University of Groningen

Novel techniques of computational intelligence for analysis of astronomical structures

Mohammadi, Mohammad

DOI:
[10.33612/diss.196895383](https://doi.org/10.33612/diss.196895383)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2022

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):
Mohammadi, M. (2022). *Novel techniques of computational intelligence for analysis of astronomical structures*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen.
<https://doi.org/10.33612/diss.196895383>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

rijksuniversiteit
 groningen

Novel techniques of computational intelligence for analysis of astronomical structures

Proefschrift

ter verkrijging van de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus Prof. dr. C. Wijmenga
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op
dinsdag 25 januari 2022 om 11.00 uur

door

Mohammad Mohammadi

geboren op 29 juni 1984
te Bamyan, Afghanistan

Promotores:

Prof. dr. N. Petkov

Prof. dr. M. Biehl

Prof. dr. K. Bunte

Beoordelingscommissie:

Prof. dr. M. Opper

Prof. dr. F. Rossi

Prof. dr. G. Gaydadjiev

Contents

Acknowledgements

List of figures

List of algorithms

1	Introduction	1
1.1	Scope of this thesis	2
1.2	Outline	2
I	Astronomical Simulations	5
2	Manifold Alignment Aware Ants: a Markovian process for manifold extraction	7
2.1	Introduction	7
2.2	Background and notation	10
2.3	Homogeneous Markov Chain	13
2.3.1	Weights Based on Kernels	13
2.3.2	Weights Based on Tangent Spaces	14
2.4	Ant Colony: Non-Homogeneous Markov Chain	16
2.4.1	Manifold Alignment Aware Ants (M3A)	17
2.4.2	Complexity analysis	19
2.5	Theoretical Analysis	20
2.5.1	Spectral Analysis	20
2.5.2	Pheromone distribution for a linear manifold	23
2.5.3	Pheromone distribution for a non-linear manifold	32

2.6	Experiments	35
2.6.1	Strategy for automatic hyper-parameter selection	35
2.6.2	Homogeneous Markov Chain	36
2.6.3	Manifold Alignment Aware Ants (M3A)	41
2.6.4	Improved density estimation	43
2.7	Conclusion	51
3	Multi-Agent based Manifold Denoising	53
3.1	Introduction	53
3.2	Related work	56
3.2.1	Problem setting	56
3.2.2	Manifold Blurring Mean Shift (MBMS)	56
3.2.3	Evolutionary Game Theory (EGT)	57
3.3	A Multi-Agent Paradigm for Manifold Denoising	57
3.3.1	Manifold Alignment Aware Agent	58
3.3.2	Evolutionary Manifold Alignment Aware Agents (EM3A) . .	60
3.3.3	Specific case (EM3A ⁺)	61
3.3.4	Complexity analysis	63
3.4	Experiments	64
3.4.1	Denoising manifolds	64
3.4.2	Manifold denoising as a preprocessing step	69
3.5	Conclusion	72
II	Astronomical Observations	73
4	Globular cluster detection in the halo of the Milky Way	75
4.1	Introduction	75
4.2	Laboratory for stellar evolution	77
4.3	Data and Preprocessing	78
4.3.1	Preprocessing	79
4.4	Methods	80
4.4.1	Nearest Neighbor Retrieval	81
4.4.2	Novelty Detection	82
4.5	Complexity analysis	84
4.6	Experiments	85
4.7	Results	87
4.8	Conclusion and Outlook	89

Contents

5	Detection of extragalactic Ultra-Compact Dwarfs and Globular Clusters using Explainable AI techniques	93
5.1	Introduction	94
5.2	Astronomical Data and Preprocessing	96
5.3	Methods	97
5.3.1	Learning Vector Quantization (LVQ)	97
5.3.2	Ensemble methods	99
5.3.3	Interpretability	100
5.3.4	Nonlinear visualization with charting	100
5.4	Experiments and Discussion	101
5.4.1	Experimental setup and Evaluation Measures	101
5.4.2	Discusssion	103
5.5	Conclusion	109
6	Summary	111
7	Future works	113
	Bibliography	115
	Samenvatting	125

Acknowledgments

My journey in Machine Learning (ML) started in 2014 in Mittweida where I did my Masters. During these years, I have had the opportunity to know many wonderful people, from supervisors to colleagues. These people have showed me that the academic environment is about not only doing research but also getting to know more people, doing coffee chat and etc. Therefore, these experiences not only made me a better researcher but also a better person.

During my stay in Mittweida, I had the privilege to meet two great professors. I would like to thank professor Titmann, who really helped me to develop my mathematical thinking, and professor Villmann who instructed me in learning ML. Coming from countries where lecturers are formal, Thomas surprised me about how a lecturer can be friendly and funny and still be an effective instructor. He asked me several times not to call him 'professor' which was difficult for me. During this time, I had a chance to be part of Computational Intelligence group. I want to thank Marika, David, Tina, Mandy and Mehrdad for sharing their knowledge during weekly meetings.

I would like to express my gratitude to Michael (B). Being TA in your courses was a great experience, and I learned many things during that time. I always enjoyed your talks, especially the cartoons in the slides. I am grateful to Peter. Your guidance had a significant role on my work about Ant Colony. In particular, your comments were a great source for me to understand how to present a precise proof. Thus, thanks for your instructions and your patience, in careful reading of the proofs.

My special thank to my daily supervisor, Kerstin, who was so supportive during my PhD. Indeed, your guidance was a key point in my PhD and it showed me how one can be well-organized and flexible simultaneously. Thanks for your generosity in sharing your experience: from your experience in writing/reviewing papers, to sharing templates for writing CV, cover letter, slides, to introducing necessary tools

(Git, Linