbf{y}^{(t)} \mathbf{a}_j \mathbf{q}_1$  and  $\mathbf{y}^{(t)} \mathbf{a}_{j-1} \mathbf{q}_1$  as explained in Inequalities (2.9). (b) Using a single reference point (black square) located far from the data points, the dataset can be sliced into non-overlapping sections which are perpendicular to the line joining the centroid (gray asterisk) and the reference point, then slices are used to draw a grid pattern using splines. (c) A hole in the data will result in multiple sub-clusters (shown in different colors red and blue) within a cluster.

cluster  $C_j^1$ ,  $j = 1, \dots, n_C^1$ , by using the inequalities

$$\cos^{-1} \left( \frac{(\mathbf{y}^{(t)} - \mathbf{a}_{j-1})^T (\mathbf{q}_i - \mathbf{a}_{j-1})}{\|\mathbf{y}^{(t)} - \mathbf{a}_{j-1}\| \|\mathbf{q}_i - \mathbf{a}_{j-1}\|} \right) < \pi/2 \quad , \quad \cos^{-1} \left( \frac{(\mathbf{y}^{(t)} - \mathbf{a}_j)^T (\mathbf{q}_i - \mathbf{a}_j)}{\|\mathbf{y}^{(t)} - \mathbf{a}_j\| \|\mathbf{q}_i - \mathbf{a}_j\|} \right) \geq \pi/2, \quad (2.9)$$

for  $t = 1, \dots, T$  with  $i = 1$  (Figure 2.3(a) and (b)). The clustering method is adapted to detect gaps or holes in the data by setting a threshold on the minimum distance between neighboring points based on a nearest neighbor search [97]. If a set of points do not satisfy the threshold, they are automatically assigned into another cluster giving rise to sub-clusters [84], as shown in Figure 2.3(c).

The above procedure is repeated for the second reference point  $\mathbf{q}_2 = \boldsymbol{\mu} + 3\mathbf{v}_2\sigma_2$

to partition the dataset along the line joining  $\boldsymbol{\mu} \pm 3\mathbf{v}_2\sigma_2$  using Equation (2.8) with  $i = 2$ . Then, Equation (2.9) is used for  $t = 1, \dots, T$  with  $i = 2$  to make another set of clusters  $\mathcal{C}_j^2; j = 1, \dots, n_C^2$ . The resulting clustering algorithm inputs the data  $\mathcal{D}$ , and the number of clusters  $n_C^{1,2}$  with respect to each reference point, which can be set on the basis of data density in each direction. Choosing the best values for parameters  $n_C^1$  and  $n_C^2$  in the clustering algorithm is important to have an accurate embedding which is further addressed in Chapter 6 where we include the discussion. The set of clusters for each reference point are stored for subsequent operations. The clustering step is presented as Algorithm 2.

---

**Algorithm 2** *Clustering* : Clustering algorithm makes two sets of non-overlapping slices of data, one from each reference point. This inputs data matrix  $\mathcal{D}$  and number of clusters with respect to the first ( $n_C^1$ ) and second ( $n_C^2$ ) reference points. As the output, this produces two sets of clusters  $\mathcal{C}_j^i; j = 1, \dots, n_C^i$  for  $i = 1, 2$  from  $\mathcal{D}$  such that one set for each reference point.

---

- 1: **procedure** CLUSTERING( $\mathcal{D}, n_C^1, n_C^2$ )
  - 2:     Compute the mean  $\boldsymbol{\mu} \in \mathbb{R}^d$  of the input data  $\mathcal{D}$ .
  - 3:     Perform PCA [59, 68] on the input data matrix  $\mathcal{D}_m = [\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}, \dots, \mathcal{Y}^{(T)}]$  to obtain two largest principal components  $(\mathbf{v}_1, \sigma_1)$  and  $(\mathbf{v}_2, \sigma_2)$ , where  $\mathbf{v}_i$  is the  $d$ -dimensional coefficients and  $\sigma_i$  is the eigenvalue of the  $i$ -th PC for  $i = 1, 2$ .
  - 4:     In order to assure that two reference points are in the directions of two PCs from the mean, compute the first reference point  $\mathbf{q}_1 = \boldsymbol{\mu} + 3\sigma_1\mathbf{v}_1$ , and the second  $\mathbf{q}_2 = \boldsymbol{\mu} + 3\sigma_2\mathbf{v}_2$ .
  - 5:     **for** each reference point  $\mathbf{q}_i; i = 1, 2$  **do**
  - 6:         Segment the line joining the reference points  $\mathbf{q}_i = \boldsymbol{\mu} + 3\sigma_i\mathbf{v}_i$  and the point  $\tilde{\mathbf{q}}_i = \boldsymbol{\mu} - 3\sigma_i\mathbf{v}_i$  into  $n_C^i$  parts with equal width using the ratio formula given in Equation (2.8) to obtain a set of points  $\mathbf{a}_j; j = 0, \dots, n_C^i$  [104], where  $\mathbf{a}_0 = \tilde{\mathbf{q}}_i$  and  $\mathbf{a}_{n_C^i} = \mathbf{q}_i$ .
  - 7:         For  $j = 1, \dots, n_C^i$ , choose data between hyperplanes, which are made through points  $\mathbf{a}_{j-1}$  and  $\mathbf{a}_j$  normal to the line joining  $\tilde{\mathbf{q}}_i$  and  $\mathbf{q}_i$ , into the cluster  $\mathcal{C}_j^i$  by satisfying Inequalities (2.9).
  - 8:     **end for**
  - 9: **end procedure**
-

### 2.1.2 Smoothing the manifold

The clustering step is followed by smoothing, where cubic splines are used to represent the data in clusters. This approach of using a spline to approximate the data is similar to forming a principal curve on a set of points [62, 15]. Briefly, the principal curve runs smoothly through the dataset such that any point on the curve coincides with the average of all points that project orthogonally onto this point [62]. The algorithm in [62] searches for a parameterized function that satisfies this condition of self-consistency. In the context of smoothing splines on a cluster with  $n_s$  points  $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n_s)}$ , the principal curve  $\mathbf{g}(\lambda)$  parameterized in  $\lambda \in [0, 1]$  minimizes over all smooth functions  $\mathbf{g}$  [62]

$$\sum_{k=1}^{n_s} \|\mathbf{y}^{(k)} - \mathbf{g}(\lambda_k)\|^2 + p' \int_0^1 \|\mathbf{g}''(\lambda)\|^2 d\lambda, \quad (2.10)$$

where  $p'$  weights the smoothing of the spline and  $\lambda_k \in [0, 1]$  for  $k = 1, \dots, n_s$ . As  $\mathbf{y}^{(k)} \in \mathbb{R}^d$ , Equation (2.10) yields  $d$  individual expressions such that one for each dimension in the input space.

Since the principal curve is created in a multi-dimensional space, it is not necessarily follow the general curvature of the dataset. To follow the general curvature of the points that are input to the Equation (2.10), we perform an additional operation to build an geodesic within each cluster before making a principal curve. The geodesics are created using the *range-search*<sup>1</sup>[14], with range distance set as the cluster width  $6\sigma_1/n_C^1$  or  $6\sigma_2/n_C^2$  depending on the reference point. Specifically, first we search all the points within the specific distant to the  $i$ -th point using the range-search and adjacent

---

<sup>1</sup>A neighbor search which choses all neighbors within a specific distance.

them with the  $i$ -th point. We continue this search for all the agents and make a neighborhood graph. An *geodesic* is a shortest path between any two nodes (points) in the graph constructed using the Dijkstra's algorithm [42, 81]. The longest geodesic within the cluster is used to represent the full range and curvature of the cluster. Using  $n_g$  number of  $d$ -dimensional points  $\{\mathbf{y}^{(l)} = [y_1^{(l)}, \dots, y_d^{(l)}]^T | l = 1, \dots, n_g\}$  on the longest geodesic, the  $d$ -dimensional cubic smoothing spline  $\mathbf{S}(\lambda) = [S_1(\lambda), S_2(\lambda), \dots, S_d(\lambda)]^T$  parameterized in  $\lambda \in [0, 1]$  is computed as a piecewise polynomial that minimizes the quantity [62, 17]

$$p \sum_{l=1}^{n_g} |y_j^{(l)} - S_j(\lambda_l)|^2 + (1-p) \int_{\lambda_1}^{\lambda_{n_g}} [S_j''(\lambda)]^2 d\lambda, \quad (2.11)$$

where  $p \in [0, 1]$ , in each dimension  $j = 1, \dots, d$ . Thus,  $p$  weights the sum of square error and  $(1-p)$  weights the smoothness of the spline such that  $p = 0$  gives the natural cubic smoothing spline and  $p = 1$  gives an exact fit (Figure 2.4(a)). This procedure is repeated for all clusters for each reference point (Figure 2.4(b)) resulting in a grid-like surface of smoothing splines. We denote cubic smoothing splines made with respect to the first reference point by  $\mathbf{S}_j^1; j = 1, \dots, n_C^1$ , and second reference point by  $\mathbf{S}_j^2; j = 1, \dots, n_C^2$ . The smoothing step is given as Algorithm 3.

**Definition 2.1.1. Geodesic** [41]. *A geodesic of a graph is a shortest path between two vertices.*

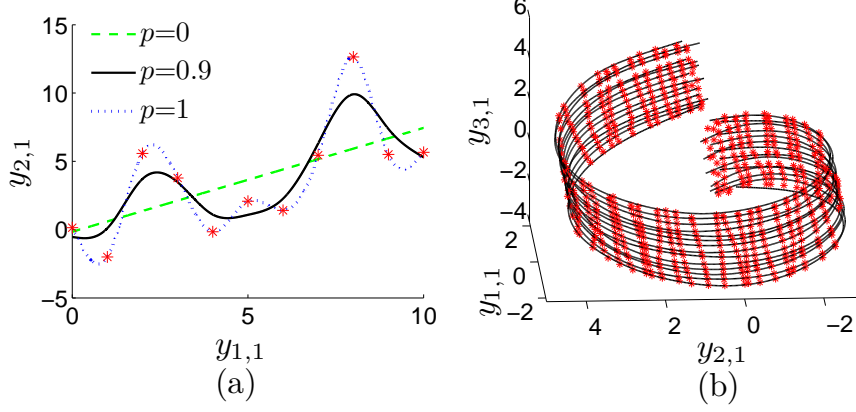


Figure 2.4: Approximating clusters with smoothing splines. (a) A cubic smoothing spline is fit onto two-dimensional data,  $\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, T\}$  for some  $T$ , where  $\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}]$  and  $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}]^T$ , using different values of the smoothing parameter  $p$ . (b) Smoothing splines fit clusters of a swiss-roll dataset from a single reference point.

---

**Algorithm 3** *Smoothing* : Smoothing algorithm produces two sets of cubic smoothing splines with respect to both reference points to represent the data in clusters. This algorithm inputs spline smoothing parameter  $p$  and two sets of clusters  $\mathcal{C}_j^i$ ;  $j = 1, \dots, n_{\mathcal{C}}^i$  for  $i = 1, 2$  made in the clustering algorithm, and outputs two sets of cubic smoothing splines  $\mathcal{S}_j^i$ ;  $j = 1, \dots, n_{\mathcal{C}}^i$  for  $i = 1, 2$ .

---

- 1: **procedure** SMOOTHING( $p$ ;  $\mathcal{C}_j^i$ ;  $j = 1, \dots, n_{\mathcal{C}}^i$  for  $i = 1, 2$ )
  - 2:     **for** each reference point  $\mathbf{q}_i$ ;  $i = 1, 2$  **do**
  - 3:         **for** all clusters  $\mathcal{C}_j^i$ ;  $j = 1, \dots, n_{\mathcal{C}}^i$  **do**
  - 4:             Compute the neighborhood graph using range-search with the distance set as the cluster width  $6\sigma_i/n_{\mathcal{C}}^i$ .
  - 5:             Compute the longest geodesic  $\mathcal{G}_j^i$  using Dijkstra's algorithm [42, 81].  $\mathcal{G}_j^i$  is a set of points, where each point is in  $\mathbb{R}^d$ .
  - 6:             For points in  $\mathcal{G}_j^i$ , use Expression (2.11) and the value of the smoothing parameter  $p$  to produce a smoothing spline  $\mathcal{S}_j^i \in \mathbb{R}^d$  representation for data in the cluster  $\mathcal{C}_j^i$ .
  - 7:         **end for**
  - 8:     **end for**
  - 9: **end procedure**
-

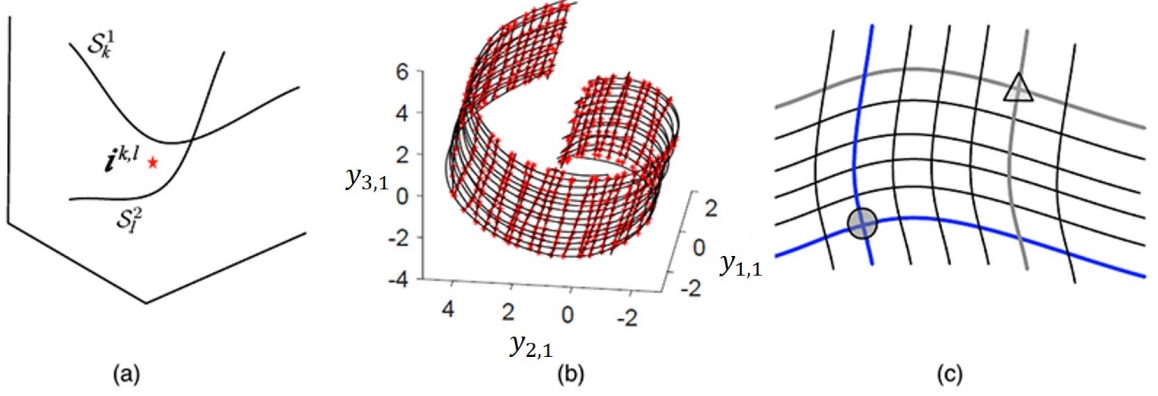


Figure 2.5: Embedding the data into two-dimensional coordinates. (a) The intersection point (red) between two splines lies midway on the shortest distance between them. (b) Intersection points (red) for a swiss-roll, and (c) the two-dimensional coordinate of a point (triangle) is the geodesic distance along the intersecting splines (grey) to the axis splines (blue) from an origin (circle).

### 2.1.3 Embedding

Here we compute intersections of splines and construct an intrinsic coordinate system for embedding. Intersection points between all pairs of splines, one from each reference point, are located as landmarks to embed new points onto the PM. Since the splines from two reference points may not actually intersect each other unless  $p = 1$  for all splines, virtual intersection points between two non-intersecting splines are located at the midpoint on the shortest distance between them (Figure 2.5(a)). The embedding algorithm loops through all pairs of splines to locate such points, and once located they are denoted as  $\mathbf{i}^{k,l} \in \mathbb{R}^d$  where  $(k, l) \in \mathcal{S}_k^1 \times \mathcal{S}_l^2$  (Figure 2.5(b)). In order to define the embedding coordinates, an intersection point is selected randomly and assigned as the manifold origin  $\mathbf{O}$  of the PM. The set of smoothing splines corresponding to the origin are called axis splines (Figure 2.5(c)).

The embedding coordinates of a point  $\mathbf{z} \in \mathbb{R}^d$  are defined in terms of the dis-

tances along the axis splines as shown in Figure 2.6. For that, first we find the closest intersection  $\tilde{\mathbf{z}}$  for  $\mathbf{z}$  in Euclidean distance as

$$\tilde{\mathbf{z}} = \operatorname{argmin}_{k,l} \|\mathbf{i}^{k,l} - \mathbf{z}\| \quad (2.12)$$

and unit tangents to splines at this point are called the local spline directions and denoted by  $(\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1, \hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2)$ . Distance from  $\mathbf{O}$  to  $\tilde{\mathbf{z}}$  is computed along the axis splines by

$$[\tilde{x}_1, \tilde{x}_2]^T = [d_k(\tilde{\mathbf{z}}, \mathbf{O}), d_l(\tilde{\mathbf{z}}, \mathbf{O})]^T, \quad (2.13)$$

where  $d_k(\mathbf{a}_1, \mathbf{a}_2)$  is the distance between points  $\mathbf{a}_1$  and  $\mathbf{a}_2$  along the spline  $k$ . We project the vector  $\tilde{\mathbf{z}} - \mathbf{z}$  on the local tangential directions  $(\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1, \hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2)$  as

$$\begin{aligned} \delta x_1 &= (\tilde{\mathbf{z}} - \mathbf{z})^T \hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1, \\ \delta x_2 &= (\tilde{\mathbf{z}} - \mathbf{z})^T \hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2, \end{aligned} \quad (2.14)$$

at the intersection point  $\tilde{\mathbf{z}}$ . The final embedding coordinates are obtained by summing the quantities in Equations (2.13) and (2.14) as

$$[x_1, x_2]^T = [\tilde{x}_1 + \delta x_1, \tilde{x}_2 + \delta x_2]^T. \quad (2.15)$$

The embedding step is given as Algorithm 4.

---

**Algorithm 4** *Embedding* : This produces a grid structure by approximating the intersections of splines, followed by embedding of a new point based on this structure. Embedding algorithm inputs a new point  $\mathbf{z} \in \mathbb{R}^d$  to embed and two sets of cubic smoothing splines  $\mathbf{S}_j^i$  ;  $j = 1, \dots, n_C^i$  for  $i = 1, 2$  produced by smoothing algorithm, and this outputs two dimensional embedding coordinates  $[x_1, x_2]^T$  of  $\mathbf{z}$ .

---

- 1: **procedure** EMBEDDING( $\mathbf{S}_j^i, j = 1, \dots, n_C^i$  for  $i = 1, 2$  ; a point  $\mathbf{z} \in \mathbb{R}^d$  for embedding)
  - 2:     **for** all pairs  $(k, l)$  of smoothing splines  $\mathbf{S}_k^1 \times \mathbf{S}_l^2$  belonging to reference points 1 and 2 **do**
  - 3:         Approximate the minimum distance between the two splines after discretizing each spline.
  - 4:         Choose the midpoint between the two closest points of the splines as the intersection point  $\mathbf{i}^{k,l} \in \mathbb{R}^d$ .
  - 5:     **end for**
  - 6:     Pick a random intersection point as the origin  $\mathbf{O} \in \mathbb{R}^d$ , and smoothing splines corresponding to the origin as axis splines.
  - 7:     Find the closest intersection point  $\tilde{\mathbf{z}}$  for  $\mathbf{z}$  in terms of Euclidean distance by Equation (2.12), and the unit tangents to the splines at this point are called the local spline directions  $(\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1, \hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2)$ .
  - 8:     Compute the distances  $[\tilde{x}_1, \tilde{x}_2]^T$  from the manifold origin  $\mathbf{O}$  to  $\tilde{\mathbf{z}}$  along axis splines by Equation (2.13).
  - 9:     Project the vector  $\tilde{\mathbf{z}} - \mathbf{z}$  onto the local coordinate system created using the spline directions as Equation (2.14) at the intersection point  $\tilde{\mathbf{z}}$  and find the projections  $[\delta x_1, \delta x_2]^T$ . The final embedding coordinates are given as  $x_1 = \tilde{x}_1 + \delta x_1$ ,  $x_2 = \tilde{x}_2 + \delta x_2$ .
  - 10: **end procedure**
-



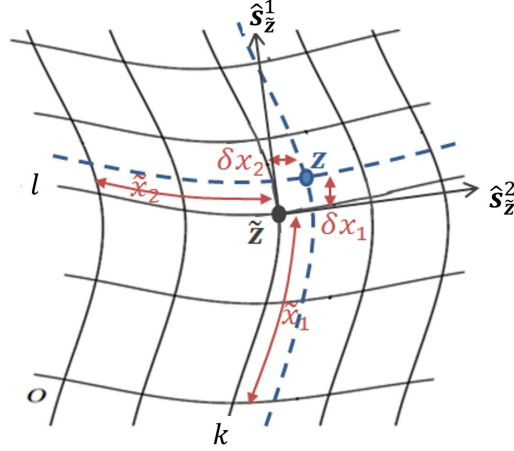


Figure 2.6: Embedding a higher-dimensional point  $\mathbf{z}$  onto the manifold using distances along the axis splines from the origin  $\mathbf{O}$  to the closest intersection  $\tilde{\mathbf{z}}$  and the local distance form  $\tilde{\mathbf{z}}$  to  $\mathbf{z}$ . Local distances are computed by projecting the distance  $\tilde{\mathbf{z}} - \mathbf{z}$  onto the unit tangent vectors  $\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^1$  and  $\hat{\mathbf{s}}_{\tilde{\mathbf{z}}}^2$ .

#### 2.1.4 Mapping between the manifold and raw data

In this section, we propagate points in the embedding space to the higher-dimensional space which is similar to find  $\Phi^{-1}$  in Equation 2.2. All the intersections and splines describing the intrinsic coordination system are available at this point to recover the point  $\mathbf{y}^{(t)} \in \mathbb{R}^d$  corresponds to the point  $\mathbf{x}^{(t)} \in \mathbb{R}^2$ .

The mapping between the manifold and the raw data is created by inverting the steps of the embedding algorithm. In particular, a two-dimensional point  $\mathbf{x} = [x_1, x_2]^T$  in the embedding space is located in terms of the embedding coordinate  $\mathbf{x}^{k,l} = [x_1^{k,l}, x_2^{k,l}]^T$  of the closest intersection point  $\mathbf{i}^{k,l} \in \mathbb{R}^d$ . We then complete the local coordinate system by finding the two adjacent intersection points. The resulting three points on the PM, the origin and the two adjacent points denoted by  $\{\mathbf{x}^{k,l} = [x_1^{k,l}, x_2^{k,l}]^T, \mathbf{x}^{k,l+1} = [x_1^{k,l+1}, x_2^{k,l+1}]^T, \mathbf{x}^{k+1,l} = [x_1^{k+1,l}, x_2^{k+1,l}]^T\}$ , are used to

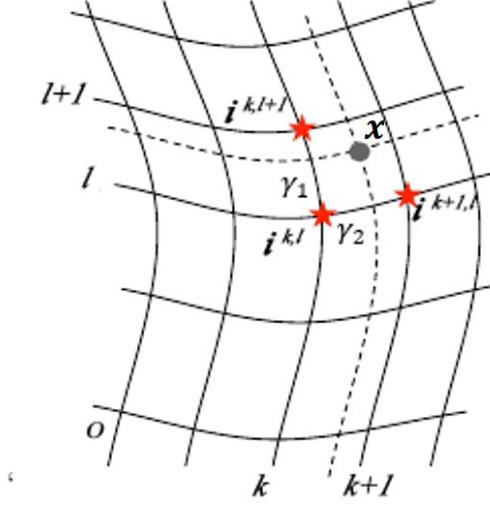


Figure 2.7: The mapping of the point  $\mathbf{x}$  on the underlying manifold to the original high-dimensional space is done by finding the projections  $\gamma_1, \gamma_2$  locally and propagating the same into the high dimensions. Local approximation can be created using adjacent splines  $\{k, k+1\}$ , and  $\{l, l+1\}$  and their intersections  $\{\mathbf{i}^{k,l}, \mathbf{i}^{k,l+1}, \mathbf{i}^{k+1,l}\}$ .

solve for ratios  $\gamma_1$  and  $\gamma_2$  such that

$$\begin{aligned}\gamma_1 &= \frac{x_2 - x_2^{k,l}}{x_2^{k,l+1} - x_2^{k,l}}, \\ \gamma_2 &= \frac{x_1 - x_1^{k,l}}{x_1^{k+1,l} - x_1^{k,l}}.\end{aligned}\tag{2.16}$$

The ratios are then propagated to the high-dimensional space (Figure 2.7) by using the same relation on the corresponding points in the higher dimension  $\{\mathbf{i}^{k,l}, \mathbf{i}^{k,l+1}, \mathbf{i}^{k+1,l}\}$  to obtain the high-dimensional point

$$\mathbf{y} = \mathbf{i}^{k,l} + (\mathbf{i}^{k,l+1} - \mathbf{i}^{k,l})\gamma_1 + (\mathbf{i}^{k+1,l} - \mathbf{i}^{k,l})\gamma_2.\tag{2.17}$$

## 2.2 Examples

In this section, we evaluate and compare the PM finding algorithm with other DR methods on three different datasets: a three-dimensional paraboloid, a swiss-roll constructed with a high additive noise (noisy swiss-roll), and a simulation of collective motion. Specifically, we use standard versions of eight different DR methods implemented in the Matlab Toolbox for DR [134] with default parameters listed in Table 2.1.

Table 2.1: Default parameters implemented in Matlab Toolbox for DR [134] are used in the paraboloid and noisy swiss-roll examples.

Method	Parameters
MDS [75, 76]	none
Isomap [130]	nearest neighbors = 12
Diffusion maps [77]	variance = 1 and operator on the graph = 1
KPCA [119]	variance = 1 and Gaussian kernel is used
LSE [141]	nearest neighbors = 12
LLE [113]	nearest neighbors = 12
HLLE [44]	nearest neighbors = 12
LE [13]	nearest neighbors = 12 and variance = 1

### 2.2.1 Paraboloid

A three-dimensional paraboloid (Figure 2.8(a)) of having 2000 points ( $T$ ) is generated using

$$y_{3,1}^{(t)} = \left(y_{1,1}^{(t)}\right)^2 + \left(y_{2,1}^{(t)}\right)^2 + \epsilon, \quad (2.18)$$

with  $y_{1,1}^{(t)}, y_{1,2}^{(t)} \in [-2, 2]$ , and  $\epsilon \sim \mathbb{U}(-0.05, 0.05)$  is a noise variable sampled from a *uniform distribution* ranging between -0.05 and 0.05. This example represents a single-agent system revolving in three dimensions, thus it produce the data  $\mathcal{D} = \{\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}] \mid t = 1, \dots, 2000\}$  ;  $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}, y_{3,1}^{(t)}]^T$ . Since the data is generated along the  $y_{1,1}, y_{2,1}$  axes (Equation (2.18)), the two reference points are selected along the directions just outside the range for each value. We run the clustering algorithm with the value of  $n_c^{1,2} = 14$  for each reference point to generate corresponding sets of clusters with Algorithm 2. Next, we run the smoothing algorithm with a smoothing parameter  $p = 0.9$  to produce a set of representative splines for the data (Figure 2.8(b)). Finally, we embed the manifold by first finding the intersection points and then computing the distance of each point from the axis splines (Figure 2.8(c)).

**Definition 2.2.1.** *Uniform distribution [24]. A distribution having a random variable  $x$  restricted to a finite interval  $[a, b]$  and a constant density function*

$$\mathbb{U}(x) = \begin{cases} \frac{1}{b-a} & , a \leq x \leq b \\ 0 & , otherwise. \end{cases}$$

The two-dimensional embedding manifold generated using our algorithm preserves the topology of the data as shown in Figure 2.8(c). Furthermore, the axes in the embedding manifold retain the scale of the original data in terms of the distance between individual points. For a comparison, we run other DR methods on this data set with parameters specified in Table 2.1 (Figure 2.9). Results show that except Isomap and

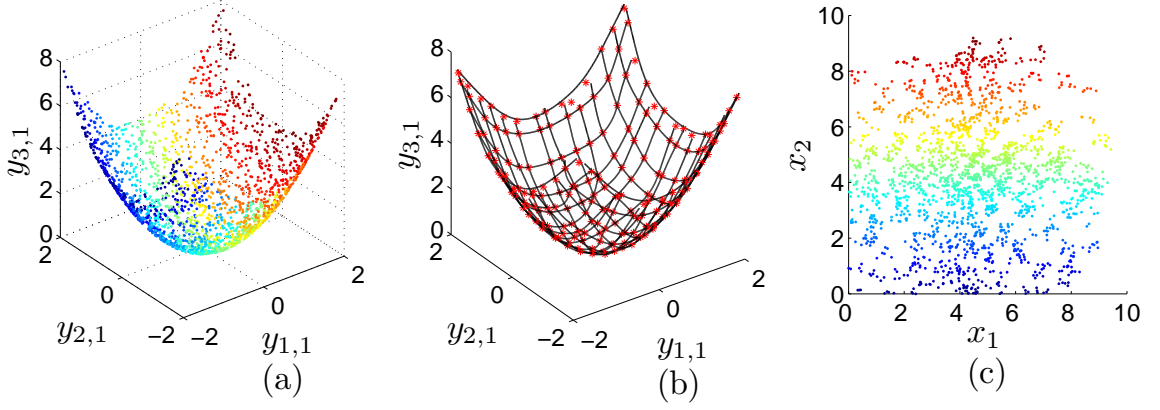


Figure 2.8: Embedding of raw data from (a) a three-dimensional paraboloid with 2000 points is attained after (b) smoothing the data by using cubic smoothing splines (black). The red points herein show the virtual intersections of splines. (c) Two dimensional embedding manifold of the data along with the intrinsic coordinates. Points are colored to highlight the relative configuration.

the proposed method, none of the DR methods are able to preserve the two-dimensional square-shape embedding after reducing the dimensionality. However, we observe that the PM method retains the scale of the manifold than that of Isomap.

### 2.2.2 A noisy swiss-roll

In the second example, we run our algorithm on a three-dimensional noisy swiss-roll dataset having 2500 points ( $T$ ) (Figure 2.10(a)). The dataset is generated using

$$\begin{aligned}
 y_{1,1}^{(t)} &= \left(\theta_1^{(t)}\right)^{0.8} \cos \theta_1 + \epsilon \\
 y_{2,1}^{(t)} &= \left(\theta_1^{(t)}\right)^{0.8} \sin \theta_1 + \epsilon \\
 y_{3,1}^{(t)} &= \theta_2^{(t)} + \epsilon
 \end{aligned} \tag{2.19}$$

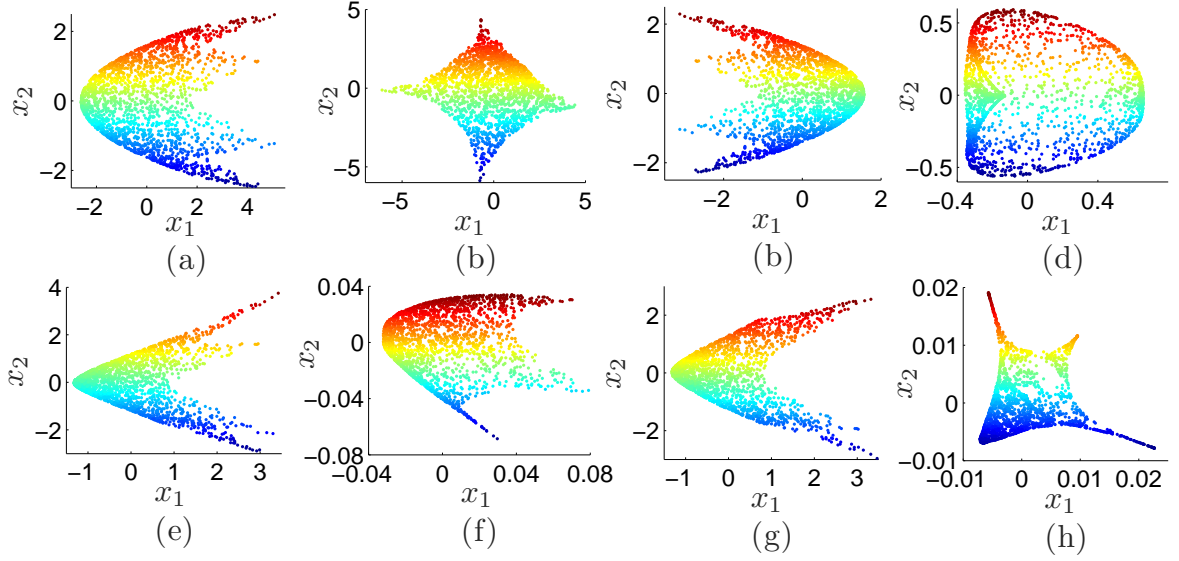


Figure 2.9: Two-dimensional embeddings of the paraboloid by MDS (a), Isomap (b), Diffusion maps (c), KPCA (d), LSE (e), LLE (f), HLLE (g), and LE (h). Points are colored to highlight the relative configuration with the raw data in Figure 2.8(a).

where  $\theta_1^{(t)} \in [0.25\pi, 2.5\pi]$ ,  $\theta_2^{(t)} \in [-2, 2]$ , and  $\epsilon \sim \mathbb{U}(-0.4, 0.4)$  is a noise variable sampled from a uniform distribution ranging between -0.4 and 0.4. This example also represents a single-agent system having data  $\mathcal{D} = \{\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}] \mid t = 1, \dots, 2500\}$ ;  $\mathbf{y}_1^{(t)} = [y_{1,1}^{(t)}, y_{2,1}^{(t)}, y_{3,1}^{(t)}]^T$ . Two reference points are chosen along two major principal component directions  $\mathbf{v}_1$  and  $\mathbf{v}_2$  at a distance of  $3\sigma_1$  and  $3\sigma_2$  units away from the data centroid  $\boldsymbol{\mu}$ .

We run the clustering algorithm with the value of  $n_C^{1,2} = 15$  for each reference point, however, due to folds and therefore holes in the data, the clusters along  $\mathbf{v}_2$  are further split into 42 subclusters. The value of the smoothing parameter  $p$  is set to 0.75. Figure 2.10(b) and Figure 2.10(c) show the smoothing splines in a grid pattern that are used to embed the manifold, and the resulting embedding. For a comparison, we run other DR methods on this data set with parameters specified in Table 2.1

(Figure 2.11). Comparison between existing DR methods show that while only Isomap and KPCA are able to flatten the swiss-roll into two dimensions, Isomap preserves the general rectangular shape of the flattened swiss-roll. A majority of DR methods suppress one component out of the two principal components in the raw data revealing an embedding that resembles a one-dimensional curve. In contrast, the PM approach is able to embed the swiss-roll into a rectangle while preserving the scale.

We note that by changing the value of the smoothing parameter we control for the level of noise rejection in the original data, a feature that is not available in the existing DR algorithms [135]. Although the splines form illegal connections, our algorithm is still able to preserve the two-dimensional embedding; this is due to the inherent embedding step which is able to overcome such shortcuts while computing the low-dimensional coordinates. Furthermore, we note that our method is able to better exclude noisy data from the embedding automatically in the form of outliers; neighborhood based algorithms would be mislead to attempt to include these as valid points [130]. For example, the two-dimensional embedding obtained using Isomap shows an unforeseen bend demonstrating that, despite fewer outliers, the general topological structure is compromised more heavily. This global anomaly is a direct consequence of the Isomap’s attempt to include noisy points in the embedding; points that are otherwise ignored by the principal manifold.

### **2.2.3 Collective motion: simulation of predator mobbing**

In the third example of low-dimensional embedding, we generate a dataset representing a form of collective motion. We simulate point-mass particles revolving on a translat-

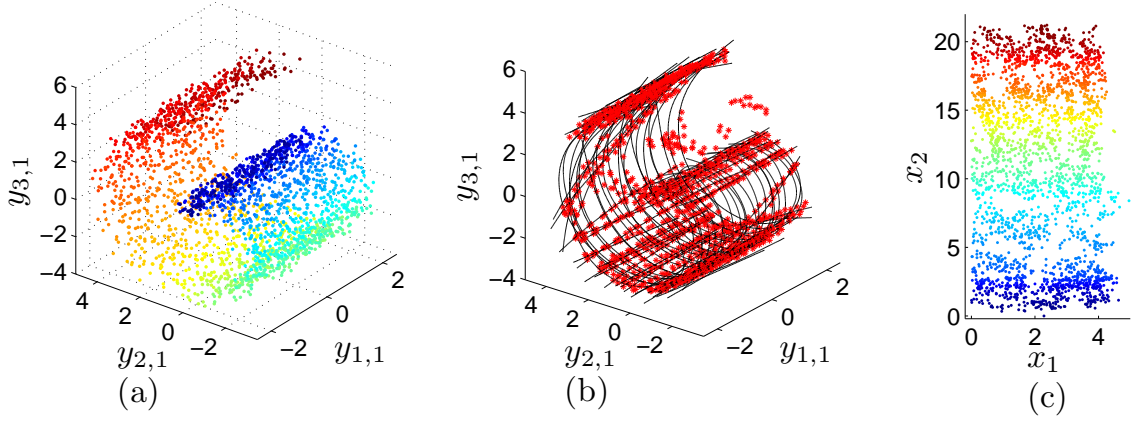


Figure 2.10: Two-dimensional embedding of the noisy swiss-roll with 2500 points. (a) Raw data of the noisy swiss-roll is clustered and (b) smoothened using splines (black) in order to obtain the principal manifold. The red points herein show the virtual intersections of splines. (c) Two dimensional embedding with intrinsic coordinates denoting the distance from the origin of the manifold coordinate. Points are colored to highlight the relative configuration.

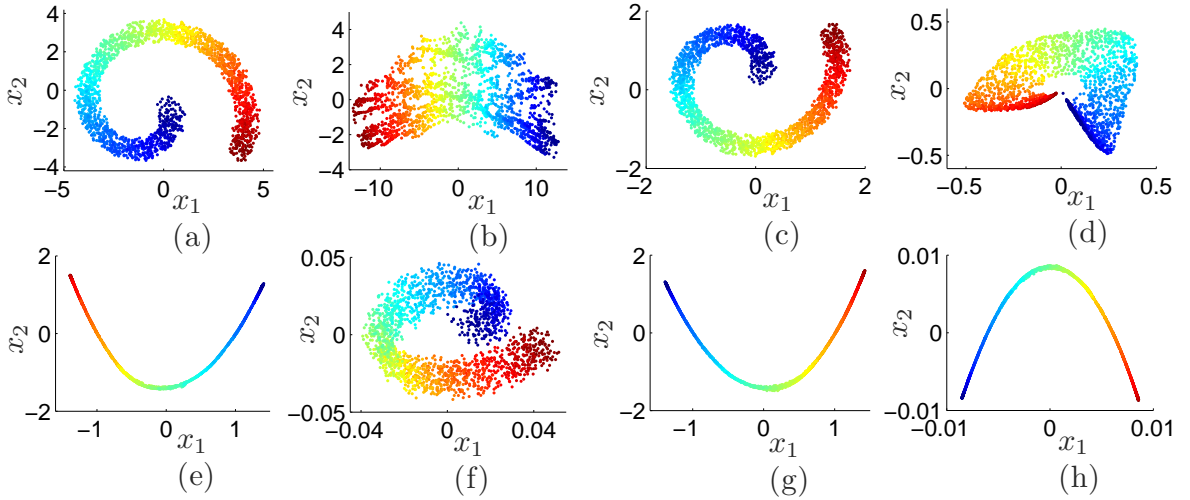


Figure 2.11: Two-dimensional embeddings of the noisy swiss-roll by MDS (a), Isomap (b), Diffusion maps (c), KPCA (d), LSE (e), LLE (f), HLLE (g), and LE (h). Points are colored to highlight the relative configuration with the raw data in Figure 2.10(a).



ing circle to represent a form of behavior called predator induced mobbing [43]. In prey animals, predator mobbing, or foraging around a predator, is often ascribed to the purpose of highlighting the presence of a predator [43]. In our idealized model, we assume that  $n$  agents ( $n = 20$ ) are revolving on the circumference of a circle whose center marks the predator and translating on a two-dimensional plane which is orthogonal to the axis of the revolution. To generate a two-dimensional trajectory with  $\omega$  revolutions around a translating center, we represent the two-dimensional position of an agent  $i$  at  $t$ ,  $\mathbf{y}_i^{(t)} = [y_{1,i}^{(t)}, y_{2,i}^{(t)}]^T$ ,  $\mathbf{y}_i^{(t)} \in \mathbb{R}^2$

$$\mathbf{y}_i^{(t)} = r_i^{(t)} \begin{bmatrix} \cos(2\pi\omega t/T + \pi i/n) \\ \sin(2\pi\omega t/T + \pi i/n) \end{bmatrix} + t\mathbf{w}_p^{(t)} + \boldsymbol{\epsilon}^{(t)}, \quad (2.20)$$

where  $\mathbf{w}_p^{(t)} = [1, 1]^T/80$  is the velocity of the predator,  $r_i^{(t)} = 2$  is the radius of the rotation of the agent  $i$ ,  $T$  is the total time steps, and  $\boldsymbol{\epsilon}^{(t)} \sim \mathbb{N}(\mathbf{0}, \mathbf{1}/100)$  is the Gaussian noise variable with mean  $\mathbf{0} \in \mathbb{R}^2$  and standard deviation  $\mathbf{1}/100 \in \mathbb{R}^2$ . In this setup, the data is  $\mathcal{D} = \{\mathbf{y}^{(t)} = [\mathbf{y}_1^{(t)}; \dots; \mathbf{y}_{20}^{(t)}] \mid t = 1, \dots, T\}$  where  $\mathbf{y}_i^{(t)} = [y_{1,i}^{(t)}, y_{2,i}^{(t)}]^T$ . The first term of Equation (2.20) assigns the relative positions of  $n$  agents onto equally spaced points on the circumference of a half circle which rotates with the speed of  $w_i^{(t)}$  units, and the second term gives the velocity of the virtual center (predator) of the translating circle. The agent-dependent phase  $\pi i/n$  creates spatial separation of individual members around the virtual center.

Figure 1.4(a) shows the resulting trajectories for 2000 time steps ( $T$ ) with  $\omega = 14$  in a  $d$ -dimensional space (note that  $d = 2n$  in this case, where  $n$  is the number of

agents). The interaction between the agents is captured in the form of noise  $\epsilon^{(t)}$ . A high value for  $\epsilon^{(t)}$  means that the agents interact less with each other. The underlying manifold obtained by Isomap is shown in Figure 1.4(c) with the start and end points of the trajectory. For comparison, we also run the Isomap algorithm on this dataset with a neighborhood parameter value ( $\alpha$ ) set to 10. To ensure that the embedding is consistent and robust to our choice of neighborhood parameter, we run the algorithm with neighborhood parameters 5 and 15 and verify that the output is similar. Figure 1.4(b) shows the resulting two-dimensional embedding with different neighborhood parameters, while Figure 1.4(c) shows two dimensional embedding.

We run our algorithm over the data with smoothing parameter  $p = 0.9$  and  $n_c^{1,2} = 20, 3$ , respectively. Unlike the paraboloid and the noisy swiss-roll, the predator mobbing dataset represents a multi-agent dynamical system. Therefore, we also run Isomap over this data to compare the embedding results. We transform the original data by a clockwise rotation of  $\pi/4$  such that

$$\tilde{\mathbf{y}}_i^{(t)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \mathbf{y}_i^{(t)} \quad (2.21)$$

and consider this be the ground truth. The position of the first agent  $\tilde{\mathbf{y}}_1^{(t)} = [\tilde{y}_{1,1}^{(t)}, \tilde{y}_{2,1}^{(t)}]^T$  is used for further analysis as all the agents in this system have approximately similar dynamics. The distribution  $\{\tilde{\mathbf{y}}_1^{(t)} | t = 1, \dots, 2000\}$  serves as a descriptive global observable of the predator mobbing agents separated by a constant phase, than, for example, the group centroid, where the revolving motion is suppressed due to the instantaneous

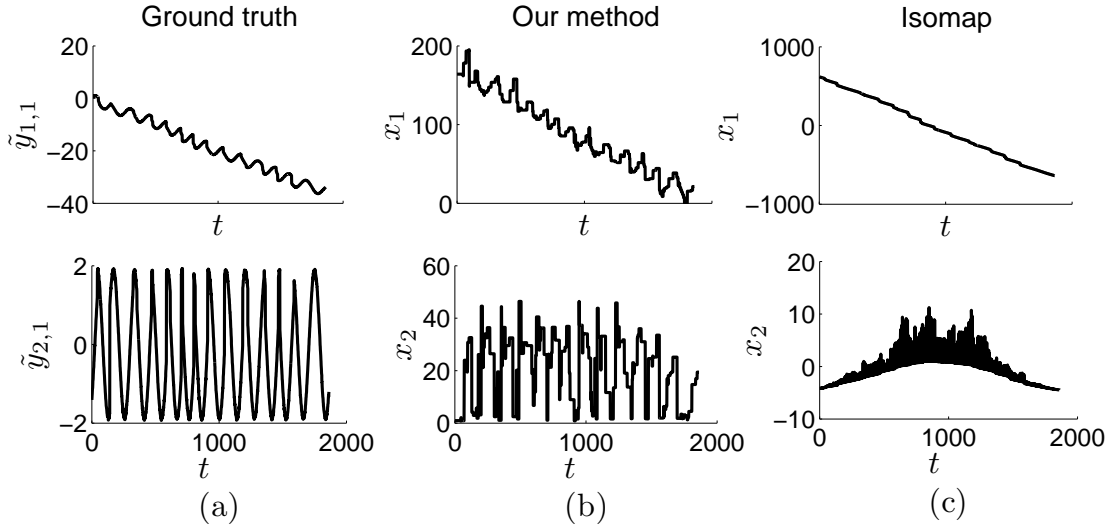


Figure 2.12: Embedding a forty dimensional dataset of an idealized simulation of predator mobbing where twenty particles move around a translating circle. A comparison of individual axes of the embedding produced by our method and those produced using Isomap shows that our method performs distinctly better in terms of preserving the time signature as well as range of the axes. The nearest neighbor parameter used for Isomap is equal to 10.

two-dimensional arrangement.

Two time series of the coordinates of the first agent's position are shown separately in Figure 2.12(a). Therein, the top panel represents the transverse motion while the bottom panel represents the oscillatory motion. As the length of this trajectory is 180 units in 2000 time steps, the size-of-unit is  $180/2000 = 0.09$ . Figure 2.12(b) and Figure 2.12(c) represent the time series of the embedding under our method and Isomap, respectively. Selection of axes in the intrinsic coordinate system is an automated process in any DR method which is specifically done using first two largest principal components in our method. As the ranges of the axes are fully dependent on how they are chosen, quantitative comparison between embedding results is not effective. However,

visual comparison of the variations of the time series convinces that the embedding result in our method is closely represents the ground truth comparative to that of the Isomap.

## 2.3 Performance analysis

In this section, we assess the performance of the method using two factors; error and computational complexity. Error associated with smoothing, noise, and data density are computed separately using a distance-preserving matrix. The computational complexity encounters from algorithms clustering, smoothing, and embedding are also computed.

### 2.3.1 Error analysis

We compute the error of the embedding using a distance-preserving metric between the original and the embedding data in terms of distance of graph connectivity. For both the original and the embedding data, we first search  $\alpha$ -nearest neighbors through the algorithm given in [51] and then produce two individual weighted graphs based on points connectivity in the data sets. A graph constructed through nearest neighbor search is a *simple graph* that does not contain self-loops or multiple edges.

**Definition 2.3.1.** *Simple graph [9]. A simple graph is an undirected graph that does not contains loops (edges connected at both ends to the same vertex ) and multiple edges (more than one edge between any two different vertices).*

We compute the distance matrix  $A$  for the original data as

$$A_{i,j} = \begin{cases} d(i,j) & ; \exists \text{ an edge } ij \text{ in the graph of the original data,} \\ 0 & ; \text{otherwise,} \end{cases} \quad (2.22)$$

and the distance matrix  $\tilde{A}$  for the embedding data as

$$\tilde{A}_{i,j} = \begin{cases} d(i,j) & ; \exists \text{ an edge } ij \text{ in the graph of the embedding data,} \\ 0 & ; \text{otherwise.} \end{cases} \quad (2.23)$$

Here,  $A_{ij}$  and  $\tilde{A}_{ij}$  are the  $(i,j)$ -th entries of the adjacency matrices  $A$  and  $\tilde{A}$ , respectively, and  $d(i,j)$  is the Euclidean distance between nodes  $i$  and  $j$ . For  $n$  points, this metric is computed as the normalized number of pairwise errors between entries of the adjacency distance matrices as

$$\Gamma(\tilde{A}, A) = \frac{1}{n\alpha} \sum_{i,j} | \tilde{A}_{ij} - A_{ij} | . \quad (2.24)$$

Equation (2.24) is a modification of the structure preserving metric in [121] where the connectivity is replaced by the distance and the denominator  $n^2$  with the number of edges  $n\alpha$ . We study the dependence of  $\Gamma(\tilde{A}, A)$  on the smoothing parameter, the noise in the dataset, and the data density.

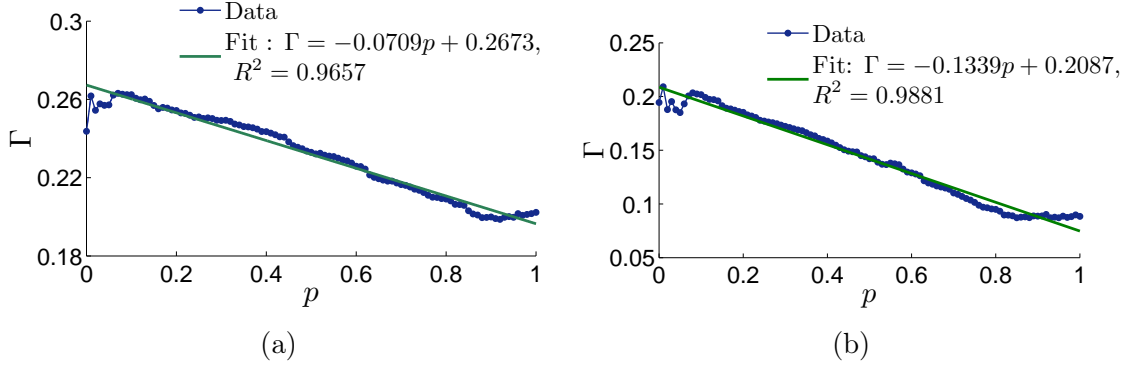


Figure 2.13: Variation of  $\Gamma$  with respect to  $p$  and the corresponding linear fit for (a) a noise free swiss-roll, and (b) a noise free paraboloid.  $\Gamma$  is computed using Equation (2.24) between raw and embedded data.

### Error dependence on smoothing

The primary input to the algorithm is the smoothing parameter that controls the degree of fitness and noise-rejection by the cubic smoothing splines. Figure 2.13 shows the dependence of the distance preservation error  $\Gamma$  in term of adjacency distance on the smoothing parameter  $p$ . Using a noise-free swiss-roll dataset and a noise free paraboloid each comprising 3000 points in three dimensions, we vary the smoothing parameter between 0 and 1 with an increment of 0.01 while keeping the location of reference points and cluster width consistent. The error dependence shows a linear decay in both cases with *coefficient of determination* or simply  $R^2$ -value<sup>2</sup> of 0.9728 in the swiss-roll and 0.9881 in the paraboloid. In the swiss-roll example, the error becomes nearly constant at a  $p = 0.88$ , beyond which the change in error for an increase in the value of  $p$  is negligible, contrast to the paraboloid example that saturates the error at around  $p = 0.82$ . A higher value of  $p$  improves the data-fit, however a low value

---

<sup>2</sup>The  $R^2$ -value ranges between 0 and 1, measures the goodness between data and the fit so that 1 is the best fit while 0 is the worst fit.

improves the smoothness. Since the graph connectivity is dependent on the relative point configuration, it is expected that  $\Gamma$  will rise with increasing smoothness. On the other hand, the error dependence shows that a given degree of smoothing can be attained beyond some value of  $p$  without an accompanied increase error. Thus, as a design parameter, the value  $p = 0.9$  may be used as a starting point for all datasets to investigate the embedding manifold.

**Definition 2.3.2. Coefficient of determination [9].** For a set of points  $\{z_i\}_{i=1}^T$  associated with fitted values  $\{f_i\}_{i=1}^T$ , coefficient of determination or  $R^2$ -value is defined as

$$R^2 = 1 - \frac{\sum_i^T (z_i - f_i)^2}{\sum_i^T (z_i - \bar{z})^2} ; \text{ where } \bar{z} = \sum_i^T z_i / T.$$

### Error with noise

To analyze the change in pairwise error with increase in noise, we create multiple same sized swiss-roll datasets generated using Equation (2.19) with noise value  $\epsilon$  ranging between 0 and 1.035 with an increment of 0.015. The number of points in the dataset are 3000 and the value of smoothing parameter  $p$  is 0.9. Figure 2.14 shows the variation of  $\Gamma$  with respect to  $\epsilon$ . The error  $\Gamma$  increases quadratically with  $R^2$ -value of 0.9983 when noise increases. This relationship shows that the error of the embedding is affected quadratically by the intense of the associated noise in the data.

### Data density

To analyze the dependence of  $\Gamma$  on data density we sub-sample the swiss-roll dataset such that the number of points are systematically increases. The amount of noise is

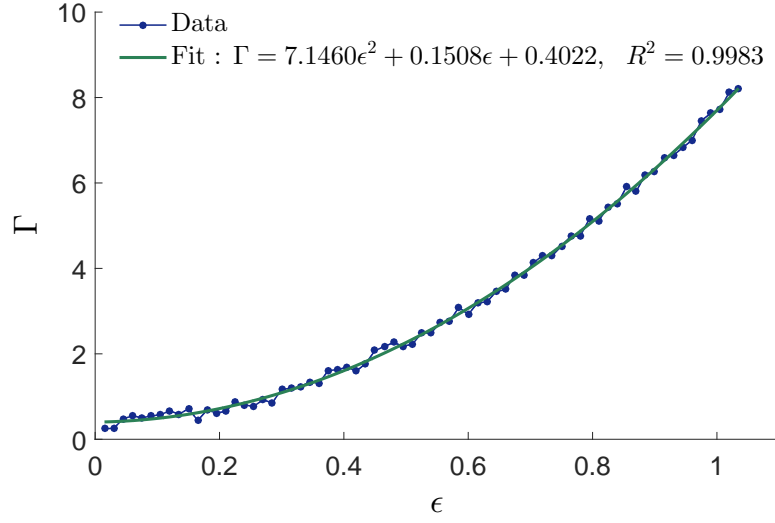


Figure 2.14: Variation of  $\Gamma$  with respect to noise ( $\epsilon$ ) and the corresponding quadratic fit  $\Gamma$  is computed between adjacency matrices for the raw data and embedded data by Equation (2.24).

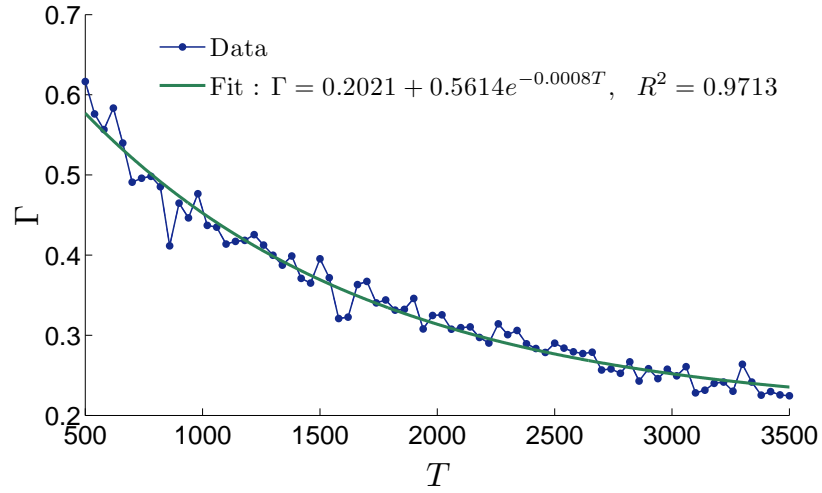


Figure 2.15: Variation of  $\Gamma$  with respect to number of points ( $T$ ) and the corresponding exponential fit with  $p = 0.9$  and  $\epsilon = 0.2$ .  $\Gamma$  is computed between adjacency matrices for the raw data and embedded data by Equation (2.24).



set at  $\epsilon = 0.2$  and a sequence of data sets are generated with number of points between 500 and 3500 with increment of 40. The value of the smoothing parameter  $p$  is fixed at 0.9. We see that the pairwise error ( $\Gamma$ ) decreases exponentially with  $R^2$ -value of 0.9713 when data density increases (Figure 2.15).

### 2.3.2 Computational complexity

The three steps of the algorithm, namely, clustering, smoothing, and embedding have different computational complexities. In particular, for  $T$  points with dimensionality  $d$ , the clustering algorithm has a complexity of

$$O(Td^2 + d^3) \tag{2.25}$$

which is dominated by the PCA [68, 67] step. This is a summation of costs encountered through the computation of the covariance matrix,  $O(Td^2)$ , and singular value decomposition in that matrix,  $O(d^3)$ .

The computational complexity of the smoothing algorithm is dominated by the Dijkstra's algorithm [42] for calculating the longest geodesics. Here, we first assume that each cluster with respect to the first reference point contains an equal number of points. Thus, partitioning a data set of  $T$  points into  $n_c^1$  clusters yields  $T/n_c^1$  points in a cluster. The complexity of generating the longest geodesic in each such cluster is  $O(T/n_c^1 \log(T/n_c^1))$  [42, 81], which sums-up  $n_c^1$  times and implies the total complexity  $O(T \log(T/n_c^1))$  of making all geodesics with respect to the first reference point. Similarly, for the second reference point, the same procedure is followed for all  $n_c^2$  clusters,

each containing  $T/n_c^2$  points, to obtain a complexity of  $O(T \log(T/n_c^2))$ . Altogether, geodesics from both reference points contributes a computational complexity of

$$O\left(T \log\left(\frac{T^2}{n_c^1 n_c^2}\right)\right) \quad (2.26)$$

for the smoothing algorithm.

In the embedding algorithm, discretizing splines and approximating the minimum distance are dominant in the total cost of that algorithm. Here, we first calculate the mean length of splines with respect to the first reference point, denoted as  $\tilde{s}^1 \in \mathbb{R}^d$ , and the one for the second reference point, denoted as  $\tilde{s}^2 \in \mathbb{R}^d$ . Then, the computational cost associated with approximating intersection of these two splines are computed. For a pair of  $d$ -dimensional splines, computation of the Euclidean distance between any two points, one from each spline, has complexity of  $3d$  from the operations of subtraction, square, and summation along all  $d$  dimensions. Since each spline  $\tilde{s}^{1,2}$  is discretized at a mesh size  $m_s$ , while  $\tilde{s}^1$  has  $\tilde{s}^1/m_s$  points on it,  $\tilde{s}^2$  has  $\tilde{s}^2/m_s$  points. The computational cost of approximating closest distance between any two points one from each spline  $\tilde{s}^{1,2}$  is  $3d\tilde{s}^1\tilde{s}^2/m_s^2$ . Here, for simplicity, we assume that the length of each spline with respect to the first and second reference points have same lengths as  $\tilde{s}^1$  and  $\tilde{s}^2$ , respectively. Thus, the total complexity of the embedding algorithm, which has  $n_c^1$  and  $n_c^2$  number of splines with respect to the first and second reference points, is

$$O\left(\frac{3d}{m_s^2} n_c^1 n_c^2 \tilde{s}^1 \tilde{s}^2\right). \quad (2.27)$$

Clustering, embedding algorithms, and steps 1–6 in the embedding algorithm are performed only once for a given data set, thus after they are completed,  $T'$  new points are embedded with an additional complexity of  $O(T')$ .

In this chapter, we presented a NDR method constructed using cubic smoothing splines in three steps clustering, smoothing, and embedding. The smoothing parameter assures the smoothness of the embedding even for noisy data sets. The examples emphasized the validity of the method for variety of data sets. The performance analysis presented here provides a good insight about the method.

Dimensionality reduction reveals an underlying manifold of a higher-dimensional dataset. However, a transition in collective motion, which is defined as singularities on the manifold, splits the underlying manifold into two sub-manifolds such that they are joined at the locus of singularities. The salient scenario is that the manifold possesses a significantly high curvature at the locus of singularities. Thus, in order to detect transitions in collective motion, high curvature on the manifold can be detected. Chapter 3 provides a detailed presentation of a framework to detect transitions using the manifold's curvature.

## Chapter 3

# Detecting Phase Transitions using Manifold's Curvature

An internal or external perturbation such as a food source or a predator attack in a multi-agent system results an abrupt change of the activity which is defined as a *phase transition* [12, 23, 31, 64, 90, 96, 137]. As an example, a predator attack on a flying flock of birds sometime causes landing [83, 102, 133]. As a distinct manifold underlies in each phase about the transition, the manifold representing whole system  $\mathcal{M}$ , can be represented as a union of two distinct manifolds  $\mathcal{M}^{1,2}$ ,

$$\mathcal{M} = \bigcup_{j=1,2} \mathcal{M}^j. \quad (3.1)$$

**Definition 3.0.3.** *Phase transition* [136]. *An abrupt change of an activity of a multi-agent system.*

We assert that a transition in a multi-agent system is represented as a *locus of*

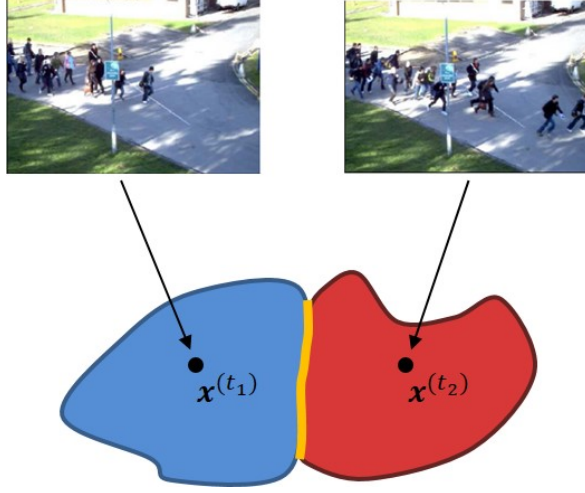


Figure 3.1: A cartoon of two manifolds (blue and red) and the locus of singularity (orange) of the human crowd example in Section 1.2. Snapshots sampled at time  $t_1$  and  $t_2$  representing the first and second phases of the motion embed as points  $x^{(t_1)}$  and  $x^{(t_2)}$  in the corresponding manifold, respectively.

*singularity*  $\mathcal{L}$ , on the manifold where the manifold is not differentiable. Geometrically, the intersection of these two manifolds represents the locus of the singularity as

$$\mathcal{L} = \bigcap_{j=1,2} \mathcal{M}^j \quad (3.2)$$

[52]. A cartoon representing two manifolds and the locus of singularity for the example in Section 1.2 is presented in Figure 3.1. The locus of singularity on the manifold technically represents the transition from walk to run in the video.

**Definition 3.0.4. Locus of singularity [52].** All  $\mathbf{x}' \in \mathcal{M}$  such that  $\left. \frac{d\mathcal{M}}{dx} \right|_{\mathbf{x}=\mathbf{x}'}$  is not defined.

Infinite curvature at the singularity is not observed often in empirical data as it mostly consists noise. However, we assert that the curvature of the underlying manifold

is significantly high at the locus of singularities on the manifold created by a phase transition in the empirical data derived from a multi-agent system. This assertion is justified by the following simple geometric example illustrating a high curvature on a manifold at the singularity. We use a three-dimensional joined-manifold that we call a ‘sombbrero-hat’ (Figure 3.2(a)) of 2000 points produced by the equations

$$\begin{aligned} y_1 &= x_1 \cos x_2 + \epsilon, \\ y_2 &= x_1 \sin x_2 + \epsilon, \\ y_3 &= \begin{cases} 4 - x_1^2 + \epsilon, & \text{if } x_1 \leq 2, \\ \epsilon, & \text{if } x_1 > 2, \end{cases} \end{aligned} \tag{3.3}$$

for  $x_1 \in \mathbb{U}[0, 4]$  and  $x_2 \in \mathbb{U}[0, 2\pi]$ . Here,  $\epsilon$  is a noise parameter sampled from the uniform distribution  $\mathbb{U}[-0.05, 0.05]$ . This noise parameter ensures the generality of the data such that it is sampled from an empirical process. We observe this sombrero-hat joins two sub-manifolds, brim (green) and crown (blue), along with the locus (red) representing the singularities. In reality, while two manifolds represent two distinct behaviors of the system, the locus represents the transition. Figure 3.2(b) showing a vertical section at  $x_2 = 0$  indicates an infinite curvature at the locus (two red points). Isomap residual plots (Figure 3.2(c)) reveal that the embedding dimensionality of the locus is three, while that of two sub-manifolds are two each. If the noise parameter is zero, we observe that the dimensionality of the locus is one in this example.

Dimensionality reduction algorithms running on a dataset having a transition find a non-existing single manifold instead of two manifolds which are joined together.

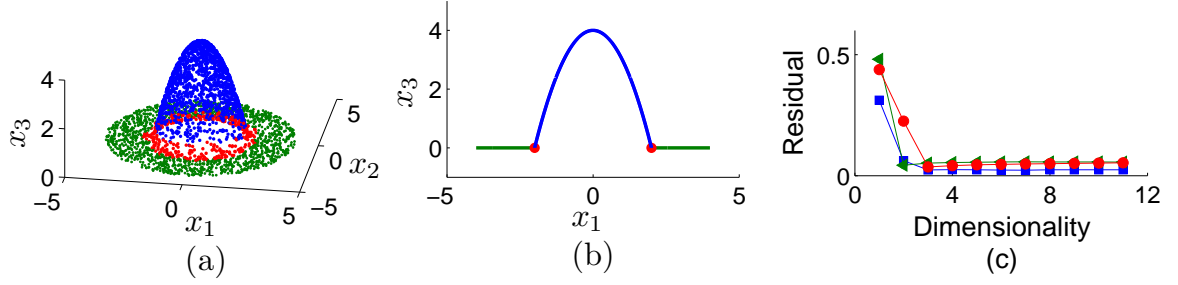


Figure 3.2: (a) A sombrero-hat of 2000 points having brim (green), crown (blue), and locus (red). (b) The vertical section at  $x_2 = 0$  of the manifold indicates a high curvature at red points. (c) Isomap residual plots showing embedding dimensionalities by elbows reveal that the dimensionalities of two sub-manifolds (blue and green) are two while the dimensionality of the locus (red) is three.

Thus, before utilizing DR algorithms existence of transitions should be checked and the data should be split if a transition exists. Thereby, here we present a framework to detect phase transitions via abrupt curvature changes on the manifold in the following section.

### 3.1 Method of detecting a phase transition

In this section, we analyze the geometry of a manifold's curvature based on points sampled on it and develop a framework to detect transitions. Therein, we first approximate the point-wise curvature of a curve in a two-dimensional space by using singular value ratios computed over the data. Then, we extend the assertion into higher dimensions by intersecting principal sections, made through the shape operator which we define in Section 3.1.2, with the manifold. Finally, we attest that the same transition can also be approximated locally by singular value ratios computed on the data sampled on the manifold. Each of these is elucidated with details in the following sections.

### 3.1.1 Approximating the curvature of a curve

Point-wise curvature of a curve is approximated by using singular value ratio computed over the data sampled in a neighborhood at the point as follows: We assume that the data is evenly distributed with density  $\rho$  along the curve. As in Figure 3.3, for any given point  $\mathbf{y}$  on the curve, we can superimpose a neighborhood in it with an arc  $\mathbf{y}_1\mathbf{y}_2$  of a translating circle such that the arc subtends a small angle  $2\Theta$  at the center,  $\mathbf{O}$ . We approximate the curvature of the circular sector  $\mathbf{y}_1\mathbf{y}_2$  (Figure 3.3(b)) in a translating circle (Figure 3.3(a)) instead of that of the curve itself since they are similar in aforesaid setup.

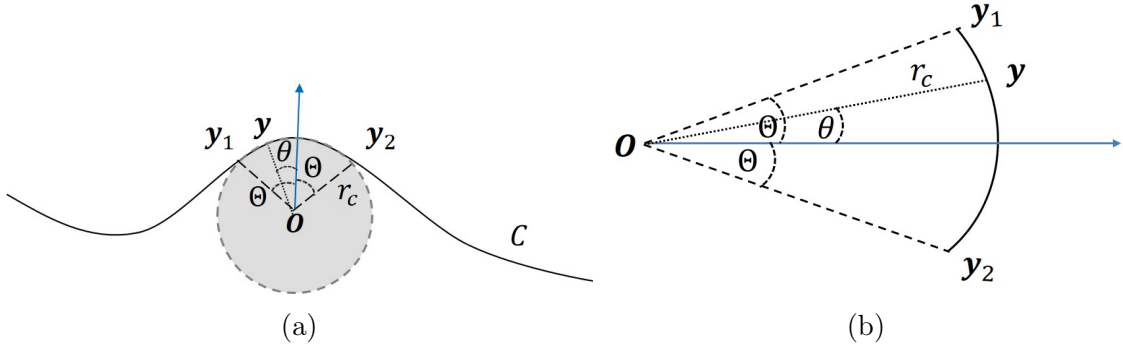


Figure 3.3: (a) Superimposing a neighborhood of the curve  $C$  at point  $\mathbf{y}$  with an arc  $\mathbf{y}_1\mathbf{y}_2$  which subtend an small angle of  $2\Theta$  at the origin  $\mathbf{O}$  of a translating circle having a radius  $r_c$ . (b) Circular sector  $\mathbf{y}_1\mathbf{y}_2$  which is rotated such that the blue arrow represents the horizontal axis of the coordinate system.

**Theorem 1.** *Given a circle with radius  $r_c$  centered at the origin, let  $\alpha$  points be uniformly distributed with density  $\rho$  on an arc which subtends an angle of  $2\Theta$  at the center. Let  $\sigma_1$  and  $\sigma_2$ , such that  $\sigma_1 > \sigma_2$ , are two singular values computed upon the data sampled on the arc  $\mathbf{y}_1\mathbf{y}_2$  (Figure 3.3(b)), then  $\sigma_2/\sigma_1 \approx K\kappa$  where  $\kappa = 1/r_c$ , and*



$$K = \frac{\alpha}{2\rho\sqrt{15}} \in \mathbb{R}^+.$$

*Proof.* A point  $\mathbf{y}(y_1, y_2)$  on the arc is given by  $y_1 = r_c \cos \theta$ ,  $y_2 = r_c \sin \theta$  for  $\theta \in [-\Theta, \Theta]$ . The expected values of variables  $y_1$  and  $y_2$  are computed as

$$\mu_{y_1} = \frac{1}{2\Theta} \int_{-\Theta}^{\Theta} r_c \cos \theta \, d\theta = \frac{1}{\Theta} r_c \sin \Theta, \quad (3.4)$$

$$\mu_{y_2} = \frac{1}{2\Theta} \int_{-\Theta}^{\Theta} r_c \sin \theta \, d\theta = 0, \quad (3.5)$$

respectively. We then compute pairwise *covariance* between variables as

$$\text{cov}(y_1, y_1) = \frac{1}{2T} \int_{-\Theta}^{\Theta} (r_c \cos \theta - \mu_{y_1})^2 d\theta = \frac{r_c^2}{2\Theta} \left( \Theta + \frac{1}{2} \sin 2\Theta + \frac{2}{\Theta} \sin^2 \Theta \right), \quad (3.6)$$

$$\text{cov}(y_1, y_2) = \frac{1}{2\Theta} \int_{-\Theta}^{\Theta} (r_c \cos \theta - \mu_{y_1})(r_c \sin \theta - \mu_{y_2}) d\theta = 0, \quad (3.7)$$

$$\text{cov}(y_2, y_1) = \frac{1}{2\Theta} \int_{-\Theta}^{\Theta} (r_c \sin \theta - \mu_{y_2})(r_c \cos \theta - \mu_{y_1}) d\theta = 0, \quad (3.8)$$

$$\text{cov}(y_2, y_2) = \frac{1}{2\Theta} \int_{-\Theta}^{\Theta} (r_c \sin \theta - \mu_{y_2})^2 d\theta = \frac{r_c^2}{4\Theta} (2\Theta - \sin 2\Theta). \quad (3.9)$$

The *covariance matrix*  $\Sigma_C$ , of the data on the arc is

$$\Sigma_C = \begin{pmatrix} \frac{r_c^2}{2\Theta} \left( \Theta + \frac{1}{2} \sin 2\Theta + \frac{2}{\Theta} \sin^2 \Theta \right) & 0 \\ 0 & \frac{r_c^2}{4\Theta} (2\Theta - \sin 2\Theta) \end{pmatrix}. \quad (3.10)$$

The covariance matrix which is approximated to 5<sup>th</sup> order of  $\Theta$  by using the Taylor's

expansion is

$$\tilde{\Sigma}_C = \begin{pmatrix} \frac{r_c^2 \Theta^4}{45} & 0 \\ 0 & \frac{r_c^2 \Theta^2}{3} \left(1 - \frac{\Theta^2}{5}\right) \end{pmatrix} + O(\Theta^6). \quad (3.11)$$

Let, eigenvalues of  $\tilde{\Sigma}_C$  are  $\lambda_1$  and  $\lambda_2$  such that  $\lambda_1 > \lambda_2$ , then

$$\lambda_1 \approx \frac{r_c^2 \Theta^2}{3} \left(1 - \frac{\Theta^2}{5}\right) \text{ and } \lambda_2 \approx \frac{r_c^2 \Theta^4}{45}. \quad (3.12)$$

As eigenvalues are computed upon the covariance matrix, they also relate to singular values, denoted by  $\sigma_1$  and  $\sigma_2$  where  $\sigma_1 > \sigma_2$  such that  $\sigma_2/\sigma_1 = \sqrt{\lambda_2/\lambda_1}$  [57]. Then,

$$\frac{\lambda_2}{\lambda_1} \approx \frac{\Theta^2}{3(5 - \Theta^2)} \implies \frac{\sigma_2}{\sigma_1} \approx \frac{\Theta}{\sqrt{3(5 - \Theta^2)}}. \quad (3.13)$$

Further,

$$\frac{\sigma_2}{\sigma_1} \approx \frac{\Theta}{\sqrt{15}} \quad (3.14)$$

since  $\Theta$  is small. Let,  $\alpha$  points are uniformly distributed with the density  $\rho$  on the arc, then,

$$\Theta = \frac{\alpha}{2\rho} \kappa, \text{ where } \kappa = \frac{1}{r_c}. \quad (3.15)$$

By Equations (3.14) and (3.15),

$$\frac{\sigma_2}{\sigma_1} \approx K\kappa \text{ where } K = \frac{\alpha}{2\rho\sqrt{15}} \in \mathbb{R}^+ \quad (3.16)$$

□

**Definition 3.1.1. Covariance [138].** The covariance of random variables  $y_1$  and  $y_2$

is defined as

$$\text{cov}(y_1, y_2) = E[(y_1 - \mu_{y_1})(y_2 - \mu_{y_2})] \quad (3.17)$$

where  $\mu_{y_1}$  and  $\mu_{y_2}$  are means of variables  $y_1$  and  $y_2$ , respectively.

**Definition 3.1.2. Covariance matrix [138].** For random variables  $\{y_1, y_2, \dots, y_m\}$ , the covariance matrix is

$$\begin{bmatrix} \text{cov}(y_1, y_1) & \text{cov}(y_1, y_2) & \dots & \text{cov}(y_1, y_m) \\ \text{cov}(y_2, y_1) & \text{cov}(y_2, y_2) & \dots & \text{cov}(y_2, y_m) \\ \dots & \dots & \dots & \dots \\ \text{cov}(y_m, y_1) & \text{cov}(y_m, y_2) & \dots & \text{cov}(y_m, y_m) \end{bmatrix}. \quad (3.18)$$

### 3.1.2 Approximating the curvature of a manifold

We extend the two dimensional assertion into higher dimensions by intersecting *principal sections*, made through the *shape operator* which is defined bellow, with the manifold. Curvatures of a manifold in *mutually orthogonal* tangential directions are measured using eigenvalues and eigenvectors of the shape operator [99, 112], such that eigenvalues provide magnitudes and eigenvectors provide directions of principal curvatures along tangential directions [3].

**Definition 3.1.3. Mutually orthogonal [78].** A set of vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$  are mutually orthogonal, if  $\mathbf{v}_i \cdot \mathbf{v}_j = 0$  for all  $i \neq j$ .

Let  $\mathcal{M}^m = (f_1(x_1, \dots, x_m), \dots, f_m(x_1, \dots, x_m))$  represents the parametric form of

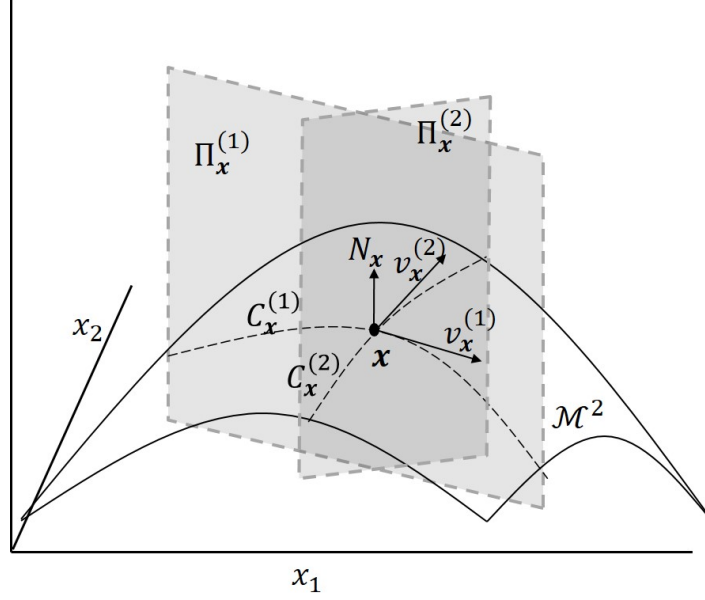


Figure 3.4: Computation of principal sections  $\Pi_x^{(1)}$  and  $\Pi_x^{(2)}$ , and principal directions  $\mathbf{v}_x^{(1)}$  and  $\mathbf{v}_x^{(2)}$ , of a two-dimensional manifold  $\mathcal{M}^2$  at the point  $\mathbf{x}$ . The principal sections are mutually orthogonal as they are constructed from a mutual orthogonal tangential basis.

the manifold embedded in the euclidean space  $\mathbb{R}^{m+1}$ . For  $j = 1, \dots, m$ ,

$$\mathbf{u}_x^{(j)} = \frac{\partial \mathcal{M}^m}{\partial x_j} \quad \text{and} \quad \hat{\mathbf{u}}_x^{(j)} = \frac{\mathbf{u}_x^{(j)}}{\|\mathbf{u}_x^{(j)}\|} \quad (3.19)$$

provide mutually orthogonal tangent and unit tangent vectors of  $\mathcal{M}^m$  at  $\mathbf{x}$ , respectively.

Thus,  $\{\hat{\mathbf{u}}_x^{(j)} \mid j = 1, \dots, m\}$  is a orthonormal basis for the tangential space at  $\mathbf{x}$ . The

shape operator  $\mathcal{S}_x$ , of the manifold  $\mathcal{M}^m$  at the point  $\mathbf{x}$  is defined as

$$\mathcal{S}_x = \left( -\nabla_{x_{j_1}} N_x \cdot \hat{\mathbf{u}}_x^{(j_2)} \right)_{j_1, j_2} \quad \text{for} \quad j_1, j_2 = 1, \dots, m, \quad (3.20)$$

where  $N_x$  is the unit normal at  $\mathbf{x}$  [99, 112].

Let  $(\kappa(\mathbf{v}_\mathbf{x}^{(j)}), \mathbf{v}_\mathbf{x}^{(j)})$  for  $j = 1, \dots, m$  denote eigenpairs of  $\mathcal{S}_\mathbf{x}$ , then magnitudes and directions of principal curvatures are given by  $\kappa(\mathbf{v}_\mathbf{x}^{(j)})$  and  $\mathbf{v}_\mathbf{x}^{(j)}$ , respectively [3]. The two-dimensional plane through  $\mathbf{x}$  in  $\mathbb{R}^{m+1}$  which is spanned by the principal direction,  $\mathbf{v}_\mathbf{x}^{(j)}$ , and  $N_\mathbf{x}$ , is defined as the  $j$ -th principal section

$$\Pi_\mathbf{x}^{(j)} = \left\{ \gamma_1 \mathbf{v}_\mathbf{x}^{(j)} + \gamma_2 N_\mathbf{x} \mid \gamma_1, \gamma_2 \in \mathbb{R} \right\}. \quad (3.21)$$

Thus, for  $j = 1, \dots, m$ ,  $\Pi_\mathbf{x}^{(j)}$  are mutually orthogonal. Intersection of  $\Pi_\mathbf{x}^{(j)}$ 's with the manifold makes curves  $C_\mathbf{x}^{(j)} \in \mathbb{R}^{m+1}$  such that each passing through  $\mathbf{x}$ . While Figure 3.4 illustrates this scenario when  $m = 2$ , an example of computing principal sections is presented bellow.

Without loss of generality, for  $j = 1, \dots, m$ , we assume that the data is distributed uniformly with sufficiently density on  $C_\mathbf{x}^{(j)}$  to contain  $\alpha$  points in a neighborhood at the point  $\mathbf{x}$ . For all  $j$ , singular values  $\sigma_1^{(j)}$  and  $\sigma_2^{(j)}$ , such that  $\sigma_1^{(j)} > \sigma_2^{(j)}$ , are computed up on the data sampled in the neighborhood of  $\mathbf{x}$  on  $C_\mathbf{x}^{(j)}$ . By theorem 1, curvature  $\kappa^{(j)}$  on the curve  $C_\mathbf{x}^{(j)}$  is approximated by  $\sigma_2^{(j)} / \sigma_1^{(j)} \approx K^{(j)} \kappa^{(j)}$  for some  $K^{(j)} \in \mathbb{R}^+$ . A transition differentiates the manifold into two sub-manifolds such that each maps with a different Euclidean space [80]. Thus, under a transition, the geometry permits that some curves  $C_\mathbf{x}^{(j)}$ 's, undergo abrupt changes in the curvature which also results abrupt changes of the ratio  $\sigma_2^{(j)} / \sigma_1^{(j)}$ .

### An example: computing principal curvatures using the shape operator

Here, we provide an example of computing principal curvatures and directions of a two dimensional manifold in  $\mathbb{R}^3$  (Figure 3.5) given in term of the parametric representation

$$\mathcal{M}^2 = (x_1, x_2, x_1^3 - 3x_1x_2^2) \in \mathbb{R}^3. \quad (3.22)$$

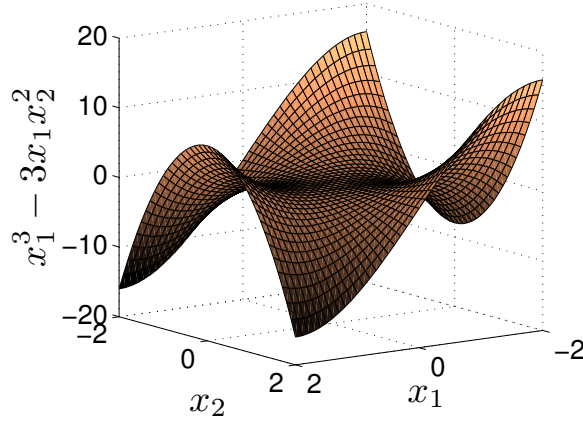


Figure 3.5: Two dimensional saddle surface  $\mathcal{M}^2$ , described by the Equation (3.22) for  $x_1, x_2 \in \mathbb{U}[-2, 2]$ .

For an arbitrary point  $\mathbf{x} = (x_1, x_2, x_1^3 - 3x_1x_2^2)$  on the manifold, we compute tangential and unit tangential vectors in orthogonal directions as,

$$\mathbf{u}_{\mathbf{x}}^{(1)} = \frac{\partial \mathcal{M}^2}{\partial x_1} = (1, 0, 3[x_1^2 - x_2^2]), \quad \mathbf{u}_{\mathbf{x}}^{(2)} = \frac{\partial \mathcal{M}^2}{\partial x_2} = (0, 1, -6x_1x_2), \quad (3.23)$$

and

$$\hat{\mathbf{u}}_{\mathbf{x}}^{(1)} = \frac{\mathbf{u}_{\mathbf{x}}^{(1)}}{\|\mathbf{u}_{\mathbf{x}}^{(1)}\|} = \frac{(1, 0, 3[x_1^2 - x_2^2])}{\sqrt{9(x_1^2 - x_2^2)^2 + 1}}, \quad \hat{\mathbf{u}}_{\mathbf{x}}^{(2)} = \frac{\mathbf{u}_{\mathbf{x}}^{(2)}}{\|\mathbf{u}_{\mathbf{x}}^{(2)}\|} = \frac{(0, 1, -6x_1x_2)}{\sqrt{36x_1^2x_2^2 + 1}} \quad (3.24)$$

respectively. Then, the unit normal at  $\mathbf{x}$   $N_{\mathbf{x}}$ , is

$$N_{\mathbf{x}} = \frac{\mathbf{u}_{\mathbf{x}}^{(1)} \times \mathbf{u}_{\mathbf{x}}^{(2)}}{\|\mathbf{u}_{\mathbf{x}}^{(1)} \times \mathbf{u}_{\mathbf{x}}^{(2)}\|} = \frac{(-3[x_1^2 - x_2^2], -6x_1x_2, 1)}{\sqrt{9(x_1^2 + x_2^2)^2 + 1}}, \quad (3.25)$$

while the tangential directions are

$$\nabla_{x_1} N_{\mathbf{x}} = \frac{\partial N}{\partial x_1} = \frac{-6(x_1[18x_2^2[x_1^2 + x_2^2] + 1], x_2[9[-x_1^4 + x_2^4] + 1], 3x_1[x_1^2 + x_2^2])}{[9(x_1^2 + x_2^2)^2 + 1]^{3/2}} \quad (3.26)$$

and

$$\nabla_{x_2} N_{\mathbf{x}} = \frac{\partial N}{\partial x_2} = \frac{6(x_2[18x_1^2[x_1^2 + x_2^2] + 1], -x_1[9[x_1^4 - x_2^4] + 1], -3x_2[x_1^2 + x_2^2])}{[9(x_1^2 + x_2^2)^2 + 1]^{3/2}}. \quad (3.27)$$

By Equations (3.23), (3.24), (3.26), and (3.27), we compute

$$\mathcal{S}_{\mathbf{x}}^{(1,1)} = -\nabla_{x_1} N_{\mathbf{x}} \cdot \hat{\mathbf{u}}_{\mathbf{x}}^{(1)} = \frac{-6x_1}{\sqrt{[9(x_1^2 + x_2^2)^2 + 1][9(x_1^2 - x_2^2)^2 + 1]}}, \quad (3.28)$$

$$\mathcal{S}_{\mathbf{x}}^{(1,2)} = -\nabla_{x_1} N_{\mathbf{x}} \cdot \hat{\mathbf{u}}_{\mathbf{x}}^{(2)} = \frac{-6x_2[9(x_1^2 + x_2^2)(x_2^2 - 3x_1^2) + 1]}{\sqrt{[9(x_1^2 + x_2^2)^2 + 1][9(x_1^2 - x_2^2)^2 + 1]}}, \quad (3.29)$$

$$\mathcal{S}_{\mathbf{x}}^{(2,1)} = -\nabla_{x_2} N_{\mathbf{x}} \cdot \hat{\mathbf{u}}_{\mathbf{x}}^{(1)} = \frac{6x_2[9(x_1^2 + x_2^2)^2 + 1]}{\sqrt{[9(x_1^2 + x_2^2)^2 + 1][9(x_1^2 - x_2^2)^2 + 1]}}, \quad (3.30)$$

$$\mathcal{S}_{\mathbf{x}}^{(2,2)} = -\nabla_{x_2} N_{\mathbf{x}} \cdot \hat{\mathbf{u}}_{\mathbf{x}}^{(2)} = \frac{6x_1[-9(x_1^2 + x_2^2)(x_1^2 - 3x_2^2) + 1]}{\sqrt{[9(x_1^2 + x_2^2)^2 + 1][9(x_1^2 - x_2^2)^2 + 1]}}. \quad (3.31)$$

Particularly, the shape operator  $\mathcal{S}_{\mathbf{x}} = \begin{pmatrix} \mathcal{S}_{\mathbf{x}}^{(1,1)} & \mathcal{S}_{\mathbf{x}}^{(1,2)} \\ \mathcal{S}_{\mathbf{x}}^{(2,1)} & \mathcal{S}_{\mathbf{x}}^{(2,2)} \end{pmatrix}$ , at  $\mathbf{x} = (1, 0, 1)$  is

$$\mathcal{S}_{(1,0,1)} = \begin{pmatrix} -3/50 & 0 \\ 0 & -12/25 \end{pmatrix}. \quad (3.32)$$

Eigenpairs  $(-3/50, (1, 0, 1))$  and  $(-12/25, (0, 1, 1))$  of  $\mathcal{S}_{(1,0,1)}$  describe two principal curvatures and directions in the tangential space at  $\mathbf{x} = (1, 0, 1)$ . Since the unit normal at  $\mathbf{x} = (1, 0, 1)$ , denoted as  $N_{(1,0,1)}$ , is  $\frac{1}{\sqrt{10}}(-3, 0, 1)$ , principal sections in  $\mathbb{R}^3$  at that point are given by

$$\begin{aligned} \Pi_{(1,0,1)}^{(1)} &= \{\gamma_1(1, 0, 1) + \gamma_2(-3, 0, 1) \mid \forall \gamma_1, \gamma_2 \in \mathbb{R}\}, \\ \Pi_{(1,0,1)}^{(2)} &= \{\gamma_1(0, 1, 1) + \gamma_2(-3, 0, 1) \mid \forall \gamma_1, \gamma_2 \in \mathbb{R}\}. \end{aligned} \quad (3.33)$$

### 3.1.3 Transition via local data distribution on the manifold

Constructing the shape operator at a neighborhood of each point on the manifold is computationally expensive, thus here we present an alternative approach to compute the singular value ratios and detect transitions. Therein, we first make a neighborhood  $\mathcal{N}_{\mathbf{x}} \subset \mathbb{R}^{m+1}$  at each point  $\mathbf{x}$  such that it contains  $\alpha \in \mathbb{N}$  points by running nearest neighbor search algorithm [51, 142]. Then, we perform *singular value decomposition* over the data in each  $\mathcal{N}_{\mathbf{x}}$  and denote the descending order by  $\sigma_1, \dots, \sigma_{\alpha}$  for some  $\alpha \in \mathbb{N}$ .



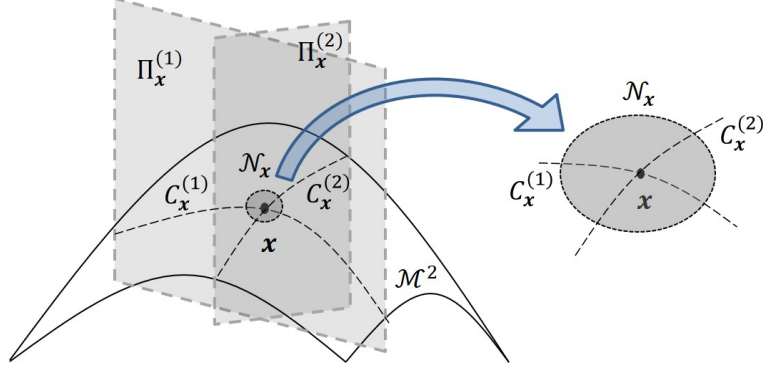


Figure 3.6: Local distribution of data in the neighborhood  $\mathcal{N}_x$  on a two dimensional manifold  $\mathcal{M}^2$ . Principal sections  $\Pi_x^{(1)}$  and  $\Pi_x^{(2)}$  are created by using the shape operator around  $\mathbf{x}$ , and curves  $C_x^{(1)}$  and  $C_x^{(2)}$  are produced by intersecting  $\Pi_x^{(1)}$  and  $\Pi_x^{(2)}$  with  $\mathcal{M}^2$ .

Without loss of generality, we assume that the data distribution in each  $C_x^{(j)}$  is dense enough for  $\mathcal{N}_x$  to contain data from each curve  $C_x^{(j)}$  for  $j = 1, \dots, m$ , made in Section 3.1.2. However,  $\mathcal{N}_x$  contains at least one point  $\mathbf{x}$ , from each  $C_x^{(j)}$  as shown in Figure 3.6 which depicts this scenario for a two dimensional manifold  $\mathcal{M}^2$ , in  $\mathbb{R}^3$ .

**Definition 3.1.4. Singular value decomposition [138].** Singular value decomposition finds two unitary matrices  $U_{n \times \alpha}$  and  $V_{m \times \alpha}$  such that  $U'_{\alpha \times n} U_{n \times \alpha} = V'_{\alpha \times m} V_{m \times \alpha} = I_{\alpha \times \alpha}$  those provide the decomposition  $\mathcal{D}_{n \times m} = U_{n \times \alpha} \Sigma_{\alpha \times \alpha} V'_{\alpha \times m}$  for  $\Sigma_{\alpha \times \alpha} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_\alpha)$  with  $\sigma_1 > \dots > \sigma_\alpha$ .

According to this setting, we can assert that the abrupt changes which are detected by the ratio  $\sigma_2^{(j)} / \sigma_1^{(j)}$  in some curves created through  $\mathbf{x}$  are also detected by the ratio  $\sigma_\alpha / \sigma_1$  computed over the data in  $\mathcal{N}_x$  on the manifold  $\mathcal{M}^m$ . For clarity, from here onward, the nearby points on the manifold are indexed with nearby values and the point  $\mathbf{x}$  on the manifold is denoted as the  $t$ -th point. Let consider  $T$  data points are

embedded on the manifold, thus the distribution

$$\{(\sigma_\alpha/\sigma_1)_i \mid t = 1, 2, \dots, T\} \quad (3.34)$$

provides the magnitudes of the *phase* at each point. A group of agents evolving with mutual interaction can be explained in terms of a hidden manifold structure such that each frame corresponds to a single data point on the manifold. A behavioral change, known as a *phase change*, between consecutive frames is measured by moving difference of singular value ratios computed over neighborhoods of corresponding frames as

$$\mathcal{T}_t = (\sigma_\alpha/\sigma_1)_{t+1} - (\sigma_\alpha/\sigma_1)_t \text{ for } t = 1, 2, \dots, T-1. \quad (3.35)$$

**Definition 3.1.5. *Phase change* [136].** A change (small) of the collective activity of a multi-agent system between two consecutive time steps (or frames).

Since we observe that the distribution  $\{\mathcal{T}_t \mid t = 1, \dots, T\}$  is highly volatile, we utilize  $\alpha$ -points moving sum

$$\Sigma_t^\alpha = \sum_{k \in [t - \lceil \frac{\alpha}{2} \rceil, t + \lceil \frac{\alpha}{2} \rceil] \cap \mathbb{N}} \mathcal{T}_k \quad ; \quad t \in \left[ \left\lceil \frac{\alpha}{2} \right\rceil, T - 1 - \left\lceil \frac{\alpha}{2} \right\rceil \right] \cap \mathbb{N}, \quad (3.36)$$

where  $\lceil \frac{\alpha}{2} \rceil$  is the smallest integer greater than or equal to  $\frac{\alpha}{2}$ , to smooth the distribution. The distribution  $\{\Sigma_t^\alpha \mid t = 1, \dots, T\}$ , for some  $T$ , provides the phase change of the multi-agent system from one time step to the other. After a transition occurs, manifold curvature is abruptly changed and then that abrupt change is also effected in singular values. Thus, phase transitions can be quantified by abrupt changes of the distribution

$$\{\Sigma_t^\alpha | t = 1, \dots, T\}.$$

### 3.2 Algorithm for detecting phase transitions

Here we present the algorithm of detecting phase transitions in multi-agent systems. This algorithm inputs data in terms of a data matrix  $\mathcal{D}_m$  and the parameter  $\alpha$  controlling number of nearest neighbors. As we use frames partitioned from a collective motion videos for data, here we explain the process of constructing the data matrix  $\mathcal{D}_m$  from frames.

We assume gray-color images are used for the study. Let the dimensionality of frames be  $m_1 \times m_2$ , then for all the time steps ( $t$ 's), we reshape each intensity matrix

$$\begin{bmatrix} y_1^{(t)} & y_{m_1+1}^{(t)} & \cdots & y_{(m_2-1)m_1+1}^{(t)} \\ y_2^{(t)} & y_{m_1+2}^{(t)} & \cdots & y_{(m_2-1)m_1+2}^{(t)} \\ \cdots & \cdots & \cdots & \cdots \\ y_{m_1}^{(t)} & y_{2m_1}^{(t)} & \cdots & y_{m_1m_2}^{(t)} \end{bmatrix}_{m_1 \times m_2},$$

of a frame into a column matrix

$$\left[ y_1^{(t)}, y_2^{(t)}, \dots, y_{m_1+1}^{(t)}, \dots, y_j^{(t)}, \dots, y_{m_1m_2}^{(t)} \right]^T$$

[18], where  $y_j^{(t)}$  is the intensity of the  $j$ -th pixel. Thus, the input data matrix is

$$\mathcal{D} = \left[ \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(t)}, \dots, \mathbf{y}^{(T)} \right] \ ; \ \mathbf{y}^{(t)} = \left[ y_1^{(t)}, \dots, y_j^{(t)}, \dots, y_{m_1m_2}^{(t)} \right]^T. \quad (3.37)$$

In this setup, each  $m_1 m_2$  dimensional vector  $\mathbf{y}^{(t)}$  is a data point on the manifold representing the collective behavior. Thus, the Euclidean distance  $d_{\mathcal{D}}(t_1, t_2)$ , between any two points,  $t_1$  and  $t_2$ , on the manifold is computed as

$$d_{\mathcal{D}}(t_1, t_2) = \|\mathcal{D}_m(t_1, :) - \mathcal{D}_m(t_2, :)\|; \quad t_1, t_2 = 1, \dots, T, \quad (3.38)$$

where  $T$  is the total number of frames in the data. The nearest neighbor search algorithm available in [142] computes nearest neighbors using Equation (3.38). Based on this distance, the  $\alpha$  nearest neighbors for the  $t$ -th point are extracted as  $\mathcal{N}_t$  ( same as  $\mathcal{N}_x$  in Section 3.1.3). Then, we compute  $\sum_t^\alpha$  from  $\mathcal{N}_t$  as explained in Section 3.1.3. The algorithm outputs the magnitudes of changes  $\Sigma_t^\alpha$ , so we can chose the largest among them as transitions. The method of detecting phase transitions is given as Algorithm 5.

---

**Algorithm 5** *Transition detection* : This detects transitions in collective motion using manifold's curvature. Algorithm inputs data in terms of a matrix  $\mathcal{D}$  and a parameter  $\alpha$  for number of nearest neighbors. The output observable  $\Sigma_t^\alpha$  ranks the magnitude of the phase change at each time step, thus the largest ones among them are used as phase transitions.

---

- procedure** *TRANSDETEC* ( $\alpha, \mathcal{D}$ )
- 1: Perform nearest neighbor search in [142](this computes distance using Equation (3.38)), and obtain  $\alpha$  nearest neighbors for each entry of  $\mathcal{D}$ . Denote data for  $\alpha$  nearest neighbors of the  $t$ -th point of  $\mathcal{D}_m$  by  $\mathcal{N}_t$ .
  - 2: Perform singular value decomposition on  $\mathcal{N}_t$  and denote the descending order of singular values by  $\sigma_1, \dots, \sigma_\alpha$  for some  $\alpha \in \mathbb{N}$ .
  - 3: Compute ratio  $\sigma_\alpha/\sigma_1$  for all  $\mathcal{N}_t$  and denote by  $(\sigma_\alpha/\sigma_1)_t$  where  $t = 1, \dots, T$ .
  - 4: Compute moving difference  $\mathcal{T}_t = |(\sigma_\alpha/\sigma_1)_{t+1} - (\sigma_\alpha/\sigma_1)_t|$  for  $t = 1, \dots, T-1$ .
  - 5: The  $\alpha$ -points moving sum  $\Sigma_t^\alpha$  of  $\mathcal{T}_t$  for  $t \in [\lceil \alpha/2 \rceil, T-1 - \lceil \alpha/2 \rceil] \cap \mathbb{N}$  is computed as,  $\Sigma_t^\alpha = \sum_{k \in [t - \lceil \frac{\alpha}{2} \rceil, t + \lceil \frac{\alpha}{2} \rceil] \cap \mathbb{N}} \mathcal{T}_k$ .
  - 6: Based on the distribution  $\Sigma_t^\alpha$ , phase transitions are extracted.
- end procedure**
-

### 3.3 Group behavioral examples

In this section, we evaluate the transition detection algorithm on four datasets: a synthetic dynamical simulation of the Vicsek model, and three natural data sets; a crowd of human, a flock of birds, and a school of fish.

#### 3.3.1 A simulation of the Vicsek model

A swarm of self propelled particles with three imposed transitions simulated by the Vicsek model [136] is examined to investigate the sensitivity of the algorithm. Since we simulate positions of particles in each frame, to avoid any ambiguity, notation for the set of nearest neighbors is moderated as  $\mathcal{N}_i^{(t)}$  to denote all the nearest-neighbors of the  $i$ -th agent within a unit distance at the  $t$ -th time step (each time step will be converted into a frame later).

The Vicsek model in [136] updates the orientation  $\theta_i^{(t)}$  of the  $i$ -th agent at the  $t$ -th time step as

$$\begin{aligned}\theta_i^{(t)} &= \arg\left(\phi_i^{(t-1)}\right) + \epsilon_i^{(t-1)}; \\ \phi_i^{(t-1)} &= \frac{1}{|\mathcal{N}_i^{(t-1)}|} \sum_{j \in \mathcal{N}_i^{(t-1)}} \begin{bmatrix} \cos\left(\theta_j^{(t-1)}\right) \\ \sin\left(\theta_j^{(t-1)}\right) \end{bmatrix},\end{aligned}\tag{3.39}$$

where  $\epsilon_i^{(t)}$  is the noise parameter sampled from a Gaussian distribution with mean zero and standard deviation  $\sigma$ . Here,  $\phi_i^{(t)}$  is the average direction of motion of all the agent in  $\mathcal{N}_i^{(t)}$ . The noise parameter plays an important role in this model as it determines

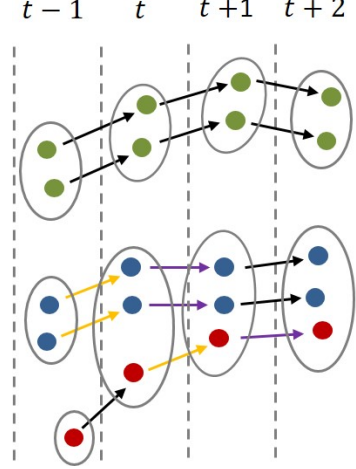


Figure 3.7: A particle swarm simulated from the Vicsek model that evolves at four consecutive time steps indicated by  $t - 1$ ,  $t$ ,  $t + 1$ , and  $t + 2$ . Gray color ellipses show neighbors. As the red particle joins with the blue particles at the  $t$ -th time step and becomes neighbors, the red particle's orientation at the current time step is determined as the average direction of the blue particles in the previous time step. This interaction of the orientations is shown by colored arrows.

the strength of the interaction of agents. A large noise level decreases the interaction and then collective behavior, while a low noise level causes the opposite.

The position  $\mathbf{y}_i^{(t)}$  of the  $i$ -th agent at the  $t$ -th time step is therefore updated as

$$\mathbf{y}_i^{(t)} = \mathbf{y}_i^{(t-1)} + w_i^{(t-1)} \begin{bmatrix} \cos(\theta_i^{(t-1)}) \\ \sin(\theta_i^{(t-1)}) \end{bmatrix} m_s, \quad (3.40)$$

where  $m_s$  is the time step size and  $w_i^{(t-1)}$  is the speed of the  $i$ -th agent at the time step  $(t-1)$ . In the Vicsek model, speed of a particle is independent of the interaction, but the orientation of a particle at the current time step is computed as the average direction of particles within a neighborhood in the previous time step. The basic mechanism of the Vicsek model is illustrated in Figure 3.7.

We run a simulation of this model and generate a synthetic data set of a particle

swarm of 50 agents in 200 time steps with periodic boundary conditions. We set  $m_s = 0.05$ ,  $w_i^{(t-1)} = 0.1$  for all  $i$  and  $t$ , and impose three transitions by varying the noise  $\epsilon_i^{(t)}$ , using a variable standard deviation

$$\sigma = \begin{cases} 0.25, & \text{if } t < 50 \\ 1, & \text{if } 51 \leq t < 100 \\ 0.05, & \text{if } 101 \leq t < 150 \\ 0.75, & \text{if } t \geq 150 \end{cases}. \quad (3.41)$$

for all  $i$  in the Gaussian distribution.

For  $t = 1, \dots, 200$ , we convert each set of point-mass positions of particles,  $\{\mathbf{y}_i^{(t)} | i = 1, \dots, 50\}$ , into a gray-color frame such that each particle is represented by a black square of pixel dimensionality of  $10 \times 10$ . Therein, we first convert the positions of particles into a sparse matrix,  $P^{(t)}$ , at each time step  $t = 1, \dots, 200$ . We make a rotationally symmetric *Gaussian low-pass filter*  $Q$  of size  $10 \times 10$  pixels with standard deviation of 10 pixels and filter the matrix  $P^{(t)}$  by  $Q$  to generate

$$R^{(t)}(l_1, l_2) = \sum_{k_1=1}^{10} \sum_{k_2=1}^{10} P^{(t)}(l_1 - k_1, l_2 - k_2) Q(k_1, k_2) ; \forall l_1, l_2 \quad (3.42)$$

for  $t = 1, \dots, 200$  [19]. Then, each matrix  $R^{(t)}$  is converted into a frame, and all the frames are finally converted into a one data matrix  $\mathcal{D}_m$ , as explained in Section 3.2.

**Definition 3.3.1.** *Gaussian low-pass filter* [61]. *Two dimensional Gaussian low*

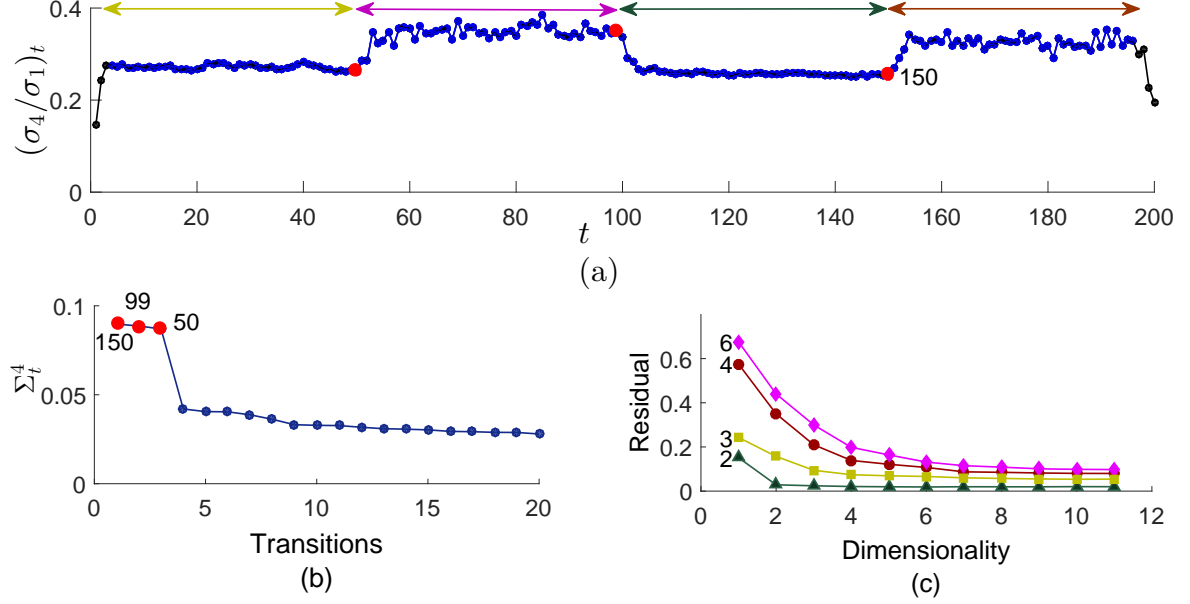


Figure 3.8: Detecting transitions in a particle swarm simulated using the Vicsek model with alternating noise levels. (a) Distribution of  $(\sigma_4/\sigma_1)_t$  versus frame numbers. Therein, the range of frames representing each sub-manifold is marked by a left-right arrow and the frame containing each transition is marked by a red circle along with the frame number associated. (b) Descending order of 20 largest transitions including frame numbers of three largest transitions. (c) Isomap residual variance versus dimensionality of each sub-manifold. Therein, while the numbers next to the plot gives the underlying dimensionalities, colors refer to the corresponding sub-manifolds in (a)

pass filter of size  $m_1 \times m_2$  is

$$Q(k_1, k_2) = \frac{1}{2\pi\sigma^2} e^{-\frac{k_1^2 + k_2^2}{2\sigma^2}},$$

where  $-\lceil m_1/2 \rceil \leq k_1 \leq \lceil m_1/2 \rceil$ ,  $-\lceil m_2/2 \rceil \leq k_2 \leq \lceil m_2/2 \rceil$ , and  $\sigma$  is standard deviation.

We run the transition detection algorithm upon frames with several  $\alpha$  values and observe that the distribution  $(\sigma_4/\sigma_1)_t$  generated by  $\alpha = 4$  (Figure 3.8(a)) shows clear transitions with respect to noise changes. The plot of descending order magnitudes of 20 largest transitions (Figure 3.8(b)) reveals that three transitions at frame numbers



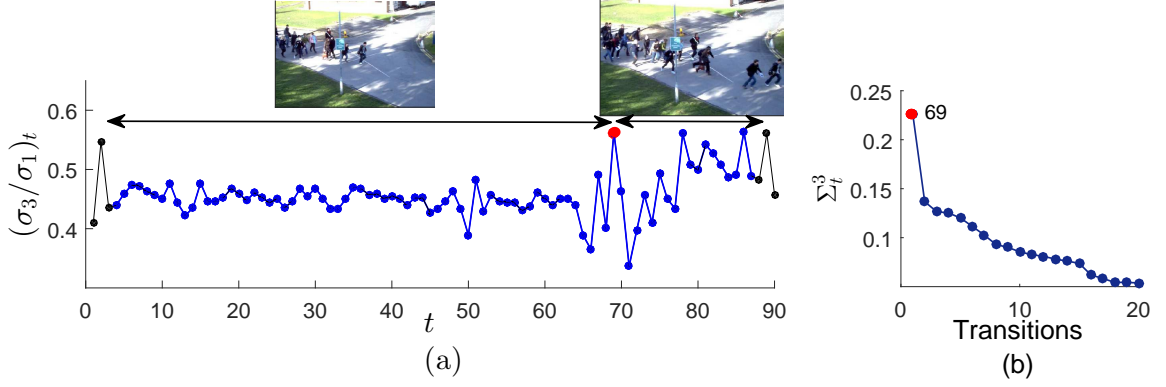


Figure 3.9: Detecting a transition between phases of walking and running in a crowd of human. (a) Distribution of  $(\sigma_3/\sigma_1)_t$  versus frame numbers. Therein, left-right arrows and the red circle represent ranges of frames in each sub-manifold and the frame at the transition, respectively, while the snapshots show instances in each phase. (b) Plot of descending order of 20 largest transitions representing the largest transition in red along with the frame number.

150, 99, and 50 are more significant than others. Thus, the manifold representing the whole motion is partitioned into transition free 4 sub-manifolds ranging 1–50, 51–99, 100–150, and 151–200. We run Isomap on each range of frames with the neighborhood parameter ( $\alpha$ ) four and obtain the residual plots given in Figure 3.8(c). The Isomap residual plots indicating the embedding dimensionality by an elbow, reveal that the sub-manifolds are embedded in three, six, two, and four dimensions, respectively.

### 3.3.2 A crowd of human

A video of a crowd of human available on-line in [48] which contains a transition at the 69-th frame from walk to run is considered next.

As explained in Section 3.2, we first convert all the frames into a data matrix  $\mathcal{D}_m$ , and then run transition detection algorithm on it. When  $\alpha = 3$ , we observe that the distribution  $(\sigma_3/\sigma_1)_t$  given in Figure 3.9(a) shows a clear trade-off about the phase

transition. Figure 3.9(b) shows only the first transition, having a magnitude of 0.23, is significant. The manifold representing the crowd is partitioned such that frames 1–69 represent one sub-manifold and frames 70–90 represent the other sub-manifold. The snapshots in Figure 3.9(a) show the phases walking (left) and running (right) of the crowd. Isomap running on frames 1–69 and 70–90 with  $\alpha = 3$  reveals that embedding dimensionalities of sub-manifolds are three and four, respectively.

### 3.3.3 A flock of birds

Now we detect a transition differentiating phases of sitting and flying at the 58-th frame in a video of a flock of bird obtained on-line in [100].

We formulate the data matrix  $\mathcal{D}_m$  as before and run transition detection algorithm on  $\mathcal{D}_m$  with several  $\alpha$  values. As we observe that  $\alpha = 3$  produces clear transitions, we use the distribution  $(\sigma_3/\sigma_1)_t$  given in Figure 3.10(a) for further study. Magnitudes of the descending order of 20 largest transitions in Figure 3.10(b) illustrates that one transition with magnitude of 0.32 is significant at the 58-th frame. Then, the manifold is partitioned into two sub-manifolds representing frames in the ranges 1 – 58 and 59 – 100. Isomap running on ranges of frames 1 – 58 and 59 – 100 with  $\alpha = 3$  reveals that the embedding dimensionalities of corresponding sub-manifolds are two and three.

### 3.3.4 A school of fish

Finally, we use a video of a school of fish having few transitions to validate the method. This school is stimulated with panics and the behavior is recorded.

We run the algorithm on the data matrix  $\mathcal{D}_m$ , with several  $\alpha$ 's and study the

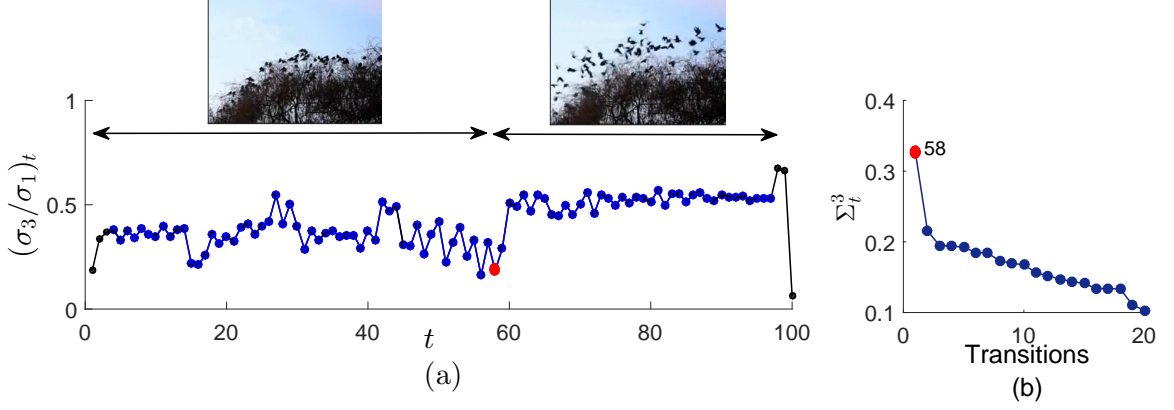


Figure 3.10: Detecting a transition in a bird flock between phases sitting and flying. (a) Distribution of  $(\sigma_3/\sigma_1)_t$  shows ranges of frames representing two sub-manifolds by left-right arrows and instances of phases by snapshots. (b) Descending order of 20 largest transitions. The frame at the largest transition is represented by red in Figures (a) and (b).

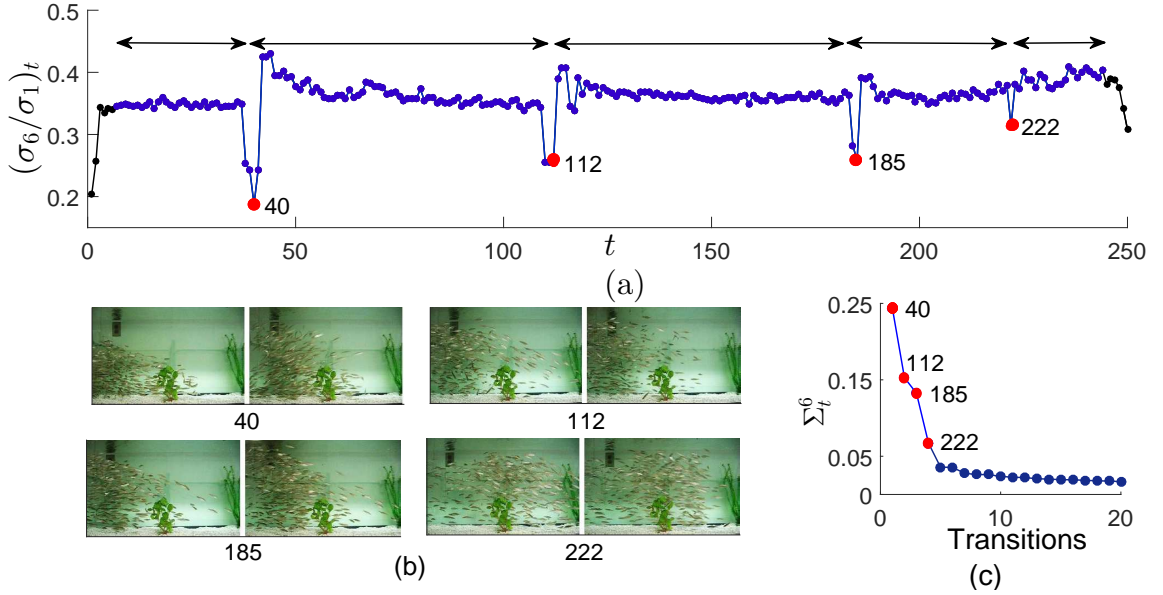


Figure 3.11: Detecting transitions in a school of fish. (a) Distribution of  $(\sigma_6/\sigma_1)_n$  with left-right arrows showing ranges of frames in sub-manifolds, and red dots showing frames at transitions. (b) Snapshots of the school before (left) and after (right) each transition. (c) Descending order of 20 largest transitions consisting four largest transitions marked in red with their frame numbers.

dynamics of the distribution  $(\sigma_6/\sigma_1)_t$  for  $\alpha = 6$  (Figure 3.11(a)). The plot of transitions (Figure 3.11(c)) reveals that the first four transitions located at frames 40, 112, 185, and 222 are significant. Now, the manifold is partitioned into transition free sub-manifolds representing frames in ranges 1–40, 41–112, 113–185, 186–222, and 223–250 as shown by left-right arrows in Figure 3.11(a). According to Figure 3.11(b), the pair of snapshots at the 40-th frame exhibits a large global abrupt change since the school reacts to the panic together. However, by the evidence observed through Figure 3.11(b) and snapshots, the second and the third transitions are abrupt and local while the last is gradual and local. Isomap is run on all sub-manifolds with  $\alpha = 6$  and embedding dimensionalities are obtained as two, four, three, two, and six, respectively.

In this chapter, we presented a framework of detecting phase transitions in collective motion by using the underlying manifold’s curvature. A transition in the collective motion can be explained as a singularity on the manifold. Two distinct manifolds hidden in two phases of the motion intersect each other at the locus of singularities having a high curvature. Thus, we assessed the high curvature on the manifold and therein detected transitions in the collective motion. The approach is equipped with a one dimensional relationship between curvature and singular value ratios, and then an extension of the one dimensional assertion to higher dimensions using the shape operator. The method was validated using one particle simulation and three different real-life examples.

Collective motion sometimes undergoes alternation of behaviors resulting the same alternation between representative underlying manifolds. Such alternation occurs due to changes of the speed, coordination, and structure of the agents. We will charac-

terize alternating manifolds using a metric assessing agents' speed, coordination, and structure. Therein we will compute their coordination using polarization and assess their structure using connected components. The detailed method of characterizing manifolds is presented in Chapter 4.

## Chapter 4

# Characterizing Alternating Manifolds

As we discussed in the Introduction (Chapter 1), the manifold representing the motion is switched to another manifold by an abrupt behavioral change (phase transition) of a multi-agent system. This phenomenon is sometimes extended towards an alternation between two manifolds due to alternation of the behavior of the system. In this chapter, we discuss a novel method of characterizing such alternation of manifolds.

We assume that agents move in a two-dimensional space, so that the raw data for the method consists of two-dimensional position vectors of all the agents in time. For a group of  $n$  agents, the position vector of the  $i$ -th agent at the  $t$ -th time step is denoted by  $\mathbf{y}_i^{(t)} \in \mathbb{R}^2$ . The configuration vector, defined as

$$\mathcal{Y}^{(t)} = [\mathbf{y}_1^{(t)}; \mathbf{y}_2^{(t)}; \dots; \mathbf{y}_n^{(t)}] \in \mathbb{R}^{2n}, \quad (4.1)$$

is a point on the manifold that represents the entire group at time  $t$ . The order of the agents in the configuration vectors is not required to be preserved for this method,

since the mapping scheme described in Section 4.1.1 determines the order of each agent in consecutive configurations.

The data acquired from the multi-agent group at distinct times  $t = 1, 2, \dots, T$  is consolidated in the dataset

$$\mathcal{D} = \{\mathbf{y}^{(t)} | t = 1, \dots, T\}. \quad (4.2)$$

If the multi-agent system alternates between  $m$  distinct behaviors, then the complete dataset  $\mathcal{D}$  of configurations can be partitioned into  $m$  distinct sub-manifolds such that

$$\mathcal{M} = \bigcup_{j=1, \dots, m} \mathcal{M}^j. \quad (4.3)$$

We can compute the dimensionality of each sub-manifold by implementing a DR algorithm such as Isomap [130] over the sample of configurations representing that sub-manifold. Our approach of characterizing the alternation of manifolds is based on a metric that we construct in Section 4.1.

## 4.1 Construction of the metric

In this section, we construct a metric that is susceptible of assessing similar manifolds created by the alternation of collective behavior. We hypothesize that the abrupt changes of speed, coordination, and structure of the agents create transitions and then aforesaid alternation in a multi-agent system. The metric consists of three components each measures changes that contribute to the system either by speed, coordination, or

structure.

As we explained in the beginning of this chapter, this method inputs only configurations. However, in future, we enhance this method such that it inputs frames partitioned from collective motion videos. For that, we extract their positions in frames and apply the method presented here to characterize alternating manifolds. In that purpose, we assume that the agents' order is not preserved between consecutive time steps. Thus, we define a simple mapping method to determine which agent is which between consecutive configurations. This special mapping is presented in Section 4.1.1, however, it can be bypassed when the agents are simulated from a deterministic process that preserves their order.

#### 4.1.1 Mapping of agents and computation of speed

Here, we first construct a bijective evolution function  $\Phi^{(t)}$ , for agents between time steps. This function arranges agents in configurations such that the same agent is represented by the same entry in each configuration. Then we compute displacement of agents between consecutive time steps and consider it as the speed since the unit of time is the step size.

We assume that the position of an agent at the next step is in a neighborhood of its position at the previous time step. For an agent, say  $\mathbf{y}_i^{(t)} \in \mathcal{Y}^{(t)}$ , we search the nearest neighbor [51], say  $\mathbf{y}_j^{(t+1)} \in \mathcal{Y}^{(t+1)}$ , and define the map  $\Phi_f^{(t)}$  as

$$\Phi_f^{(t)} : \mathcal{Y}^{(t)} \rightarrow \mathcal{Y}^{(t+1)} \text{ such that } \Phi_f^{(t)} \left( \mathbf{y}_i^{(t)} \right) := \mathbf{y}_j^{(t+1)}, \quad (4.4)$$



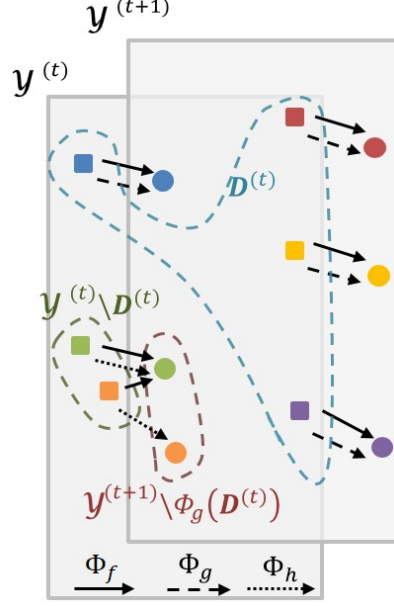


Figure 4.1: Mapping of positions of agents between two consecutive time steps. Same colored square and circle pair refers to the same agent. Squares represent agents' positions at the  $t$ -th time step and circles represent agents' positions at the  $(t + 1)$ -th time step.

see Figure 4.1.

Similarly, we apply  $\Phi_f^{(t)}$  to all the agents at the  $t$ -th time step to map with the corresponding agents at the  $(t + 1)$ -th time step. Since some agents at the  $t$ -th time step may share the same nearest neighbor at  $(t + 1)$ -th time step,  $\Phi_f^{(t)}$  maps more than one agents in  $\mathcal{Y}^{(t)}$  into the same agent in  $\mathcal{Y}^{(t+1)}$  and violate the uniqueness of the mapping. In order to ensure the uniqueness of the mapping,  $\Phi_f^{(t)}$  should be a *bijection* which is defined to be a one-to-one correspondence of agents between consecutive time steps. We define and extract the sub-domain  $\mathcal{D}^{(t)} \subseteq \mathcal{Y}^{(t)}$  as all agents those bijectively mapped with agents in  $\mathcal{Y}^{(t+1)}$  by  $\Phi_f^{(t)}$ .

**Definition 4.1.1. Domain [126].** The set  $X$  of the function  $f : X \rightarrow Y$ .

**Definition 4.1.2. Injective mapping [126].** A function  $f : X \rightarrow Y$  such that

$f(x_1) \neq f(x_2)$  whenever  $x_1 \neq x_2$  for  $x_1, x_2 \in X$ .

**Definition 4.1.3. Surjective mapping [126].** A function  $f : X \rightarrow Y$  such that each  $y \in Y$  there exist  $x \in X$  satisfying  $f(x) = y$ .

**Definition 4.1.4. Bijective mapping [126].** A function which is both injective and surjective.

The rest of the agents are mapped using a different rule that assures bijection between agents as described bellow. We denote the bijective mapping  $\Phi_f^{(t)}$  in  $\mathbf{D}^{(t)}$  by  $\Phi_g^{(t)}$

$$\Phi_g^{(t)} : \mathbf{D}^{(t)} \rightarrow \mathcal{Y}^{(t+1)}. \quad (4.5)$$

Mappings  $\Phi_f^{(t)}$  and  $\Phi_g^{(t)}$  and the domain  $\mathbf{D}^{(t)}$  for six agents are illustrated in Figure 4.1. For convenience and without loss of generality, we assume that the first  $n_1$  agents at the  $t$ -th time step are in  $\mathbf{D}^{(t)}$ . The velocity of the  $i$ -th agent in  $\mathbf{D}^{(t)}$ , say  $\mathbf{w}_i^{(t)} \in \mathbb{R}^2$  for  $i = 1, 2, \dots, n_1$ , is defined as the displacement from the current positions to the next position

$$\mathbf{w}_i^{(t)} = \Phi_g^{(t)}(\mathbf{y}_i^{(t)}) - \mathbf{y}_i^{(t)}. \quad (4.6)$$

We compute the mean velocity of the agents in  $\mathbf{D}^{(t)}$  as  $\boldsymbol{\mu}_1^{(t)} = \text{mean}\{\mathbf{w}_i^{(t)} ; i = 1, 2, \dots, n_1\}$  and use this quantity to map the remaining agents in  $\mathbf{D}^{(t)}$ .

Maps of the remaining agents in the configuration  $\mathbf{D}^{(t)}$ , denoted as  $\mathcal{Y}^{(t)} \setminus \mathbf{D}^{(t)}$ , with the remaining agents in  $\mathcal{Y}^{(t+1)}$ , denoted as  $\mathcal{Y}^{(t+1)} \setminus \Phi_g^{(t)}(\mathbf{D}^{(t)})$ , is presented below. We choose an agent  $\mathbf{y}_i^{(t)} \in \mathcal{Y}^{(t)} \setminus \mathbf{D}^{(t)}$  for  $i = n_1 + 1, \dots, n$  and compute displacement components from that agent to all the agents  $\mathbf{y}_j^{(t+1)} \in \mathcal{Y}^{(t+1)} \setminus \Phi_g^{(t)}(\mathbf{D}^{(t)})$ ;  $j = n_1 +$

$1, \dots, n$ . We use the mean velocity  $\boldsymbol{\mu}_1^{(t)}$ , to approximate the positions of unmapped agents in the  $t$ -th time step. If the displacement from  $\mathbf{y}_i^{(t)}$  to some  $\mathbf{y}_j^{(t+1)}$  is the closest to the mean velocity  $\boldsymbol{\mu}_1^{(t)}$ , that is

$$\|\mathbf{y}_i^{(t)} - \mathbf{y}_j^{(t+1)}\| \approx \|\boldsymbol{\mu}_1^{(t)}\| \quad \text{and} \quad \arg(\mathbf{y}_i^{(t)} - \mathbf{y}_j^{(t+1)}) \approx \arg(\boldsymbol{\mu}_1^{(t)}), \quad (4.7)$$

then  $\mathbf{y}_i^{(t)}$  is mapped to  $\mathbf{y}_j^{(t+1)}$  and define a new map  $\Phi_h^{(t)}$  as

$$\Phi_h^{(t)} : \mathcal{Y}^{(t)} \setminus \mathcal{D}^{(t)} \rightarrow \mathcal{Y}^{(t+1)} \setminus \Phi_g^{(t)}(\mathcal{D}^{(t)}) \text{ such that } \Phi_h^{(t)}(\mathbf{y}_i^{(t)}) := \mathbf{y}_j^{(t+1)}. \quad (4.8)$$

We apply this map for all the agents and map each  $\mathbf{y}_i^{(t)} \in \mathcal{Y}^{(t)} \setminus \mathcal{D}^{(t)}$  with  $\mathbf{y}_j^{(t+1)} \in \mathcal{Y}^{(t+1)} \setminus \Phi_g^{(t)}(\mathcal{D}^{(t)})$  for  $i, j = n_1 + 1, \dots, n$ . Figure 4.1 illustrates the mapping  $\Phi_h^{(t)}$ , the domain  $\mathcal{Y}^{(t)} \setminus \mathcal{D}^{(t)}$ , and the range  $\mathcal{Y}^{(t+1)} \setminus \Phi_g^{(t)}(\mathcal{D}^{(t)})$  for six agents. The velocity components  $\mathbf{w}_i^{(t)}$  for  $i = n_1 + 1, \dots, n$ , of agents in  $\mathcal{Y}^{(t)} \setminus \mathcal{D}^{(t)}$  are defined as the displacements of position

$$\mathbf{w}_i^{(t)} = \Phi_h^{(t)}(\mathbf{y}_i^{(t)}) - \mathbf{y}_i^{(t)}. \quad (4.9)$$

The set  $\mathbf{w}^{(t)} = \{\mathbf{w}_1^{(t)}, \mathbf{w}_2^{(t)}, \dots, \mathbf{w}_n^{(t)}\}$  represents all the individual velocity components at the  $t$ -th time step. The mapping  $\Phi^{(t)}$  in Equation (1.1) is defined as

$$\Phi^{(t)} = \begin{cases} \Phi_g^{(t)}, & \text{if the domain is } \mathcal{D}^{(t)}, \\ \Phi_h^{(t)}, & \text{if the domain is } \mathcal{Y}^{(t)} \setminus \mathcal{D}^{(t)}. \end{cases} \quad (4.10)$$

The mean velocity,  $\boldsymbol{\mu}_w^{(t)} = \text{mean}(\mathbf{w}^{(t)})$ , is considered to be the velocity of the whole

group at time  $t$ . The scaled mean speed

$$\frac{\|\boldsymbol{\mu}_w^{(t)}\|}{\max \left\{ \|\boldsymbol{\mu}_w^{(t)}\| \mid t = 1, \dots, T \right\}}, \quad (4.11)$$

is used to detect speed changes in the group.

As we mentioned before, the order of agents between configurations is not required to be preserved for this method, as the mapping presented above automatically finds agent's position between consecutive time steps. Given that this mapping has a rather modest computational cost of  $O(\log n)$ , we opt for this approach in favor of more accurate but computationally costly methods [69, 139].

### 4.1.2 Computation of coordination

Coordination of agents at the  $t$ -th time step is measured by the polarization [21],

$$\mathcal{P}^{(t)} = \frac{1}{n} \left\| \sum_{i=1}^n \begin{bmatrix} \cos(\theta_i^{(t)}) \\ \sin(\theta_i^{(t)}) \end{bmatrix} \right\|, \quad (4.12)$$

where  $\theta_i^{(t)} = \tan^{-1} \left( w_{2,i}^{(t)} / w_{1,i}^{(t)} \right)$  is the orientation of the  $i$ -th agent moving with velocity  $\mathbf{w}_i^{(t)} = [w_{1,i}^{(t)}, w_{2,i}^{(t)}]^T$  computed through the aforesaid mapping of agents.

We expect all the agents to have the same orientation of motion, if the agents in the group demonstrate a best coordinated behavior. Thus, Equation (4.12) adds similar sine and cosine values and outputs 1 as the polarization. However, the agents orienting into all different directions make the worst coordination. Thereby, Equation (4.12) sums arbitrary sine and cosine values which are between -1 and 1 and produces a 0 as

the polarization. Thus, the polarization ranges between 0 and 1, such that 1 identifies perfect coordination, while 0 implies no coordination.

### 4.1.3 Computation of structure

Change in structure is measured as the number of *connected components*  $C^{(t)}$ , of the interaction network of the agents at each time step, by representing the interaction network of agent positions as a time-varying graph. This network is made at each time step by linking the adjacent agents determined by the range search algorithm in [51]. For each agent, this algorithm searches all other agents within a given Euclidean distance

$$\epsilon = \frac{2}{n(n-1)T} \sum_{t=1}^T \sum_{i,j=1,\dots,n} \left\| \mathbf{y}_i^{(t)} - \mathbf{y}_j^{(t)} \right\|, \quad (4.13)$$

which is the average distance of agents over all time steps. The scaled number of connected components [129],  $C^{(t)}/n \in [0, 1]$ , is used to detect changes of the structure.

**Definition 4.1.5. Graph** [41]. A graph is an ordered pair  $G = (V, E)$  comprising a set  $V$  of vertices and a set  $E$  of edges such that elements of  $E$  are 2-element subsets of  $V$ .

**Definition 4.1.6. Path** [41]. A path is a non-empty graph  $P = (V, E)$  where  $V = \{v_0, v_1, \dots, v_k\}$  and  $E = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k\}$  such that all  $v_i$ 's are distinct.

**Definition 4.1.7. Connected component** [41]. A subgraph  $G' = (V', E')$  of  $G = (V, E)$  such that, for any two vertices  $v'_1, v'_2 \in V'$ , there exist a path between  $v'_1$  and  $v'_2$ , but all pairs of vertices  $v \in V$  and  $v' \in V'$ , no path between  $v$  and  $v'$ .

#### 4.1.4 The metric

We define the set  $\mathcal{X} = \{\mathcal{X}^{(t)}\}_{t=1}^T$ , which contains the values generated from a convex combination of the scaled speed, polarization, and number of connected components.

That is, for  $\gamma_1, \gamma_2 \in [0, 1]$  and  $\gamma_1 + \gamma_2 \leq 1$ , we let

$$\mathcal{X}^{(t)} = \gamma_1 \frac{\|\boldsymbol{\mu}_w^{(t)}\|}{\max \left\{ \|\boldsymbol{\mu}_w^{(t)}\| \mid t = 1, \dots, T \right\}} + \gamma_2 \mathcal{P}^{(t)} + (1 - \gamma_1 - \gamma_2) \frac{C^{(t)}}{n}, \quad (4.14)$$

so that the metric distance  $\Delta : \{1, \dots, T\} \times \{1, \dots, T\} \rightarrow \mathbb{R}$  is defined as

$$\Delta(t_1, t_2) = |\mathcal{X}^{(t_1)} - \mathcal{X}^{(t_2)}|. \quad (4.15)$$

Since  $\mathcal{X}^{(t)}$  is a *convex combination*, the relative importance of the speed, coordination and structure can be changed by varying the scalars  $\gamma_1$  and  $\gamma_2$  so that distinct group activities driven by changes in speed, coordination, and structure should be represented through distinct manifolds. In other words, two configurations,  $\mathcal{Y}^{(t_1)}$  and  $\mathcal{Y}^{(t_2)}$ , should give a large metric distance,  $\Delta(t_1, t_2)$ , if they are sampled from different manifolds representing different group activities.

**Definition 4.1.8. *Convex combination* [111].** A convex combination of a set  $\{X_1, \dots, X_m\}$  is

$$\gamma_1 X_1 + \dots + \gamma_m X_m = 0 \quad (4.16)$$

where real numbers  $\gamma_t$ 's satisfy  $\gamma_t \geq 0$  for all  $t$  and  $\gamma_1 + \dots + \gamma_m = 1$ .

## 4.2 Examples simulated from the Vicsek model

Here, we evaluate the metric on three simulations of self-propelled particles [136]. The self-propelled particle model [136] updates the position and orientation of each particle under the influence of its nearest-neighbors within a given distance. Briefly, for a nearest-neighbor set  $\mathcal{N}_i^{(t)}$ , comprising all agents within a unit distance from agent  $i$  (including agent  $i$  itself), the Vicsek model updates the orientation  $\theta_i^{(t)}$  of the  $i$ -th agent at the  $t$ -th time step as

$$\theta_i^{(t+1)} = \arg \left( \phi_i^{(t)} \right) + \epsilon_i^{(t)}, \quad (4.17)$$

where  $\epsilon_i^{(t)}$  is the orientation noise parameter for the  $i$ -th agent at the  $t$ -th time step and  $\phi_i^{(t)}$  is the average direction of motion of all nearest-neighbors including the agent. As explained in Section 3.3.1, the noise parameter in this model is important as it determines the strength of the interaction of agents. We assume that the noise is sampled from a uniform distribution between  $\alpha_1$  and  $\alpha_2$ ,  $\mathbb{U}[\alpha_1, \alpha_2]$ .

We augment the model by including a two-dimensional rotation matrix  $R_i^{(t)}$  for the  $i$ -th agent at the  $t$ -th time step. Such a rotation matrix can be used to change the orientation of select agents so that the group can split and rejoin. The position  $\mathbf{y}_i^{(t)}$  of the  $i$ -th agent at the  $t$ -th time step is therefore updated as

$$\mathbf{y}_i^{(t+1)} = \mathbf{y}_i^{(t)} + w_i^{(t)} R_i^{(t)} \begin{bmatrix} \cos(\theta_i^{(t)}) \\ \sin(\theta_i^{(t)}) \end{bmatrix} m_s, \quad (4.18)$$

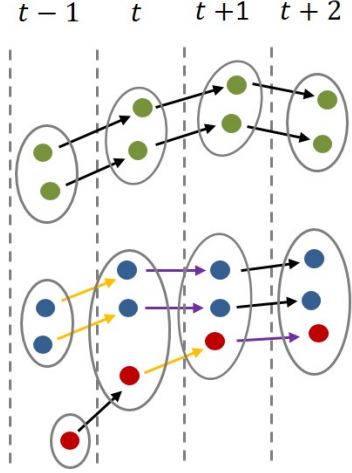


Figure 4.2: (Same as Figure 3.7) A particle swarm simulated from the Vicsek model that evolves at four consecutive time steps indicated by  $t - 1$ ,  $t$ ,  $t + 1$ , and  $t + 2$ . Gray color ellipses show neighbors. Since the red particle joins with the blue particles at the  $t$ -th time step and becomes neighbors, the red particle's orientation at the current time step is determined as the average direction of the blue particles in the previous time step. This similarity of the orientations is shown by yellow arrows.

$$\phi_i^{(t)} = \frac{1}{|N_i^{(t)}|} \sum_{j \in N_i^{(t)}} R_j^{(t)} \begin{bmatrix} \cos(\theta_j^{(t)}) \\ \sin(\theta_j^{(t)}) \end{bmatrix}, \quad (4.19)$$

where  $m_s$  is a duration of a time unit and  $w_i^{(t)}$  is the speed of the  $i$ -th agent at the  $t$ -th time step. The agent speed (assumed to be a constant in the original Vicsek model) is modeled as time varying here. If  $R_j^{(t)}$  is a  $2 \times 2$  identity matrix for all  $j$  and  $w_i^{(t)}$  is constant, the model in Equations (4.17), (4.18), and (4.19) specializes to the classical Vicsek model. The speed of a particle is independent from interactions, however, the orientation of a particle at the current time step is computed as the average direction of the neighbors at the previous time step using Equation (4.17) and (4.19). This scenario is shown in Figure 4.2 when  $R_i^{(t)} = I_{2 \times 2}$  for all  $t$  and  $i$  (the same illustration was given in Figure 3.7, however we copy it here for convenience).



In order to produce collective motion data, three simulations of the augmented Vicsek model are carried out by changing  $w_i^{(t)}$ ,  $\epsilon_i^{(t)}$ , and  $R_i^{(t)}$  for  $T$  time steps for  $n$  agents in a rectangular domain of size  $2L \times 2H$  with periodic boundary conditions [136]. Initial alignments of agents are set to zero, and positions are sampled from a uniform distribution in a circle with radius 2 units and centered at  $(-L + 2, 0)$ . The size of the time step  $m_s$  is taken to be 0.05.

The metric in Equation (4.15) is used to identify and characterize the distinct manifolds representing different group activities. For all the examples shown here, we set  $\gamma_1 = 1/3$  and  $\gamma_2 = 1/3$ , so that the speed, coordination, and structure are weighted equally. The pairwise metric distances are then used to identify separate group activities which are then further analyzed using Isomap.

#### 4.2.1 Switching speed

To study changes in collective motion governed by differences in group speed, we simulate 50 agents ( $n$ ) through 150 time steps ( $T$ ) in a domain with  $L = 8$  and  $H = 5$  and impose the following speed changes,

$$w_i^{(t)} = \begin{cases} 0.05 + \epsilon_s, & \text{if } t < 50 \\ 0.1 + \epsilon_s, & \text{if } 50 \leq t < 100, \\ 0.05 + \epsilon_s, & \text{if } t \geq 100 \end{cases} \quad (4.20)$$

where  $\epsilon_s \in \mathbb{U}[-0.01, 0.01]$  represents individual variability in speed with respect to the time. We set the orientation noise  $\epsilon_i^{(t)} \in \mathbb{U}[-0.01, 0.01]$  for all agents and choose the

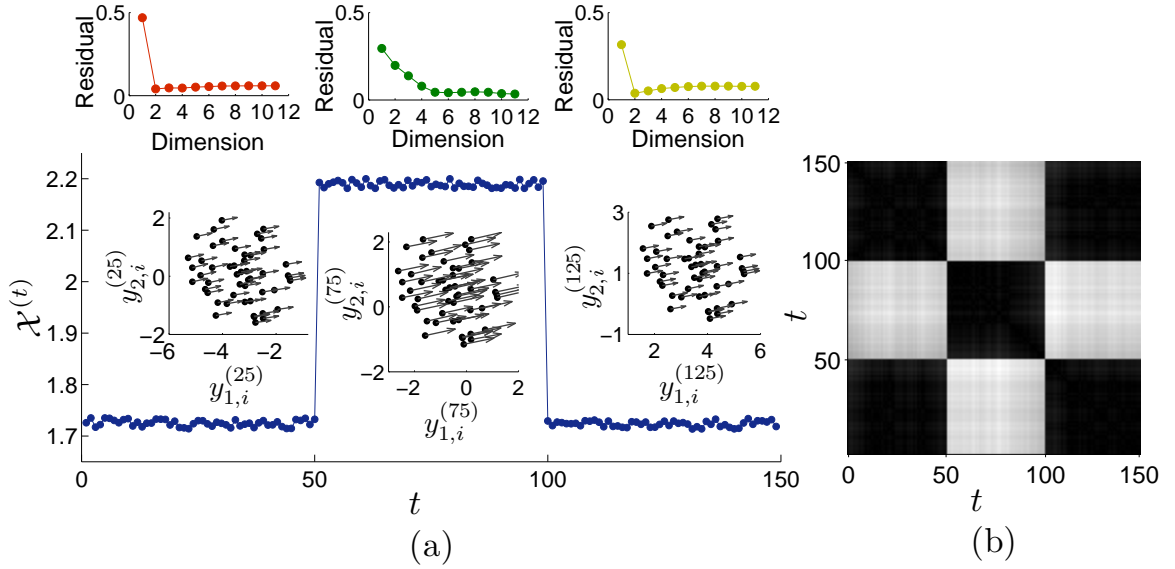


Figure 4.3: (a) Distribution of  $\mathcal{X}^{(t)}$  versus time. Snapshots in each section show that the structures and velocities of the agents. Therein, the length of the lines measure the speeds and the directions quantify the orientations of motion. Normalized Isomap residual variance versus dimensionality plots for each section are presented in the top row. For any two given configurations  $\mathbf{y}^{(t_1)}$  and  $\mathbf{y}^{(t_2)}$  where  $t_1, t_2 \in 1, \dots, T$ , and any arbitrary  $t_0 \in 1, \dots, T$ , if the distance metric image (b) shows the same color at two points  $(t_0, t_1)$  and  $(t_0, t_2)$  on the image, then those configurations lie on the same manifold, or on different manifolds otherwise.

rotation matrix to be the identity matrix for all agents.

Figure 4.3(a) shows that configurations for  $t$  between 51 and 100 are characterized by high values of  $\mathcal{X}^{(t)}$  compared to configurations through the time steps 51–100 and 101–150. This is evidenced from the right and left snapshots which display low speeds, while the middle snapshot that shows high speed. Residual variance plots (Figure 4.3(a) top row), obtained after running Isomap over the data, reveal that the first (red) and last (yellow) sections lie on an underlying manifold of dimensionality two. In contrast, Isomap indicates that the middle section (green) manifold has dimensionality five. Metric distances between all pairs of configurations are computed and rescaled to range between 0 and 1. Then, rescaled distances are converted to a gray color image,

presented as Figure 4.3(b), such that black refers to 0 and white to 1. According to the image, while configurations between 1–50 and 101–150 are embedded in the same manifold, those between 51–100 are embedded in a different manifold.

### 4.2.2 Switching coordination

To study changes elicited by differences in group coordination, we simulate a group of 50 agents through 150 time steps with  $L = 6$  and  $H = 6$  and consider the following noise levels for each agent in the three sections

$$\epsilon_i^{(t)} \in \begin{cases} 0.01\epsilon_c, & \text{if } t < 50 \\ 0.2\epsilon_c, & \text{if } 50 \leq t < 100, \\ 0.01\epsilon_c, & \text{if } t \geq 100 \end{cases} \quad (4.21)$$

where  $\epsilon_c \in \mathbb{U}[-1, 1]$ . The speed of agents is set to  $0.05 + \epsilon_s$  where  $\epsilon_s \in \mathbb{U}[-0.01, 0.01]$  and the rotation matrix is chosen to be the identity matrix.

Figure 4.4(a) demonstrates that configurations through time steps 51–100 are characterized by higher values of  $\mathcal{X}^{(t)}$  than those between time steps 1–50 and 101–150. Such a variation should be ascribed to the changes in the coordination of the group, as further evidenced from the snapshots presented therein. Specifically, while the right and left snapshots look similar, with agents moving with high coordination, the middle snapshot displays low coordination. The Isomap residual plot in Figure 4.4(b) confirms these similarities in terms of the dimensionality, in that the first (red) and last (yellow) sections lie on an underlying manifold of dimensionality three, while the second (green)

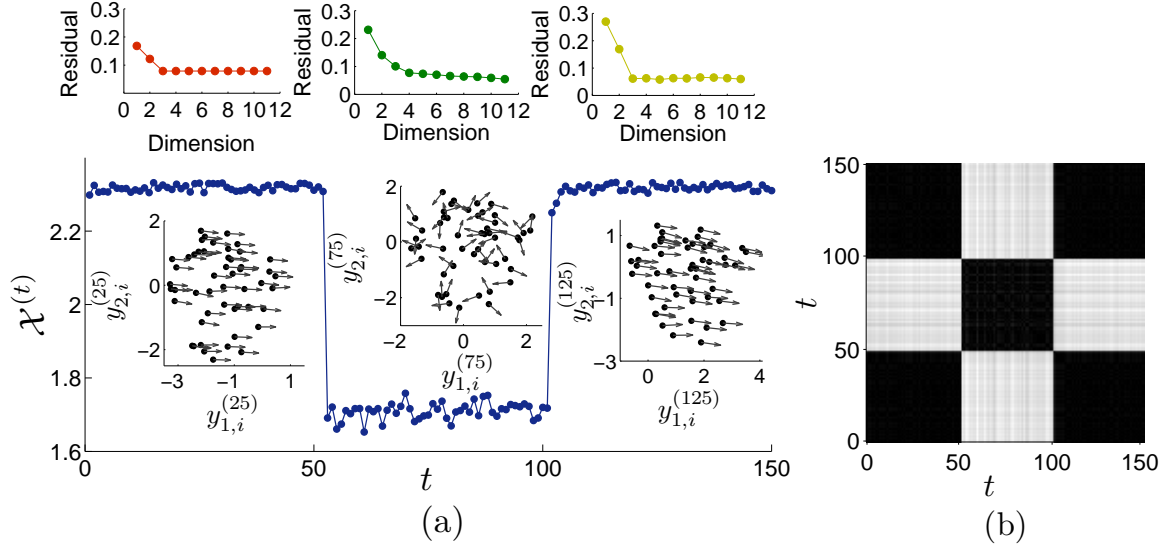


Figure 4.4: (a) Distribution of  $\mathcal{X}^{(t)}$  versus time with snapshots showing the structures along with velocities of the agents. Therein, the lengths measure their speeds and arrows quantify their orientations. Normalized Isomap residual variance plots for each section (red, green, yellow) are shown in the top row. For any two given configurations  $\mathcal{Y}^{(t_1)}$  and  $\mathcal{Y}^{(t_2)}$  where  $t_1, t_2 \in 1, \dots, T$ , and any arbitrary  $t_0 \in 1, \dots, T$ , if the distance metric image (b) gives the same color for pixels at two points  $(t_0, t_1)$  and  $(t_0, t_2)$  on the image, then those configurations are in the same manifold, or in different manifolds otherwise.

manifold has dimensionality four. As in the previous example, the metric distances between all pairs of configurations are computed and presented as an image in Figure 4.4(b). Empirical results presented therein indicate that configurations between time steps 1–50 and 101–150 embed in one manifold, while configurations between 51–100 embed in a different manifold.

### 4.2.3 Switching number of clusters

In this example, we simulate a group of agents that initially move together, then break into two subgroups, and eventually rejoin. We use a rotation matrix to change the orientation of agents such that the two subgroups move in different directions (Figure

4.5).

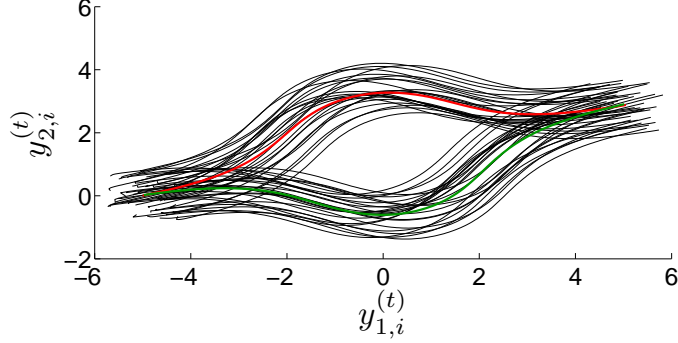


Figure 4.5: Trajectories of agents,  $\mathbf{y}_i^{(t)} = [y_{1,i}^{(t)}, y_{2,i}^{(t)}]^T$ , in two connected components (sub-groups), with red and blue curves depicting the trajectories of their centroids.

First, we compute two-dimensional coordinates  $\{\tilde{\mathbf{y}}^{(1)}, \dots, \tilde{\mathbf{y}}^{(T)}\}$  where  $\tilde{\mathbf{y}}^{(t)} = [\tilde{y}_1^{(t)}, \tilde{y}_2^{(t)}]^T \in \mathbb{R}^2$  to obtain the movement of the centroid of the upper subgroup as shown by red in the Figure 4.5. We discretize the horizontal axis into  $T$  segments ranges from  $-6$  to  $6$  by

$$\tilde{y}_1^{(t)} = 6(2t - T)/T ; t = 0, \dots, T. \quad (4.22)$$

Trajectory of the upper subgroup's centroid is obtained by using difference of two sigmoid function as

$$\tilde{y}_2^{(t)} = 5 \left( \frac{1}{1 + e^{-(\frac{12}{T}t-4)}} - \frac{1}{1 + e^{-(\frac{12}{T}t-8)}} \right) ; t = 0, \dots, T. \quad (4.23)$$

The rotation angle of the centroid of the upper subgroup between  $t$ -th and  $(t + 1)$ -th time steps is

$$\gamma^{(t)} = \tan \left( \left( \tilde{y}_2^{(t)} - \tilde{y}_2^{(t-1)} \right) / \left( \tilde{y}_1^{(t)} - \tilde{y}_1^{(t-1)} \right) \right) ; t = 2, \dots, T. \quad (4.24)$$

Then the rotation angles of the centroid of the lower subgroup is  $-\gamma^{(t)}$  for  $t = 2, \dots, T$ .

We arbitrarily split the whole group into two groups as

$$R_i^{(t)} = \begin{cases} \begin{pmatrix} \cos \gamma^{(t)} & -\sin \gamma^{(t)} \\ \sin \gamma^{(t)} & \cos \gamma^{(t)} \end{pmatrix}, & \text{if } 1 \leq i \leq \lceil \frac{n}{2} \rceil, \\ \begin{pmatrix} \cos(-\gamma^{(t)}) & -\sin(-\gamma^{(t)}) \\ \sin(-\gamma^{(t)}) & \cos(-\gamma^{(t)}) \end{pmatrix}, & \text{if } \lceil \frac{n}{2} \rceil + 1 \leq i \leq n, \end{cases} \quad (4.25)$$

where  $\lceil \frac{n}{2} \rceil$  is the smallest integer greater than  $\frac{n}{2}$ .

This special rotation matrix is used to simulate changes in group structure for 50 agents with  $L = H = 6$ . We select  $\epsilon_i^{(t)} \in \mathbb{U}[-0.01, 0.01]$  and  $w_i^{(t)} = 0.05 + \epsilon_s$  with  $\epsilon_s \in \mathbb{U}[-0.01, 0.01]$ .

Figure 4.6(a) reveals that configurations between 66–145 result in higher  $\mathcal{X}^{(t)}$  values than those in during time steps 1–65 and 146–220. The right and left snapshots illustrate that the particles move as one group, while the middle snapshot indicates the presence of two groups. Isomap (Figure 4.6(a)) implementation demonstrates that the first (red) and last (yellow) sections lie on an underlying manifold of dimensionality two, while the middle section's (green) manifold has dimensionality four. Metric distances presented as an image (Figure 4.6(b)) show that the configurations between 1–65 and 146–220 embed in one manifold and those between 66–145 in a different manifold.

Here we presented a framework to characterize alternating manifolds due to changing of behaviors of collective motion. We hypothesize that the cause for such alteration is change of speed, coordination, and structure of the multi-agent group. Thus, we de-

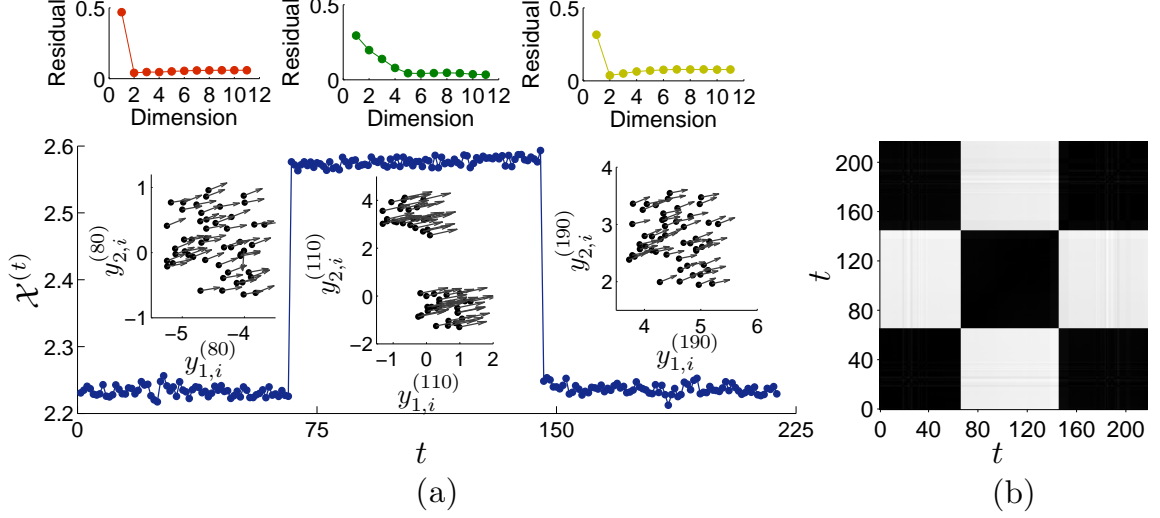


Figure 4.6: (a) Distribution of  $\mathcal{X}^{(t)}$  versus time with snapshots giving the structures and velocities. Therein, the length of lines measure their speeds and arrows quantify the their orientations. The top row of plots shows normalized Isomap residual variance versus dimensionality for each section (red, green, yellow). For any two given configurations  $\mathcal{Y}^{(t_1)}$  and  $\mathcal{Y}^{(t_2)}$  where  $t_1, t_2 \in 1, \dots, T$ , and any arbitrary  $t_0 \in 1, \dots, T$ , if the distance metric image (b) gives the same color for pixels at two points  $(t_0, t_1)$  and  $(t_0, t_2)$  on the image, then those configurations are in the same manifold, or in different manifolds otherwise.

velop a metric that is susceptible of assessing such alternations. Therein, we first introduce a bijective map that maps agents in consecutive time steps while preserving their collective motion and extract speed of agent from that. We assess the changes due to coordination using polarization and changes due to structure using number of connected components.

The synchronization is an important practice for foraging animals to self defense against predators and attain food sources. However, the synchronization requires to obey democracy and change their ideal behaviors into a one that most agents like. This encounters a cost into a multi-agent system and reduces the overall efficiency. The efficiency can be reestablished, if the multi-agent group is segregated into subgroups

such that agents are homogeneous within groups. However, further segregation creates small groups which are then vulnerable to predation. Thus, the efficient segregation is determined by optimizing a cost function assessing their hunger, fatigue, and predation risk. The detailed method is presented in Chapter 5.



## Chapter 5

# An Efficient Segregation of a Herd of Cows

The content presented in this chapter is a different branch in the thesis than that of Chapters 2–4 which are more about analyzing empirical observations of data. Here, we entirely rely on a modeling perspective to understand group motion of bovine behavior. Specifically, we model an efficient segregation of a herd of free range cows (*Bos taurus*) using a *cost function* assessing their *hunger*, *fatigue*, and *predation risk* [55]. This work has been a collaboration with Mason A. Porter and Marian S. Dawkins at University of Oxford.

Animal groups benefit from synchronization in many ways, however it requires them to switch their desired activity in a communal time rather than in the ideal time. The low efficiency encounters through the synchronization can be reestablished by segregating the group into subgroups such that activities are homogeneous within groups. However, further segregation creates small subgroups which are then vulnerable

to predation. Thus, here we assess the cost of a herd of cows as the sum of three quantities: variation of hunger, variation of fatigue, and predation risk. An efficient segregation is determined by minimizing this Cost Function (CF).

We use the temporal *Evolution Scheme* (ES) in [128] to simulate cows' desire to eat and desire to lie down, and change of states between eating, resting, and standing (Figure 1.7). The predation risk is modeled as an inverse exponential function of group size. Hunger, fatigue, and predation risk are then used in the CF that we construct here to quantify the total cost in the herd.

## 5.1 Temporal evolution and change of states of cows

The ES that interacts multiple cows provides a useful tool to understand their cooperative activities. However, here we modify that model to an iterative scheme as it can be easily combined with our CF. Moreover, without loss of generality, we reword three phrases in [128]: “lying down” as *resting*; “desire to eat” as *hunger*; “desire to lie down” as *fatigue*. In the model, the  $i$ -th cow in the  $t$ -th time step is described by three variables:  $\psi_i^{(t)}$  is for the state of the cow — either eating ( $\varepsilon$ ), resting ( $\mathcal{R}$ ), or standing ( $s$ );  $x_i^{(t)} \in [0, 1]$  represents the hunger; and  $y_i^{(t)} \in [0, 1]$  represents the fatigue.

For  $t = 1, \dots, T - 1$  and  $i = 1, \dots, n$ , the time-dependent coupling of cows is given by the differential equations

$$\begin{aligned} \dot{x}_i^{(t+1)} &= \left[ \alpha_i \left( \theta_i^{(t)} \right) + \frac{\sigma_x}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\varepsilon} \left( \theta_j^{(t)} \right) \right] x_i^{(t)}, \\ \dot{y}_i^{(t+1)} &= \left[ \beta_i \left( \theta_i^{(t)} \right) + \frac{\sigma_y}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\mathcal{R}} \left( \theta_j^{(t)} \right) \right] y_i^{(t)}, \end{aligned} \tag{5.1}$$

where

$$\begin{aligned}\alpha_i \left( \theta_i^{(t)} \right) &= -\xi_i'' \chi_\varepsilon \left( \theta_i^{(t)} \right) + \xi_i' \chi_\kappa \left( \theta_i^{(t)} \right) + \xi_i' \chi_s \left( \theta_i^{(t)} \right), \\ \beta_i \left( \theta_i^{(t)} \right) &= \zeta_i' \chi_\varepsilon \left( \theta_i^{(t)} \right) - \zeta_i'' \chi_\kappa \left( \theta_i^{(t)} \right) + \zeta_i' \chi_s \left( \theta_i^{(t)} \right),\end{aligned}\tag{5.2}$$

with

$$\chi_\psi \left( \theta_i^{(t)} \right) = \begin{cases} 1, & \theta_i^{(t)} = \psi, \\ 0, & \text{otherwise.} \end{cases}\tag{5.3}$$

Here,

$$\begin{aligned}\xi_i' &: \text{rate of increase of hunger,} \\ \xi_i'' &: \text{rate of decay of hunger,} \\ \zeta_i' &: \text{rate of increase of fatigue,} \\ \zeta_i'' &: \text{rate of decay of fatigue,}\end{aligned}\tag{5.4}$$

of the  $i$ -th cow for  $i = 1, \dots, n$ . We use the vector notations  $\boldsymbol{\xi}' = [\xi_1', \dots, \xi_n']^T$ ,  $\boldsymbol{\xi}'' = [\xi_1'', \dots, \xi_n'']^T$ ,  $\boldsymbol{\zeta}' = [\zeta_1', \dots, \zeta_n']^T$ , and  $\boldsymbol{\zeta}'' = [\zeta_1'', \dots, \zeta_n'']^T$ . The time-dependent adjacency matrix  $A^{(t)} = [a_{ij}^{(t)}]_{n \times n}$ , represents the network of cows at time  $t$ , so,  $d_i^{(t)} = \sum_{j=1}^n a_{ij}$  is the degree of the node  $i$ . Non-negative parameters,  $\sigma_x$  and  $\sigma_y$ , are coupling strengths with respect to hunger and fatigue, respectively.

The switching condition of the state variable  $\theta_i^{(t)}$  of the  $i$ -th cow at the  $t$ -th time

step is given by

$$\theta_i^{(t+1)} \rightarrow \begin{cases} \mathcal{E}, & \text{if } \theta_i^{(t)} \in \{\mathcal{R}, \mathcal{S}\} \text{ and } x_i^{(t)} = 1, \delta \leq y_i^{(t)} \leq 1, \\ \mathcal{R}, & \text{if } \theta_i^{(t)} \in \{\mathcal{E}, \mathcal{S}\} \text{ and } \delta \leq x_i^{(t)} < 1, y_i^{(t)} = 1, \\ \mathcal{S}, & \text{if } \theta_i^{(t)} \in \{\mathcal{E}, \mathcal{R}\} \text{ and } x_i^{(t)} = \delta, \delta \leq y_i^{(t)} < 1 \\ & (\text{or } \delta < x_i^{(t)} < 1, y_i^{(t)} = \delta). \end{cases} \quad (5.5)$$

We construct a Poincaré section for the switching condition in Equation (5.5) and present in Figure 5.1. For any time  $t$  and any agent  $i$ , the boundaries of this Poincaré section are

$$\begin{aligned} \partial\mathcal{E} &= \{(x_i^{(t)}, y_i^{(t)}, \theta_i^{(t)}) | x_i^{(t)} = 1, \delta \leq y_i^{(t)} \leq 1, \theta_i^{(t)} = \mathcal{E}\}, \\ \partial\mathcal{R} &= \{(x_i^{(t)}, y_i^{(t)}, \theta_i^{(t)}) | \delta \leq x_i^{(t)} < 1, y_i^{(t)} = 1, \theta_i^{(t)} = \mathcal{R}\}, \\ \partial\mathcal{S}_x &= \{(x_i^{(t)}, y_i^{(t)}, \theta_i^{(t)}) | \delta < x_i^{(t)} < 1, y_i^{(t)} = \delta, \theta_i^{(t)} = \mathcal{S}\}. \\ \partial\mathcal{S}_y &= \{(x_i^{(t)}, y_i^{(t)}, \theta_i^{(t)}) | x_i^{(t)} = \delta, \delta \leq y_i^{(t)} < 1, \theta_i^{(t)} = \mathcal{S}\}, \end{aligned} \quad (5.6)$$

We solve the dynamical system in Equation (5.1) for  $n$  cows in  $T$  time steps together with the switching condition in Equation (5.5). We present the corresponding iterative

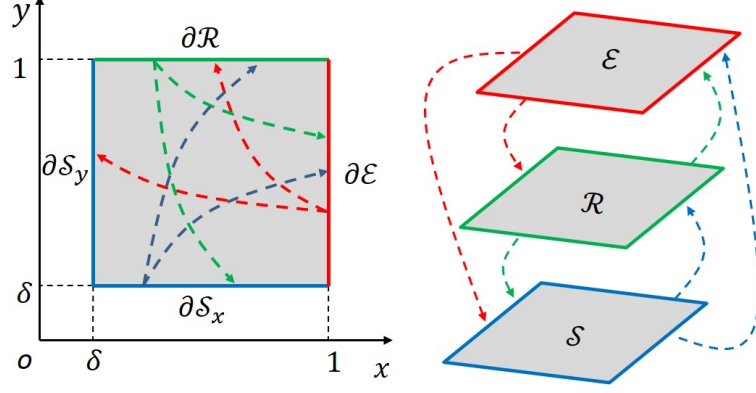


Figure 5.1: Switching condition of an arbitrary cow [128]. Edges of the square in the left panel represent boundaries given by Equation (5.6). Right panel shows three states  $\{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$  where  $\mathcal{S} = \mathcal{S}_x \cup \mathcal{S}_y$ , and the potential switching between states by arrows. Same color of arrows between panels refers to the same dynamics.

scheme in Table 5.1. For the simplicity therein, we represent

$$\begin{aligned}
 \eta'_i &= \xi'_i + \frac{\sigma_x}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\mathcal{E}}^{(t)} \left( \theta_j^{(t)} \right), \\
 \eta''_i &= -\xi''_i + \frac{\sigma_x}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\mathcal{E}}^{(t)} \left( \theta_j^{(t)} \right), \\
 \gamma'_i &= \zeta'_i + \frac{\sigma_y}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\mathcal{R}}^{(t)} \left( \theta_j^{(t)} \right), \\
 \gamma''_i &= -\zeta''_i + \frac{\sigma_y}{d_i^{(t)}} \sum_{j=1}^n a_{ij}^{(t)} \chi_{\mathcal{R}}^{(t)} \left( \theta_j^{(t)} \right),
 \end{aligned} \tag{5.7}$$

where  $i = 1, \dots, n$  and  $t = 1, \dots, T$ . Figure 5.1 illustrates the switching between cases stated in the table.

## 5.2 Cost module determining the segregation of herds

In this section, we construct the CF that we minimize later to determine the efficient grouping which gives the total number of groups and cows' occupancy in each group.

Table 5.1: Iterative scheme for the temporal evolution given by Equations (5.1), (5.2), (5.3), and (5.5). The Poincaré section describing the switching between cases is presented in Figure 5.2.

---

---

Case 1:	If $x_i^{(t)} = 1$ , $\delta \leq y_i^{(t)} \leq 1$ and $\theta_i^{(t)} = \varepsilon$ .
	Case a: State change form eating to resting.
	if $y_i^{(t)} \geq \delta^{\frac{\gamma'_i}{\eta''_i}}$ , then $x_i^{(t+1)} = \left[y_i^{(t)}\right]^{\frac{\eta''_i}{\gamma'_i}}$ , $y_i^{(t+1)} = 1$ and $\theta_i^{(t+1)} = \mathcal{R}$ .
	Case b: State change form eating to standing.
	if $y_i^{(t)} < \delta^{\frac{\gamma'_i}{\eta''_i}}$ , then $x_i^{(t+1)} = \delta$ , $y_i^{(t+1)} = \delta^{-\frac{\gamma'_i}{\eta''_i}} y_i^{(t)}$ , and $\theta_i^{(t+1)} = \mathcal{S}$ .

---

Case 2:	If $\delta \leq x_i^{(t)} < 1$ , $y_i^{(t)} = 1$ and $\theta_i^{(t)} = \mathcal{R}$ .
	Case a: State change form resting to eating.
	if $x_i^{(t)} \geq \delta^{\frac{\eta'_i}{\gamma''_i}}$ , then $x_i^{(t+1)} = 1$ , $y_i^{(t+1)} = \left[x_i^{(t)}\right]^{\frac{\gamma''_i}{\eta'_i}}$ , and $\theta_i^{(t+1)} = \varepsilon$ .
	Case b: State change form resting to standing.
	if $x_i^{(t)} < \delta^{\frac{\eta'_i}{\gamma''_i}}$ , then $x_i^{(t+1)} = \delta^{-\frac{\eta'_i}{\gamma''_i}} x_i^{(t)}$ , $y_i^{(t+1)} = \delta$ , and $\theta_i^{(t+1)} = \mathcal{S}$ .

---

Case 3:	If $x_i^{(t)} = \delta$ , $\delta \leq y_i^{(t)} \leq 1$ and $\theta_i^{(t)} = \mathcal{S}$ .
	Case a: State change form standing to eating.
	if $y_i^{(t)} \leq \delta^{\frac{\gamma'_i}{\eta'_i}}$ , then $x_i^{(t+1)} = 1$ , $y_i^{(t+1)} = \delta^{-\frac{\gamma'_i}{\eta'_i}} y_i^{(t)}$ , and $\theta_i^{(t+1)} = \varepsilon$ .
	Case b: State change form standing to resting.
	if $y_i^{(t)} > \delta^{\frac{\gamma'_i}{\eta'_i}}$ , then $x_i^{(t+1)} = \left[y_i^{(t)}\right]^{-\frac{\eta'_i}{\gamma'_i}} \delta$ , $y_i^{(t+1)} = 1$ , and $\theta_i^{(t+1)} = \mathcal{R}$ .

---

Case 4:	If $\delta < x_i^{(t)} < 1$ , $y_i^{(t)} = \delta$ and $\theta_i^{(t)} = \mathcal{S}$ .
	Case a: State change form standing to eating.
	if $x_i^{(t)} \geq \delta^{\frac{\eta'_i}{\gamma'_i}}$ , then $x_i^{(t+1)} = 1$ , $y_i^{(t+1)} = \left[x_i^{(t)}\right]^{-\frac{\gamma'_i}{\eta'_i}} \delta$ , and $\theta_i^{(t+1)} = \varepsilon$ .
	Case b: State change form standing to resting.
	if $x_i^{(t)} < \delta^{\frac{\eta'_i}{\gamma'_i}}$ , then $x_i^{(t+1)} = \delta^{-\frac{\eta'_i}{\gamma'_i}} x_i^{(t)}$ , $y_i^{(t+1)} = 1$ , and $\theta_i^{(t+1)} = \mathcal{R}$ .

---

---

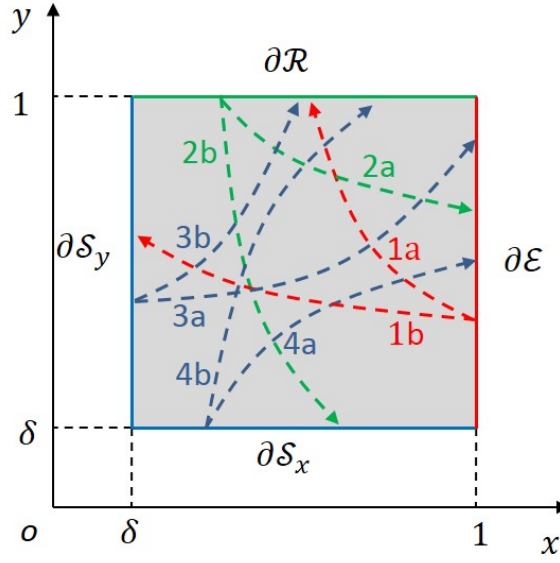


Figure 5.2: Switching of states [128],  $\{\mathcal{E}, \mathcal{R}, \mathcal{S}\}$  where  $\mathcal{S} = \mathcal{S}_x \cup \mathcal{S}_y$ , corresponding to Table 5.1. Here, an example, “1a” refers to the “case a” of the “case 1” in the table.

For some arbitrary  $L$ , suppose that the herd is allowed to segregates into a maximum of  $L$  distinct groups, represented by sets  $N_1^{(t)}, \dots, N_L^{(t)}$ , such that  $|N_l^{(t)}| = n_l^{(t)}$  for  $l = 1, \dots, L$  at each time step  $t \in T$ . If the herd segregates into  $L_1 < L$  groups at some time step  $t_1$ , then for computational purposes, we set  $|N_l^{(t_1)}| = 0$  for  $L_1 < l \leq L$ . We construct a CF based on the following assumptions:

- (i) Herds are fully democratic<sup>1</sup>.
- (ii) Cows are free to switch between groups, so that the groups are freely formed or dissolved [2, 105, 25].
- (iii) Fission and fusion of groups depend only on cows’ hunger, fatigue, and risk.
- (iv) The predation risk of a group is inversely exponential versus the group size.

---

<sup>1</sup>Decisions are taken based on majority’s choice

The CF is formulated in two modules, namely synchronization and risk. While the Synchronization Module (SM) assesses the cost incurred by the variation of fatigue and hunger of cows, the Risk Module (RM) assesses the predation risk.

### 5.2.1 Synchronization Module

Here we use cows' hunger  $x_i^{(t)} \in [0, 1]$  and fatigue  $y_i^{(t)} \in [0, 1]$  simulated in Section 5.1. Without loss of generality, we re-index the variable  $x_i^{(t)}$  as  $x_{k,l}^{(t)}$  and  $y_i^{(t)}$  as  $y_{k,l}^{(t)}$  to denote the hunger and fatigue of  $k$ -th cow in  $l$ -th group at  $t$ -th time step, respectively. As hunger and fatigue are two independent perceptions in cows, we separately make the groups homogeneous to either hunger or fatigue. Thus, we compute the two best groupings independently such that cows are homogeneous with respect to hunger in case I, and then to fatigue in case II. We then select the best set of groups among all possible groups in both cases that gives the lowest cost.

**Case I:** We sort cows such that the hunger is increasing, and first  $n_1^{(t)}$  cows are chosen into  $N_1^{(t)}$ , next  $n_2^{(t)}$  cows into  $N_2^{(t)}$ , and so forth. The cost incurred by hunger is the same as the heterogeneity of hunger within groups. Thus, we use the mean of the standard deviations of cows' hunger in groups,

$$h_1^{(t)} = \frac{1}{L} \sum_{l=1}^L \sqrt{\frac{\sum_{k=1}^{n_l^{(t)}} \left( x_{k,l}^{(t)} - \sum_{k=1}^{n_l^{(t)}} x_{k,l}^{(t)} / n_l^{(t)} \right)^2}{n_l^{(t)}}}, \quad (5.8)$$

to assess the cost carried by hunger. Similarly, we quantify the heterogeneity of groups with respect to fatigue as the mean of the standard deviations of cows' fatigue in groups



by

$$f_1^{(t)} = \frac{1}{L} \sum_{l=1}^L \sqrt{\frac{\sum_{k=1}^{n_l^{(t)}} \left( y_{k,l}^{(t)} - \sum_{k=1}^{n_l^{(t)}} y_{k,l}^{(t)} / n_l \right)^2}{n_l^{(t)}}}. \quad (5.9)$$

**Case II:** Similar to case I, we sort cows such that the fatigue increases and choose them into groups  $N_1^{(t)}, \dots, N_L^{(t)}$ . Here we also compute hunger by Equation (5.8) and fatigue by Equation (5.9), and we denote them as  $h_2^{(t)}$  and  $f_2^{(t)}$ , respectively.

### 5.2.2 Risk Module

Unlike hunger and fatigue, a herd's predation risk is independent of individuals' states and only depends on the group size. Because the group size and the risk of being attacked by predators is inversely related [33], we model the risk incurred by the  $l$ -th group,  $r_l^{(t)} \in (0, 1]$ , having the size  $n_l^{(t)}$  to be an inversely exponential function of group size,

$$r_l^{(t)} = e^{-(1-n_l^{(t)})/c}, \quad (5.10)$$

where  $c$  is a constant.

In Figure 5.3(a), we show the risk function for a group with the safety level  $\tau$  and safe size  $n_s$ . We assume that when a group consists of a sufficient number of cows, the predation risk is negligible and use this condition to compute the constant  $c$ . We name sufficient group size as *safe size* and denote by  $n_s$ , and we name the small risk at this size as *safety level* and denote by  $\tau$ . This implies that the constant  $c$  is  $-(1 - n_s) / \ln \tau$

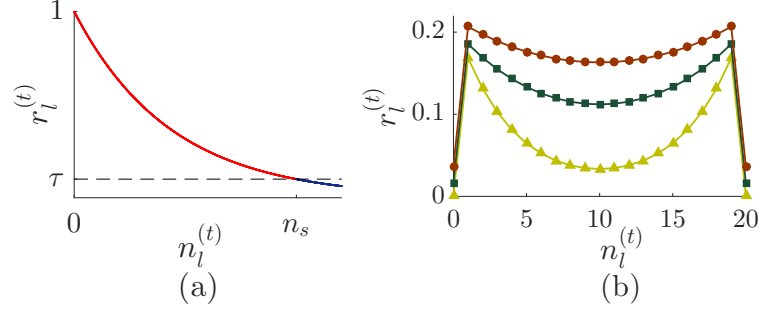


Figure 5.3: (a) Predation risk as an inverse exponential function of group size. The red curve segment shows significant risk and blue shows safe. (b) Risk of segregating a herd of 20 cows ( $n$ ) into two groups of sizes  $n_1^{(t)}$  and  $n_2^{(t)} = n - n_1^{(t)}$ , with  $\tau = 1/30$  and  $n_s = 10$  (yellow triangles), 20 (green squares), and 30 (brown circles).

and that the risk of the group is

$$r_l^{(t)} = \tau \left( \frac{1 - n_l^{(t)}}{1 - n_s} \right). \quad (5.11)$$

Figure 5.3(b) showing predation risk of a herd under different safe sizes indicates that large safe size contributes more risk to the herd independently of how they segregate. For all safe sizes, the least cost is achieved when the herd remains intact (i.e., no segregation), because larger group sizes entails safer herd. The second least cost is achieved when the herd segregates into equally-sized subgroups.

We compute the risk incurred by each group and we treat its mean,

$$r^{(t)} = \frac{1}{L} \sum_{l=1}^L \tau \left( \frac{1 - n_l^{(t)}}{1 - n_s} \right), \quad (5.12)$$

as the risk of the herd at time  $t$ .

In real-life situations, the safe size and safety level depend on the environment in which the herd lives. If the environment is either dense with predators or vulnerable

to predation, then the safe size should be comparatively large to achieve a significant safety level.

### 5.2.3 Cost function

We formulate two CFs, one with respect to the grouping when the hunger is homogeneous in groups and the other when the fatigue is homogeneous. Each cost function is formulated as a convex combination of costs incurred by hunger, fatigue, and risk. That is,

$$\begin{aligned} C_1^{(t)}(n_1^{(t)}, \dots, n_L^{(t)}) &= \gamma_1 h_1^{(t)} + \gamma_2 f_1^{(t)} + (1 - \gamma_1 - \gamma_2) r^{(t)}, \\ C_2^{(t)}(n_1^{(t)}, \dots, n_L^{(t)}) &= \gamma_1 h_2^{(t)} + \gamma_2 f_2^{(t)} + (1 - \gamma_1 - \gamma_2) r^{(t)}, \end{aligned} \tag{5.13}$$

where  $\gamma_1, \gamma_2 \in [0, 1]$  and  $\gamma_1 + \gamma_2 \leq 1$ .

### 5.2.4 Optimization

Here we use an exhaustive optimization method to minimize the cost functions. That is, for a given herd, denoted by the set  $N$ , along with known maximum number of groups, denoted by  $L$ , we minimize each CF over all plausible sets of groups that can be created from  $N$ . Let the grouping  $N_{1,1}^{(t)}, \dots, N_{1,L}^{(t)}$  provides the minimum cost of  $C_{1,m}^{(t)}$  in the first CF and the grouping  $N_{2,1}^{(t)}, \dots, N_{2,L}^{(t)}$  provides the minimum cost of  $C_{2,m}^{(t)}$  in the second CF. The best grouping ensuring the efficient segregation at the time step  $t$  is determine as the candidate grouping attaining the minimum of  $\{C_{1,m}^{(t)}, C_{2,m}^{(t)}\}$ .

### 5.3 Cost module and temporal evolution

Now we execute the CF simultaneously with the temporal ES for  $t = 1, \dots, T$ . At each time step we update the adjacency matrix  $A^{(t)} = [a_{i,j}^{(t)}]_{n \times n}$  in the scheme such that it agrees with the best grouping provided by the optimization of the CF in the previous time step. That is,

$$a_{i,j}^{(t+1)} = \begin{cases} 1, & \text{if } i, j \in N_l^{(t)} \text{ for any } l \in 1, \dots, L, \\ 0, & \text{otherwise.} \end{cases} \quad (5.14)$$

At each time step, optimization of the CF outputs the most efficient grouping until it reaches the stopping criterion, the maximum time steps ( $T$ ). The process diagram of the algorithm is shown in Figure 5.4.

### 5.4 Example

Now we carry out a detailed study of the CF to validate the method using two examples: general segregation — a herd segregates into three groups; adults-calves segregation — a herd containing half adults and half calves segregates into two groups.

#### 5.4.1 General segregation

We investigate performance of the method using an example of a herd of 12 cows ( $n$ ) in 30 time steps ( $T$ ), which allows them to segregate into maximum three groups ( $L$ ). We simulate hunger and fatigue using the ES followed by computing the CF and optimizing

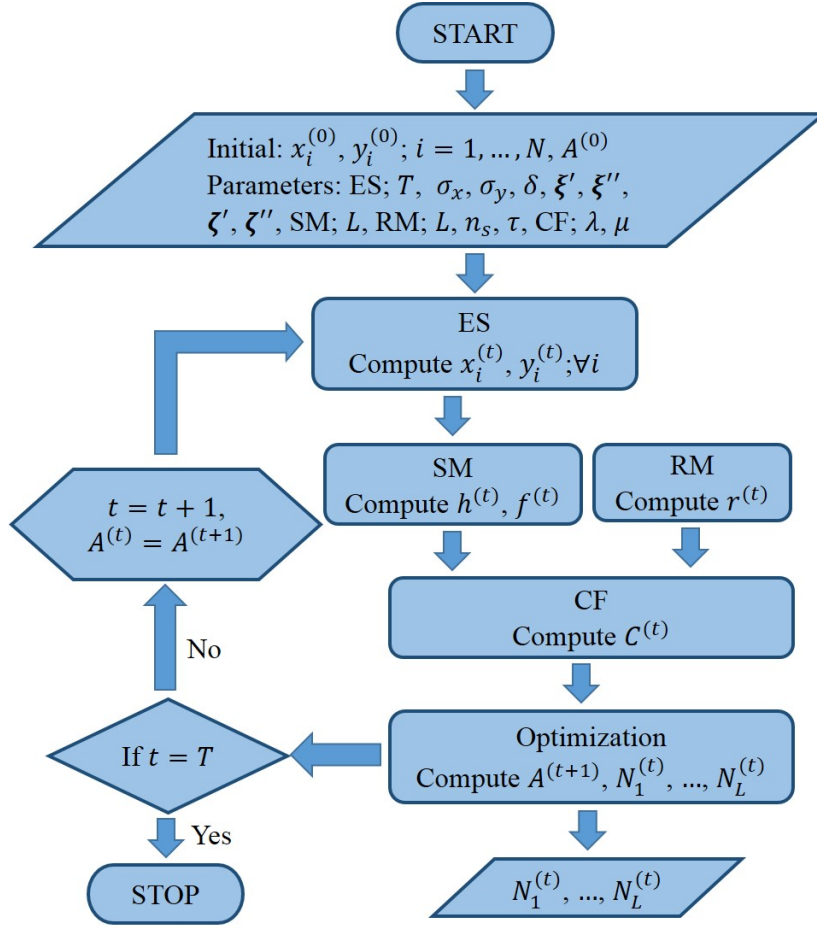


Figure 5.4: Process diagram of the algorithm. Algorithm inputs initial conditions for variables hunger, fatigue, risk, and grouping adjacency matrix. Different parameters input by ES, SM, RM, and CF are explained in Sections 5.1, 5.2.1, 5.2.2, and 5.2.4, respectively. Algorithm validates the grouping adjacency matrix in each time steps with the new grouping information obtained by optimizing the CF. Algorithm outputs groups providing the efficient segregation at each time step and stops at the stopping criteria which is the maximum number of time steps.

it to reveal the efficient grouping at each  $t \in T$ . In the ES, we set the initial states of cows arbitrarily as  $\theta_i^{(0)} \in \mathbb{U}\{\varepsilon, \mathcal{R}, \mathcal{S}\}$  for  $i = 1, \dots, n$ . We set initial variables,  $x_i^{(0)}$  and  $y_i^{(0)}$  for  $i = 1, \dots, n$ , based on initial states as,

$$\left\{ \begin{array}{ll} x_i^{(0)} = 1 \text{ and } y_i^{(0)} \in \mathbb{U}[\delta, 1], & \text{if } \theta_i^{(0)} = \varepsilon, \\ x_i^{(0)} \in \mathbb{U}[\delta, 1] \text{ and } y_i^{(0)} = 1, & \text{if } \theta_i^{(0)} = \mathcal{R}, \\ \left\{ \begin{array}{l} x_i^{(0)} = \delta \text{ and } y_i^{(0)} \in \mathbb{U}[\delta, 1], \\ \text{or} \\ x_i^{(0)} \in \mathbb{U}[\delta, 1] \text{ and } y_i^{(0)} = \delta, \end{array} \right. & \text{if } \theta_i^{(0)} = \mathcal{S}. \end{array} \right. \quad (5.15)$$

We choose the following parameters for the ES:  $\xi'_i, \zeta'_i \in \mathbb{U}[\text{.0495}, \text{.0505}]$ ,  $\xi''_i \in \mathbb{U}[\text{.0995}, \text{.1005}]$ ,  $\zeta''_i \in \mathbb{U}[\text{.1245}, \text{.1255}]$ ,  $\delta = \text{.25}$ ,  $\sigma_x = \text{.05}$ , and  $\sigma_y = \text{.05}$ . We set the parameters in the CF and RM as  $n_s = 4$ ,  $\tau = \text{.2}$ ,  $\gamma_1 = \text{.33}$ , and  $\gamma_2 = \text{.33}$ .

Table 5.2: All plausible group sizes, if a herd of 12 cows is segregated into a maximum of three groups.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$n_1^{(t)}$	12	11	10	9	8	7	6	10	9	8	7	6	8	7	6	5	6	5	4
$n_2^{(t)}$	0	1	2	3	4	5	6	1	2	3	4	5	2	3	4	5	3	4	4
$n_3^{(t)}$	0	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	3	3	4

A herd of 12 cows can be segregated into a maximum of three groups in 19 different combinations of group sizes as shown in Table 5.2. We assign an index for each combination to simplify future referencing in figures. We also run the ES together with the CF for another two instances of convex parameters:  $\gamma_1 = \text{.5}$ ,  $\gamma_2 = \text{.25}$  and  $\gamma_1 = \text{.25}$ ,  $\gamma_2 = \text{.5}$ . We extract the intermediate result at  $t = 20$  for all three instances

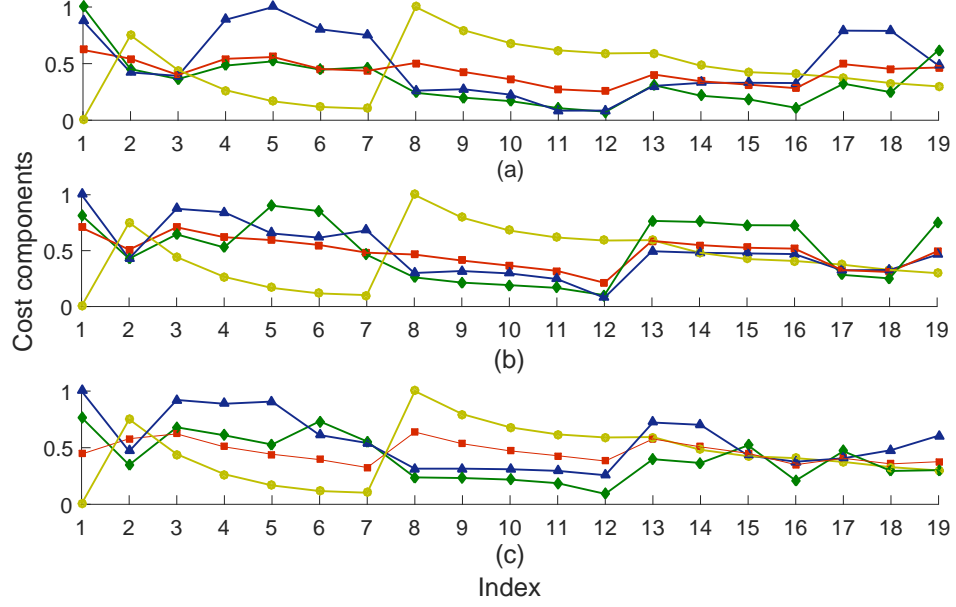


Figure 5.5: Total cost (red squares) at  $t = 20$  and its components: hunger (blue triangles), fatigue (green diamonds), and risk (yellow circles) versus the index representing the combinations of group sizes as given in Table 5.2. Total cost and its components for different convex parameters: (a)  $\gamma_1 = .33$  and  $\gamma_2 = .33$ ; (b)  $\gamma_1 = .5$  and  $\gamma_2 = .25$ ; and (c)  $\gamma_1 = .25$  and  $\gamma_2 = .5$ .

and present in Figure 5.5. The figure shows that the highest risk is encountered when  $n_{1,2,3}^{(20)} = 10, 1, 1$ , since two individual cows separated from the herd lead to a higher predation risk. The second highest risk occurs when  $n_{1,2,3}^{(20)} = 0, 1, 11$ , since one cow is separated from the herd. However, either staying together, as seen in the index one, or segregating equally with size larger than the safe size, as in the index 7, incurs the least risk. We can equally weight cost components in the convex representation (Figure 5.5(a)) or change the importance of components by weighting more on hunger (Figure 5.5(b)), fatigue (Figure 5.5(c)) or risk.

We analyze the temporal grouping using the instance  $\gamma_1 = .33$  and  $\gamma_2 = .33$ . Figure 5.6(a) shows cows freely switch their groups until they meet the optimum efficiency. The cow represented in red color remains in the same group for all the time, whereas

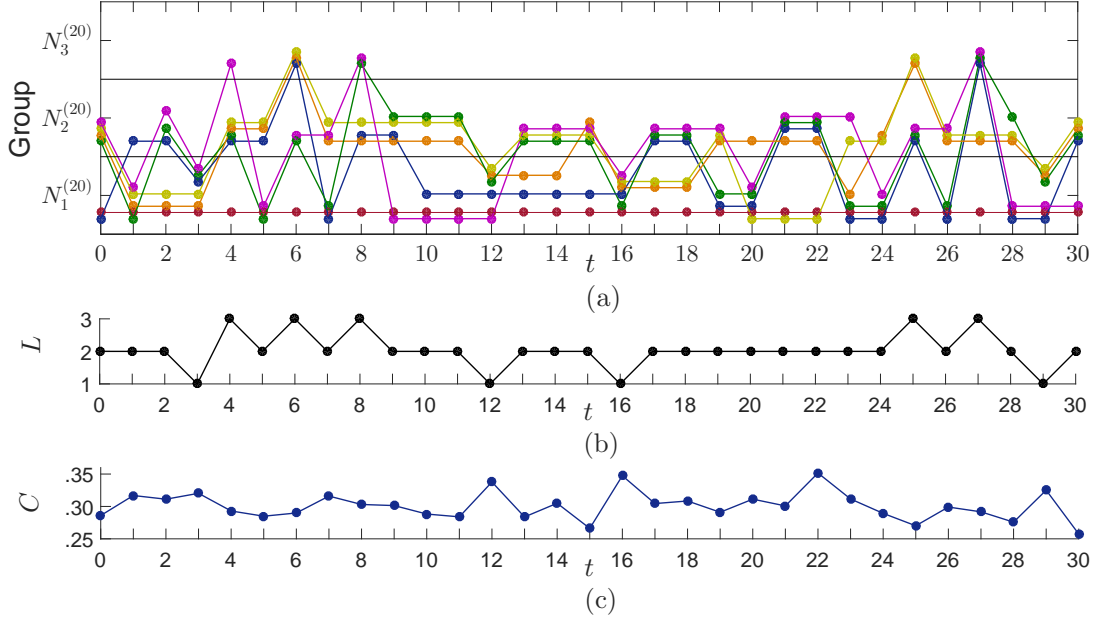


Figure 5.6: Grouping dynamics and related cost over time. (a) Occupancy,  $N_{1,2,3}^{(20)}$ , of the first six cows in three groups where each one represented in a different color. (b) The number of groups into which the herd is split is determined by (c) the total cost.

the one in green color switches its group frequently. Figure 5.6(b) representing total number of groups of the herd shows that the herd remain as a single group at  $t = 3, 12, 16$ , and  $29$ , and segregate into three groups at time steps  $t = 4, 6, 8, 25$  and  $27$ . Figure 5.6(c) representing the total cost reveals that the herd encounters a higher cost ( $t = 3, 12, 16$ , and  $29$ ) when cows stay as a single group and a lower cost ( $t = 15, 28$ , and  $30$ ) when the herd consists two or three groups.

#### 5.4.2 Inter-sexual segregation

We now study the inter-sexual grouping dynamics of a herd that consists of two distinguishable groups of adult cows: males and females. Adult male cows require more energy and rest, as their large body weights contrast to those of adult females having



comparatively small body weights. We therefore assume that the males' rates of change of hunger and fatigue are larger than those of females. Mathematically, we implement this assumption by using larger values for the parameters  $\xi'_i, \xi''_i, \zeta'_i$  and  $\zeta''_i$  of cows in the male group than for those in the female group.

We consider a herd of 10 cows consisting of two groups such that the first group has five cows ( $i = 1, \dots, 5$ ) with large body weights and the second group has the remaining five cows ( $i = 6, \dots, 10$ ) having small body weights. As in Example 5.4.1, we simulate hunger and fatigue of cows by the ES and reveal the efficient grouping by optimizing the CF. We set the initial states of cows in the first and second groups as eating and resting, respectively. Initial variables are also set similarly within groups except for the addition of a small amount uniform noise as

$$\begin{cases} \theta_i^{(0)} = \varepsilon \text{ and } (x_i^{(0)}, y_i^{(0)}) = (1, \delta) + \phi, & \text{for } i = 1, \dots, 5, \\ \theta_i^{(0)} = \kappa \text{ and } (x_i^{(0)}, y_i^{(0)}) = (\delta, 1) + \phi, & \text{for } i = 6, \dots, 10, \end{cases} \quad (5.16)$$

where  $\phi = 10^{-3}\mathbb{U}[-1, 1]$  and  $\delta = .25$ . We set  $\sigma_x = .2$  and  $\sigma_y = .2$  in ES, and we set the other parameters such that the first group consists of cows with large body mass

and the second group consists of cows with small body mass:

$$\left\{ \left\{ \begin{array}{ll} \xi'_i, \zeta'_i \in \mathbb{U}[\text{.2495}, \text{.2505}], \\ \xi''_i \in \mathbb{U}[\text{.2995}, \text{.3005}], & \text{for } i = 1, \dots, 5, \\ \zeta''_i \in \mathbb{U}[\text{.3995}, \text{.4005}], \end{array} \right. \right. \quad (5.17)$$

$$\left\{ \left\{ \begin{array}{ll} \xi'_i, \zeta'_i \in \mathbb{U}[\text{.0495}, \text{.0505}], \\ \xi''_i \in \mathbb{U}[\text{.0995}, \text{.1005}], & \text{for } i = 6, \dots, 10. \\ \zeta''_i \in \mathbb{U}[\text{.1225}, \text{.1255}], \end{array} \right. \right.$$

We set the parameters in the CF and RM to be  $n_s = 4$ ,  $\tau = .2$ ,  $\gamma_1 = .33$ , and  $\gamma_2 = .33$ . We run the ES for 30 time steps ( $T$ ) and then run the CF. Figure 5.7(a) shows that the cows occupy in their correct group during time steps 0–11, 15 and 20–28. However, cows mix between groups at time steps 12–14 and 16–19 and encounter a high cost as indicated by Figure 5.7(c). We compute the number of mismatched cows in groups and represent this in Figure 5.7(b). We observe that when the cost is large, the mismatching of cows is also large, and vice versa. We observe that the CF minimizes the cost and achieves the best grouping even after cows shuffle between groups.

In this chapter, we modeled an efficient segregation of a herd of cows by optimizing a cost function. The ES simulates cows' hunger and fatigue levels at each time step and then SM computes the cost incurred by hunger and fatigue. The predation risk

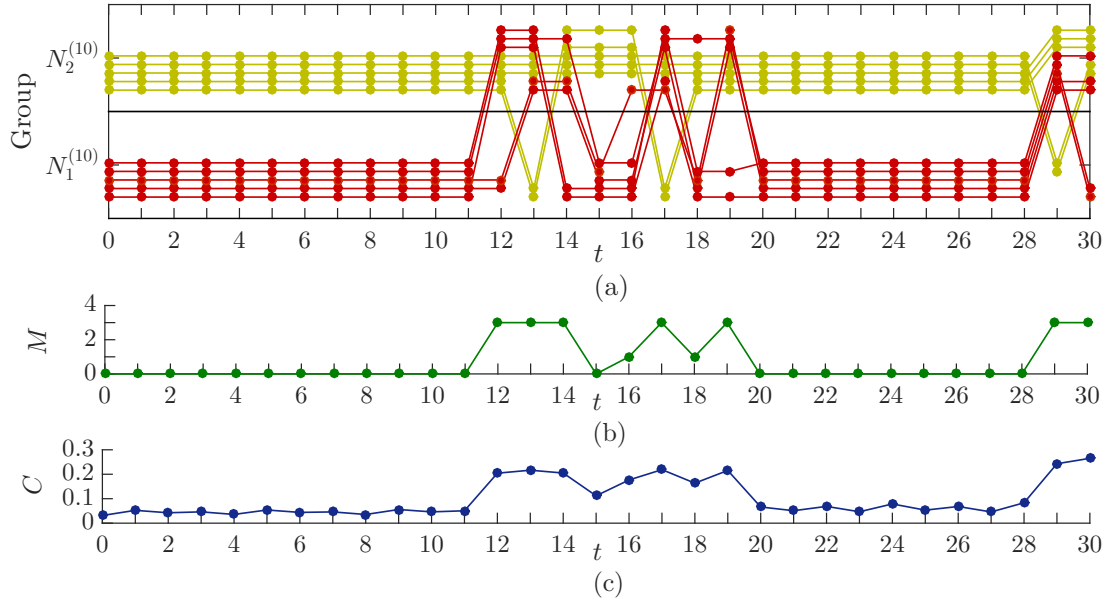


Figure 5.7: Dynamics of a herd of 10 cows in two groups and related cost over time. (a) First group containing five cows are colored in red, while the second group containing the other five cows colored in yellow. (b) Number of mismatched cows between groups in the herd. (c) The cost that determine the cows' occupancy in each group.

of a group is formulated in RM as an inverse exponential function of group size. The total cost is defined as a convex combination of hunger, fatigue, and risk. The CF is then optimized over all possible groups created from a input herd and determined the optimum segregation. A detailed discussion including future work is presented in Chapter 6.

## Chapter 6

# Concluding Remarks and Future Work

In this thesis, we developed a mathematical perspective of group behavior of animals. Manifold learning and dimensionality reduction of collective motion, and modeling of animal behavior are two main branches addressed in this work.

Multi-agent systems perform higher-dimensional dynamics, thus Dimensionality Reduction (DR) plays a vital role of revealing the lower-dimensional representation of the system as an underlying manifold. Dimensionality reduction algorithms are not always amenable to manipulate all types of data, thereby construction of new algorithms is an ongoing research. A phase transition in a multi-agent system often immobilizes the DR methods as the underlying manifold splits and represents individual phases. Thus, detection of transitions is essential before implementing DR algorithms. In fact, alternation between two activities creates the same alternation of the underlying manifolds. Characterization of this alternation without an aid of DR techniques is an interesting research. Synchronization of activities in multi-agent systems is beneficial for survival, however it may also encounters cost and then low-efficiency, as some of

the agents require to change their ideal activity to agree with the majority. Thus, the efficiency can be enhanced by modeling an segregation of the system.

In this chapter, we conclude remarks and future work under four sections as follows:

1. Dimensionality reduction by principal manifolds.
2. Detecting phase transitions using manifold's curvature.
3. Characterizing alternating manifolds.
4. An efficient segregation of a herd of cows.

## 6.1 Dimensionality reduction by principal manifolds

As collective motion is exhibited by multi-agent systems containing plenty of individuals, studying their behavior as a whole is non trivial. Dimensionality reductions is a smart tool which capable of identifying coarse observables and extract only the vital observables [50, 72, 89]. However, the DR is sometimes inept to reveal the underlying manifold or sometimes lose topological features.

In order to preserve the topology, we have introduced a novel DR algorithm that works directly on raw data by approximating the data in terms of cubic smoothing splines. We constructed the splines on two sets of non-overlapping slices of the raw data created using parallel hyperplanes that are known distance apart. The resulting splines are a grid-like structure that is embedded on a two-dimensional manifold. The smoothness of the splines was controlled via a smoothing parameter that selectively weighs fitting error versus the amount of noise rejection. The underlying manifold was

represented by the intersection points between all pairs of splines, while the embedding coordinates were represented in the form of distance along these splines.

Except the smoothing parameter, this method also inputs two parameters  $n_C^1$  and  $n_C^2$  for number of clusters in two directions of largest principal components. Small values for these parameters generate fewer clusters and thereby fewer splines in the intrinsic coordinate system. As the embedding was performed on these splines, sparse splines yield significantly large linear approximations to compute  $\delta x_1$  and  $\delta x_2$  in Section 2.1.3 and consequently encounter a high error. However, large values for these parameters generate dense grids and thereby they increase the computational complexity as computing their virtual intersections are highly cost according to Equation (2.27) in Section 2.3.2. Thus, choosing the ideal values for  $n_C^1$  and  $n_C^2$ , which are neither small nor large, depends on the user’s prior knowledge about the data.

The spline representation improves the regularity of noisy datasets. We demonstrated the applications of the algorithm on three separate examples including a paraboloid (Section 2.2.1), a noisy swiss-roll (Section 2.2.2), and a simulation of collective behavior (Section 2.2.3). Upon embedding these datasets on two-dimensional manifolds, we found that in the case of the paraboloid, the algorithm is successful in preserving the topology and the range of the data (Figures 2.8 and 2.9). In the case of the noisy swiss-roll, we found that the algorithm was able to reject high noise, thereby preserving the regularity in the original dataset (Figure 2.10). Unlike Isomap, our algorithm for finding PM is able to maintain a general structure (Figure 2.10 and 2.11). In other words, the algorithm fails gracefully giving the user enough indication of the trend of increasing error differently than Isomap, which fails abruptly and dramatically [85].

This property has a distinct advantage over other methods that are more sensitive to input parameters and therefore make it difficult to identify when the algorithm is embedding correctly. In contrast, the proposed algorithm on the swiss-roll dataset shows that in its current form, is inefficient to embed the dataset with holes faithfully onto a two-dimensional space, since it is highly focused on revealing a smooth underlying manifold (Figure 2.10).

We used a simulation of twenty particles moving in a translating circle as a representation of a two-dimensional data embedded in a forty-dimensional state (Section 2.2.3). In collective behavior, this form of motion approximates predator mobbing. Visual comparison between the embedding and the ground-truth available in the form of a rotated embedding shows that our algorithm is able to replicate the time-history on a low-dimensional space than for example Isomap (Figure 2.12). Furthermore, we saw that the range of axes in the embedding manifold available from our algorithm is closer to that in the original data than Isomap.

Operating and representing raw data for DR provides an opportunity for extracting true embeddings that preserve the structure as well as regularity of the dataset. We have also shown that this approach is amenable to high-dimensional dynamical systems such as those available in dataset of collective behavior.

The analysis of the error ( $\Gamma$ ) using distance preserving ability between data and embedding of our algorithm with respect to the smoothing parameter reveals that an increase in the smoothing parameter reduces the error in the structure (Section 2.3.1). This is expected because a high value of  $p$  implies that the cubic smoothing spline weights data fit more than smoothing. However, we also noted that the amount of

error saturates near  $p = 0.9$ , showing that values close to but not equal to one may also be used to represent the data (Figure 2.13). We observed a similar trend with respect to the amount of noise introduced in a swiss-roll dataset (Section 2.3.1), where we found that the amount of error increases quadratically with an increase in noise (Figure 2.14).

We found that the algorithm is able to work on sparse datasets (Section 2.3.1), up to a representation with five hundred points (Figure 2.15). The exponential fit indicates that the least error of the embedding is 0.2021 that is attained when  $T$  goes to infinity. This significant error is due to the approximations we made in Section 2.1.3 for computing intersections and local intrinsic coordinates.

### 6.1.1 Future work

We provided a useful visualization tool to analyze high-dimensional datasets by demanding a two-dimensional embedding. However, we will extend this two-dimensional approach to an arbitrary higher dimension by increasing the number of reference points. Each additional reference point will introduce an additional set of principal manifolds and adds one extra dimension to the intrinsic coordinate system. This extension will permit to analyze data sets that are inherently embedded in higher dimensions such as a sequence of images of an object which changes the orientation over time in three dimensions.



## 6.2 Detecting phase transitions

Multi-agent systems often undergo phase transitions due to natural perturbations [23, 31, 64, 96, 137]. Existing phase transitions in collective motion would always fail DR algorithms as transitions split the underlying manifold into two sub-manifolds at some singularities where the manifold’s curvature is not defined. Thus, prior to utilize such algorithms, careful study should be conducted to detect and eliminate such transitions.

Here, we have proposed a robust method to detect phase transitions in collective motion using ratio of singular values encountering high curvature of the underlying manifold. Therein, we first introduced an assertion to approximate the curvature of a curve by means of singular value ratios computed on it and then we extend the assertion to higher dimensions using the shape operator. Finally, we asserted that the same transition can also be detected through singular value ratios computed over the local data distribution on the manifold. We validated the method through four diverse examples ranging from simulations to natural.

In Section 3.3.1, we simulated a training data set of a particle swam consisting of known transitions to test the method’s performance. Noteworthy shifts of the distribution  $(\sigma_4/\sigma_1)_t$  from one transition to another justify the sensitivity of the method (Figure 3.8). Therein, our method was capable of detecting the exact transitions at frames 50 and 150, however, it approximated the transition at 100-th frame as at 99-th frame with 1 frame of error. Since this approximation was accurate enough, we partitioned the manifold into four sub-manifolds about the frames 50, 99, and 150. Thus, we conclude that the method is adept of perceiving the nature of the collective motion

and spatial distribution of agents to approximate transitions.

The algorithm running on a school of fish consisting four transitions ranked their magnitudes and provided a comparison (Figure 3.11). Embedding dimensionalities of sub-manifolds in each example revealed by isomap affirmed that the data was embedded in distinct smooth sub-manifolds which were joined at singularities. Examples also ensured the applicability of the method for variety of data sets ranging from simulations to natural (Figures 3.9 and 3.10), as the method requires one input parameter  $\alpha$  for number of nearest neighbors.

We detect transitions based on the curvature which is a local property on the manifold, thus, choosing the best value for  $\alpha$  is always important for the method to extract correct traditions. Therefore a user can either use one's prior-knowledge about the data or can run the algorithm with several  $\alpha$ 's to determine the best value which differentiates and highlights transitions. However, we can always rely on a small  $\alpha$  value when the data on the manifold is sufficiently dense [130]. As video recording devices are featured with high frame rates, we can always generate sufficiently large quantity of frames those would then be densely represented as data points on a manifold. Thus, in this study, we ran the transition detection algorithm with all the  $\alpha$  values less than 10 and then decided the best  $\alpha$  by analyzing corresponding transition plots.

### 6.2.1 Future work

In future, we will fabricate an automated extension for this approach to determine the best  $\alpha$  which gracefully extracts the significant transitions. For that, we will use a clustering method such as  $k$ -mean clustering to distinguish significant transitions at

each  $\alpha$  and then choose the best  $\alpha$  which provides the strongest clustering. We will also establish a novel approach of detecting transitions under gradual changes of the behavior of a multi-agent system which is not considered in this method. Therein, we will extract the trend component of the time series  $(\sigma_\alpha/\sigma_1)_1, \dots, (\sigma_\alpha/\sigma_1)_T$  and then analyze the local slope to detect gradual changes.

### 6.3 Characterizing alternating manifolds

Multi-agent systems sometimes alternate between two behaviors causing the same alternation in the representative underlying manifolds [1, 34, 136]. A dimensionality reduction free framework to characterize an alternation provides an efficient tool to manipulate such data. Thus, we have proposed a novel metric which is plausible of characterizing alternating manifolds of a multi-agent system.

This method takes two-dimensional positions at each time step as input. Individual positions in successive configurations are mapped for each agent by a simple bijective mapping. While more accurate methods exist for individual tracking [143], we found that this approach of having computational complexity  $O(\log N)$  is fast and then robust to errors in position assignments to individual agents in successive configurations, much in the spirit of a coarse observable.

The relative importance of average speed, polarization, and normalized connected components associated with the convex representation (Equations (4.14) and (4.15)) in the proposed metric can be changed by tuning the scalars  $\gamma_1$  and  $\gamma_2$  so that the metric can effectively encounter the similar group activities. Compared to [72], where

average nearest neighbor distance is used to analyze group motion, the combined metric proposed here captures change across multiple group motions. Different than [72], the connected component weighting is able to identify differences in group structure in terms of group splits.

Three diverse examples of changing speed (Section 4.2.1), coordination (Section 4.2.2), and structure (Section 4.2.3) are simulated using an augmented Vicsek model to validate our novel method of identifying manifolds. Once distinct manifolds were revealed on the basis of metric distance, Isomap DR algorithm was used to compare their dimensionalities. The three examples presented here offer some validation of our method and provide useful insight into the possibility of handling alternating group activities of multi-agent systems in a fully data-driven approach. Though the Vicsek model simulates agent positions in time and records them in the same order between all time steps, our method is also robust to the unavailability of such ordering information. This is particularly important in dealing with real experimental data, where tracking individuals and preserving their identities [69, 109, 139, 140] may be challenging.

The proposed metric is able to isolate distinct phases of collective motion in three separate scenarios. The separation of these phases as sub-manifolds in a low-dimensional space was confirmed by the Isomap algorithm, which measures a different dimensionalities for each of these phases. Further, based on metric distances it was possible to group similar activities together. This has potential application in using the proposed metric to train specific events of interest and process long videos to highlight the same.

### 6.3.1 Future work

While the artificial separation of phases was distinct in the simulated examples presented here, future work will evaluate the performance and sensitivity of this metric to gradual changes over a sequence of time steps. We will also seek to clarify inherent patterns of collective behavior across animal species [34] to characterize alternating manifolds through the novel metric presented herein. Since aforesaid framework inputs positions of agents, it reduces the applicability since most of collective motion data is videos. Accordingly, we will also develop an image processing technique to extract positions of agents from video frames using level curves created over the intensity plots. Therein, we will also utilize this method to analyze videos for classifying collective behavior [22].

## 6.4 An efficient segregation of a herd of cows

In nature, synchronization is essential for cows, as it helps them to protect from predators and attain their needs such as food [16, 47, 86, 115]. However, synchronization entails a low efficiency, as some portion of the herd has to change their behavior at a communal time rather than at a beneficial time [28, 45, 114]. To assure efficiency, we hypothesize that herds segregate into groups such that cows' hunger and fatigue are homogeneous within a group. We ensure that further segregation does not make small groups, which are then vulnerable to predation. Thus, we have developed a novel framework for modeling an efficient segregation of a herd of cows by optimizing a cost function that quantifies their hunger, fatigue, and predation risk.

The Evaluation Scheme (ES) in [128] inputs cows' time-dependent interactions in terms of an adjacency matrix  $A^{(t)}$ , which represents the efficient grouping revealed by optimizing the Cost Function (CF). This adjacency matrix provides an interface between two separate frameworks ES and CF. As hunger and fatigue are two independent perceptions of a cow, we optimized the CF in two times at each iteration and compute the two best segregations, corresponding to hunger and fatigue, and then choose the one giving the lower cost. The parameter  $L$  for the maximum number of groups limits the feasible region of the optimization and thereby reduces the computational complexity of the method. We assessed the cost incurred by hunger and fatigue using the standard deviation for the simplicity. However, the standard deviation can be replaced by any measure of dispersion.

The risk module also explains the ability to defense against predation, as a large group size devotes more defense to agents. We can imitate situations such as lack of defense power by assigning a high safety level ( $\tau$ ) even for a large safe size ( $n_s$ ). In contrast, a situation having a high defense power is imitated with the opposite of aforesaid setup of parameters.

We used Example 5.4.1 to analyze the dynamics of switching of cows between groups and fission-fusion of groups. Our primary hypothesis, that the synchronization encounters a high cost, was evidenced by Figures 5.6(b) and 5.6(c), as we observed a high cost during time steps 3, 12, 16, and 29 when the herd consists of one group. This is because the risk component gave a lower contribution to the total cost as compared to the hunger and fatigue components. However, this can be overcome by highly weighting the risk component in the convex combination. The example mimicked a

real-life scenario of a herd that chose to stay together for safety rather than segregating to compete for hunger and fatigue.

The ES can be customized by changing parameters denoting rates of hunger ( $\xi'_i, \xi''_i$ ) and rates of fatigue ( $\zeta'_i, \zeta''_i$ ) to simulate a different real-life herding instance as shown by Example 5.4.2 that mimics an inter-sexual segregation of a herd. Adult male cows generally possess larger body weights, and require more energy and rest than that of adult female cows. We implemented this asymmetry in the model by imposing larger values for  $\xi'_i, \xi''_i, \zeta'_i$ , and  $\zeta''_i$  in males than that for females. Sometimes, the hunger and fatigue of cows are escalated by the ES such that the grouping is unstable as shown during time steps 12–14, 16–19, and 29–30 in Figure 5.7(c). However, optimization of the CF faithfully established the stability as shown at the time step 15 and then at time steps 20–28. We added a uniform noise to the values of parameters  $\zeta'_i, \zeta''_i, \xi'_i$ , and  $\xi''_i$  for  $i = 1, \dots, n$  to assure the generality.

The coupling parameters,  $\sigma_x$  and  $\sigma_y$ , in ES determine the strengths of the interactions among the cows in groups. Consequently, imposing large values for these parameters increasingly influence the similar dynamics of group members such as dynamics of state changes. We set  $\sigma_x = \sigma_y = .05$  in Example 5.4.1 and  $\sigma_x = \sigma_y = .2$  in Example 5.4.2 as a higher potential grouping is expected for the herd undergoes inter-sexual segregation than regular segregation.

The CF can easily be adjusted such that it is robust for herding scenarios in different environments. A safe environment allows small groups in a herd, in contrast to a unsafe environment requires large groups to defend against attacks. The CF imitates a safe environment if the safety size  $n_s$  is large and the safety level  $\tau$  is small, or an unsafe

environment otherwise. One can control the influence from cost components, hunger, fatigue, and risk, on the CF by tuning the convex parameters  $\gamma_1$  and  $\gamma_2$  so that the method utilizes different grouping scenarios. One can even analyze the influence of one or more cost components over the others for the segregation by changing these parameters.

### 6.4.1 Future work

We primarily determine the grouping only by optimizing of the CF at each time step, thereby sometimes ES simulates hunger and fatigue those may lead to escalate the cost and encounter instability. We believe the optimization alone does not always preserve the stability of the segregation, thus in the future, introducing a learning process into the model that keeps track on perfect grouping during past time steps will be considered. This method requires the user to determine the maximum allowance of groups ( $L$ ) which is basically a arbitrary value . Choosing an appropriate value for this parameter is either obvious, as in Example in 5.4.2 where we used two, or hypothetical, as in Example 5.4.1 where we used three. Thus, we will improve the method such that it will reveal the best  $L$  providing the least cost for segregation by introducing a secondary optimization approach. In the future, we will also study the model performance under situations of having a strong defense power as well as lack of defense power by tuning parameters, safety level and safe size.



# Bibliography

- [1] Nicole Abaid, Erik Bollt, and Maurizio Porfiri. Topological analysis of complexity in multiagent systems. *Physical Review E*, 85(4):041907, 2012.
- [2] Stephen D. Albon, Timothy H. Clutton-Brock, and Rolf Langvatn. Cohort variation in reproduction and survival: implications for population demography. In *The biology of deer*, pages 15–21. Springer, 1992.
- [3] Gray Alfred. *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 1998.
- [4] Igor Rodrigues de Almeida and Claudio Rosito Jung. Change detection in human crowds. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*, pages 63–69. IEEE, 2013.
- [5] Sonia Altizer, Charles L. Nunn, Peter H. Thrall, John L. Gittleman, Janis Antonovics, Andrew A. Cunningham, Andrew A. Cunnningham, Andrew P. Dobson, Vanessa Ezenwa, Kate E. Jones, et al. Social organization and parasite risk in mammals: integrating theory and empirical studies. *Annual Review of Ecology, Evolution, and Systematics*, pages 517–547, 2003.

- [6] Ernesto L. Andrade, Scott Blunsden, and Robert B. Fisher. Hidden markov models for optical flow analysis in crowds. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 460–463. IEEE, 2006.
- [7] Marino Arroyo, Luca Heltai, Daniel Millán, and Antonio DeSimone. Reverse engineering the euglenoid movement. *Proceedings of the National Academy of Sciences*, 109(44):17874–17879, 2012.
- [8] Matteo Aureli, Francesca Fiorilli, and Maurizio Porfiri. Portraits of self-organization in fish schools interacting with robots. *Physica D: Nonlinear Phenomena*, 241(9):908–920, 2012.
- [9] Rangaswami Balakrishnan and Kanna Ranganathan. *A textbook of graph theory*. Springer, 2012.
- [10] Geoffrey H. Ball and David J. Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, DTIC Document, 1965.
- [11] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, Massimiliano Viale, et al. Empirical investigation of starling flocks: a benchmark study in collective animal behaviour. *Animal behaviour*, 76(1):201–215, 2008.
- [12] Ch Becco, Nicolas Vandewalle, Johann Delcourt, and Pascal Poncin. Experimental evidences of a structural and dynamical transition in fish school. *Physica A: Statistical Mechanics and its Applications*, 367:487–493, 2006.

- [13] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.
- [14] Jon L Bentley, Donald F Stanat, and E Hollins Williams. The complexity of finding fixed-radius near neighbors. *Information processing letters*, 6(6):209–212, 1977.
- [15] Gérard Biau and Aurélie Fischer. Parameter selection for principal curves. *Information Theory, IEEE Transactions on*, 58(3):1924–1939, 2012.
- [16] Nikolai W. F Bode, Jolyon J. Faria, Daniel W. Franks, Jens Krause, and Jamie Wood. How perceived threat increases synchronization in collectively moving animal groups. *Proceedings of the Royal Society of London B: Biological Sciences*, page rspb20100855, 2010.
- [17] Erik M. Bollt. Attractor modeling and empirical nonlinear model reduction of dissipative dynamical systems. *International Journal of Bifurcation and Chaos*, 17(4):1199–1219, 2007.
- [18] Alan C. Bovik. *Handbook of image and video processing*. Access Online via Elsevier, 2010.
- [19] Ronald Bracewell. *Fourier analysis and imaging*. Springer Science & Business Media, 2004.
- [20] Charles R. Brown and Mary B. Brown. Ectoparasitism as a cost of coloniality in cliff swallows (*hirundo pyrrhonota*). *Ecology*, 67(5):1206–1218, 1986.

- [21] Sachit Butail, Erik M. Bollt, and Maurizio Porfiri. Analysis and classification of collective behavior using generative modeling and nonlinear manifold learning. *Journal of Theoretical Biology*, 336:185–199, 2013.
- [22] Sachit Butail, Philip Salerno, Erik M Bollt, and Maurizio Porfiri. Classification of collective behavior: a comparison of tracking and machine learning methods to study the effect of ambient light on fish shoaling. *Behavior research methods*, 47(4):1020–1031, 2015.
- [23] Claudio Carere, Simona Montanino, Flavia Moreschini, Francesca Zoratto, Flavia Chiarotti, Daniela Santucci, and Enrico Allewa. Aerial flocking patterns of wintering starlings, *Sturnus vulgaris*, under different predation risk. *Animal Behaviour*, 77(1):101–107, 2009.
- [24] George Casella and Roger L. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.
- [25] Tim H. Clutton-Brock, Fiona E. Guinness, and Steve D. Albon. *Red deer: behavior and ecology of two sexes*. University of Chicago press, 1982.
- [26] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.
- [27] Larissa Conradt and Christian List. Group decisions in humans and animals: a survey. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1518):719–742, 2009.

- [28] Larissa Conradt and Timothy J. Roper. Activity synchrony and social cohesion: a fission-fusion model. *Proceedings of the Royal Society of London B: Biological Sciences*, 267(1458):2213–2218, 2000.
- [29] Larissa Conradt and Timothy J. Roper. Group decision-making in animals. *Nature*, 421(6919):155–158, 2003.
- [30] Iain D. Couzin. Collective cognition in animal groups. *Trends in cognitive sciences*, 13(1):36–43, 2009.
- [31] Iain D. Couzin, Jens Krause, Richard James, Graeme D. Ruxton, and Nigel R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1):1–11, 2002.
- [32] Trevor F. Cox and Michael A. A. Cox. *Multidimensional scaling*. CRC Press, 2010.
- [33] Scott Creel and John A. Winnie. Responses of elk herd size to fine-scale spatial and temporal variation in the risk of predation by wolves. *Animal Behaviour*, 69(5):1181–1189, 2005.
- [34] Pietro DeLellis, Giovanni Polverino, Gozde Ustuner, Nicole Abaid, Simone Macrì, Erik M Bollt, and Maurizio Porfiri. Collective behaviour across animal species. *Scientific reports*, 4, 2014.
- [35] Pietro DeLellis, Maurizio Porfiri, and Erik M. Bollt. Topological analysis of group fragmentation in multi-agent systems. *Physical Review E*, 87(2):022818, 2013.

- [36] Pietro DeLellis, Maurizio Porfiri, and Erik M. Bollt. Topological analysis of group fragmentation in multiagent systems. *Physical Review E*, 87(2):022818, 2013.
- [37] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on*, 8(1):148–154, 1997.
- [38] Jean-Louis Deneubourg and Simon Goss. Collective patterns and decision-making. *Ethology Ecology & Evolution*, 1(4):295–311, 1989.
- [39] Andreas Deutsch. Principles of biological pattern formation: swarming and aggregation viewed as selforganization phenomena. *Journal of biosciences*, 24(1):115–120, 1999.
- [40] Mario S. Di Bitetti and Charles H Janson. Social foraging and the finder’s share in capuchin monkeys, *cebus apella*. *Animal Behaviour*, 62(1):47–56, 2001.
- [41] Reinhard Diestel. *Graph theory*. Springer, 2000.
- [42] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [43] Wallace J. Dominey. Mobbing in colonially nesting fishes, especially the bluegill, *lepomis macrochirus*. *Copeia*, 1983(4):1086–1088, 1983.
- [44] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

- [45] Iva Dostálková and Marek Špinka. When to go with the crowd: modelling synchronization of all-or-nothing activity transitions in grouped animals. *Journal of theoretical biology*, 263(4):437–448, 2010.
- [46] Mark A. Elgar. Predator vigilance and group size in mammals and birds: a critical review of the empirical evidence. *Biological Reviews*, 64(1):13–33, 1989.
- [47] Inma Estevez, Inger-Lise Andersen, and Eric Nævdal. Group size, density and social dynamics in farm animals. *Applied Animal Behaviour Science*, 103(3):185–204, 2007.
- [48] James Ferryman. Data set of pet2009 at computational vision group, university of reading, 2009. <http://www.cvg.rdg.ac.uk>.
- [49] Andrew D. Fisher, Gwyneth A. Verkerk, Christopher J. Morrow, and Lisa R. Matthews. The effects of feed restriction and lying deprivation on pituitary–adrenal axis regulation in lactating cows. *Livestock Production Science*, 73(2):255–263, 2002.
- [50] Thomas A. Frewen, Iain D. Couzin, Allison Kolpas, Jeff Moehlis, Ronald Coifman, and Ioannis G. Kevrekidis. Coarse collective dynamics of animal groups. In *Coping with Complexity: Model Reduction and Data Analysis*, pages 299–309. Springer, 2011.
- [51] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

- [52] Kelum Gajamannage and Erik M. Bollt. Detecting phase transitions in collective motion using manifold’s curvature. *Mathematical biosciences and engineering: MBE*.
- [53] Kelum Gajamannage, Sachit Butail, Maurizio Porfiri, and Erik M. Bollt. Dimensionality reduction of collective motion by principal manifolds. *Physica D: Nonlinear Phenomena*, 291:62–73, 2015.
- [54] Kelum Gajamannage, Sachit Butail, Maurizio Porfiri, and Erik M. Bollt. Identifying manifolds underlying group motion in vicsek agents. *The European Physical Journal Special Topics*, 224(17-18):3245–3256, 2015.
- [55] Kelum Gajamannage, Mason A. Porter, Maurizio Porfiri, and Marian S. Dawkins. Modeling an efficient segregation of a herd of cows by optimizing a cost function. *Chaos: An Interdisciplinary Journal of Nonlinear Science*.
- [56] Michael Gashler, Dan Ventura, and Tony R. Martinez. Iterative non-linear dimensionality reduction with manifold sculpting. In *NIPS*, volume 8, pages 513–520, 2007.
- [57] Jan J. Gerbrands. On the relationships between svd, klt and pca. *Pattern recognition*, 14(1):375–381, 1981.
- [58] Robert Gerlai. High-throughput behavioral screens: the first step towards finding genes involved in vertebrate brain function using zebrafish. *Molecules*, 15(4):2609–2622, 2010.



- [59] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [60] Joseph F Hair. Multivariate data analysis. 2010.
- [61] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.
- [62] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.
- [63] Mark Hebblewhite and Daniel H. Pletscher. Effects of elk group size on predation by wolves. *Canadian Journal of Zoology*, 80(5):800–809, 2002.
- [64] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.
- [65] Dirk Helbing, Illes J Farkas, Peter Molnar, and Tamás Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and Evacuation Dynamics*, 21:21–58, 2002.
- [66] Dirk Helbing, Joachim Keltsch, and Peter Molnar. Modelling the evolution of human trail systems. *Nature*, 388(6637):47–50, 1997.
- [67] Edward Jackson. *A user’s guide to principal components*, volume 587. John Wiley & Sons, 2005.
- [68] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

- [69] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, 2005.
- [70] Robert J. Kilgour. In pursuit of “normal”: A review of the behaviour of cattle at pasture. *Applied Animal Behaviour Science*, 138(1):1–11, 2012.
- [71] Michael Kirby. *Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns*. John Wiley & Sons, Inc., 2000.
- [72] Allison Kolpas, Jeff Moehlis, and Ioannis G. Kevrekidis. Coarse-grained analysis of stochasticity-induced switching between collective motion states. *Proceedings of the National Academy of Sciences*, 104(14):5931–5935, 2007.
- [73] Jens Krause and Graeme D. Ruxton. *Living in groups*. Oxford University Press, 2002.
- [74] Christian C. Krohn and Lene Munksgaard. Behaviour of dairy cows kept in extensive (loose housing/pasture) or intensive (tie stall) environments ii. lying and lying-down behaviour. *Applied animal behaviour science*, 37(1):1–16, 1993.
- [75] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.
- [76] Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.
- [77] Stephane Lafon and Ann B Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set param-

- eterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1393–1403, 2006.
- [78] David C. Lay, Steven R. Lay, and Judi J. McDonald. *Linear Algebra and Its Applications, Books a la Carte Edition*. Pearson Education, 2014.
- [79] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57(null):49–76, March 2004.
- [80] John M. Lee. *Introduction to smooth manifolds*, volume 218. Springer, 2012.
- [81] Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, and Thomas H. Cormen. *Introduction to algorithms*. MIT press, 2001.
- [82] Hongyu Li, Li Teng, Wenbin Chen, and I-Fan Shen. Supervised learning on local tangent space. In *Advances in Neural Networks–ISNN 2005*, pages 546–551. Springer, 2005.
- [83] Steven L. Lima and Lawrence M. Dill. Behavioral decisions made under the risk of predation: a review and prospectus. *Canadian Journal of Zoology*, 68(4):619–640, 1990.
- [84] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
- [85] Nathan Mekuz and John K. Tsotsos. Parameterless isomap with adaptive neighborhood selection. In *Pattern Recognition*, pages 364–373. Springer, 2006.

- [86] Michael Mendl and Suzanne Held. Living in groups: an evolutionary perspective. *Social Behaviour in Farm Animals. Keeling, LJ & Gonyou, HW (Eds.). CAB International, Wallingford, Oxon*, pages 7–36, 2001.
- [87] Jerod A. Merkle, Marie Sigaud, and Daniel Fortin. To follow or not? how animals in fusion–fission societies handle conflicting information during group decision-making. *Ecology letters*, 2015.
- [88] Noam Miller and Robert Gerlai. Redefining membership in animal groups. *Behavior research methods*, 43(4):964–970, 2011.
- [89] Noam Y. Miller and Robert Gerlai. Oscillations in shoal cohesion in zebrafish (danio rerio). *Behavioural brain research*, 193(1):148–151, 2008.
- [90] Mark M. Millonas. Swarms, phase transitions, and collective intelligence. Technical report, Los Alamos National Lab., NM (United States), 1992.
- [91] Lisbeth Mogensen, Christian C. Krohn, Jan T. Sørensen, Jens Hindhede, and Lone H. Nielsen. Association between resting behaviour and live weight gain in dairy heifers housed in pens with different space allowance and floor type. *Applied Animal Behaviour Science*, 55(1):11–19, 1997.
- [92] James Munkres. Topology. 2000.
- [93] Lene Munksgaard, Margit B. Jensen, Lene J. Pedersen, Steffen W. Hansen, and Lindsay Matthews. Quantifying behavioural priorities—effects of time constraints on behaviour of dairy cows, bos taurus. *Applied Animal Behaviour Science*, 92(1):3–14, 2005.

- [94] Soraia Raupp Musse and Daniel Thalmann. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation'97*, pages 39–51. Springer, 1997.
- [95] Máté Nagy, Zsuzsa Ákos, Dora Biro, and Tamás Vicsek. Hierarchical group dynamics in pigeon flocks. *Nature*, 464(7290):890–893, 2010.
- [96] SRStJ Neill and Jonathan M. Cullen. Experiments on whether schooling by their prey affects the hunting behaviour of cephalopods and fish predators. *Journal of Zoology*, 172(4):549–569, 1974.
- [97] Sameer A. Nene and Shree K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(9):989–1003, 1997.
- [98] Lone Harder Nielsen, Lisbeth Mogensen, Christian Krohn, Jens Hindhede, and Jan Tind Sørensen. Resting and social behaviour of dairy heifers housed in slatted floor pens with different sized bedded lying areas. *Applied Animal Behaviour Science*, 54(4):307–316, 1997.
- [99] Barrett O’neill. *Elementary differential geometry*, volume 2. Academic press New York, 1966.
- [100] Kodanda Panikuruba. Birds flying away – stock footage, you tube, 2011. <https://www.youtube.com/watch?v=5eRj8rt6muA>.
- [101] Nikolaos Papanikolopoulos. Data set of detection of unusual crowd activity available at robotics and vision laboratory, department of

computer science and engineering, university of minnesota, 2006.  
[http://mha.cs.umn.edu/proj\\_events.shtml](http://mha.cs.umn.edu/proj_events.shtml).

- [102] Julia K. Parrish. Do predators' shape fish schools: interactions between predators and their schooling prey. *Netherlands Journal of Zoology*, 42(2):358–370, 1991.
- [103] Brian L. Partridge. The structure and function of fish schools. *Scientific American*, 246(6):114–123, 1982.
- [104] Murray H. Protter and Philip E. Protter. *Calculus with analytic geometry*. Jones & Bartlett Learning, 1988.
- [105] Shankar Raman. Factors influencing seasonal and monthly changes in the group size of chital or axis deer in southern india. *Journal of Biosciences*, 22(2):203–218, 1997.
- [106] Amandine Ramos, Odile Petit, Patrice Longour, Cristian Pasquaretta, and Cédric Sueur. Collective decision making during group movements in european bison, bison bonasus. *Animal Behaviour*, 109:149–160, 2015.
- [107] Wouter-Jan Rappel, Alastair Nicol, Armand Sarkissian, Herbert Levine, and William F Loomis. Self-organized vortex state in two-dimensional dictyostelium dynamics. *Physical review letters*, 83(6):1247, 1999.
- [108] Erik M. Rauch, Mark M. Millonas, and Dante R. Chialvo. Pattern formation and functionality in swarm models. *Physics Letters A*, 207(3):185–193, 1995.

- [109] Branko Ristic, Ba-Ngu Vo, Daniel Clark, and Ba-Tuong Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 59(7):3452–3457, 2011.
- [110] Bradley D. Robért, Brad J. White, David G. Renter, and Robert L. Larson. Determination of lying behavior patterns in healthy beef cattle by use of wireless accelerometers. *American journal of veterinary research*, 72(4):467–473, 2011.
- [111] Ralph T. Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [112] Vladimir Y. Rovenski. *Topics in extrinsic geometry of codimension-one foliations*, volume 1. Springer, 2011.
- [113] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [114] Kathree E. Ruckstuhl. To synchronise or not to synchronise: a dilemma for young bighorn males? *Behaviour*, 136(6):805–818, 1999.
- [115] Kathreen E. Ruckstuhl. Foraging behaviour and sexual segregation in bighorn sheep. *Animal Behaviour*, 56(1):99–106, 1998.
- [116] Kathreen E. Ruckstuhl and Peter Neuhaus. Sexual segregation in ungulates: a comparative test of three hypotheses. *Biological Reviews of the Cambridge Philosophical Society*, 77(01):77–96, 2002.
- [117] Oksana Samko, David Marshall, and Paul L. Rosin. Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognition Letters*, 27(9):968–979, 2006.

- [118] Kathrin Schirmann, Núria Chapinal, Daniel M. Weary, Wolfgang Heuwieser, and Marian A. G. Von Keyserlingk. Rumination and its relationship to feeding and lying behavior in holstein dairy cows. *Journal of dairy science*, 95(6):3212–3217, 2012.
- [119] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- [120] Amy L. Schreier and Larissa Swedell. Ecology and sociality in a multilevel society: ecological determinants of spatial cohesion in hamadryas baboons. *American journal of physical anthropology*, 148(4):580–588, 2012.
- [121] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944. ACM, 2009.
- [122] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [123] Nirajan Shiwakoti, Majid Sarvi, Geoff Rose, and Martin Burd. Animal dynamics based approach for modeling pedestrian crowd egress under panic conditions. *Transportation Research part B: methodological*, 45(9):1433–1449, 2011.
- [124] Richard M. Sibly. Optimal group size is unstable. *Animal Behaviour*, 31(3):947–948, 1983.



- [125] Ricard V Solé, Susanna C. Manrubia, Bartolo Luque, Jordi Delgado, and Jordi Bascompte. Phase transitions and complex systems: Simple, nonlinear models capture complex systems at the edge of chaos. *Complexity*, 1(4):13–26, 1996.
- [126] James Stewart. *Calculus: early transcendentals*. Cengage Learning, 2015.
- [127] David J. T. Sumpter. *Collective animal behavior*. Princeton University Press, 2010.
- [128] Jie Sun, Erik M. Bollt, Mason A. Porter, and Marian S. Dawkins. A mathematical model for the dynamics and synchronization of cows. *Physica D: Nonlinear Phenomena*, 240(19):1497–1509, 2011.
- [129] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.
- [130] Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [131] Michael E. Tipping and Cambridge C. Nh. Sparse kernel principal component analysis. 2001.
- [132] James R. Usherwood, Marinos Stavrou, John C. Lowe, Kyle Roskilly, and Alan M. Wilson. Flying in a flock comes at a cost in pigeons. *Nature*, 474(7352):494–497, 2011.

- [133] Rune Vabo and Leif Nottestad. An individual based model of fish school reactions: predicting antipredator behaviour as observed in nature. *Fisheries Oceanography*, 6(3):155–171, 1997.
- [134] Laurens J. P. van der Maaten, Eric O. Postma, and Hendrik J. van den Herik. Matlab toolbox for dimensionality reduction. *MICC, Maastricht University*, 2007.
- [135] Laurens J. P. van der Maaten, Eric O. Postma, and Hendrik J. van den Herik. *Dimensionality reduction: A comparative review*. TiCC TR 2009–005, Tilburg University, The Netherlands, 2009.
- [136] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.
- [137] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics Reports*, 517(3):71–140, 2012.
- [138] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.
- [139] Zheng Wu, Nickolay I. Hristov, Tyson L. Hedrick, Thomas H. Kunz, and Margrit Betke. Tracking a large number of objects from multiple views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1546–1553. IEEE, 2009.

- [140] Zheng Wu, Nickolay I. Hristov, Thomas H. Kunz, and Margrit Betke. Tracking-reconstruction or reconstruction-tracking? comparison of two multiple hypothesis tracking approaches to interpret 3d object motion from several camera views. In *Proceedings of the Workshop on Motion and Video Computing*, pages 1–8. IEEE, 2009.
- [141] Shiming Xiang, Feiping Nie, Changshui Zhang, and Chunxia Zhang. Nonlinear dimensionality reduction with local spline embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1285–1298, 2009.
- [142] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 311–321. Society for Industrial and Applied Mathematics, 1993.
- [143] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm Computing Surveys (CSUR)*, 38(4):13, 2006.
- [144] Tao Zhao and Ramakant Nevatia. Tracking multiple humans in crowded environment. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–406. IEEE, 2004.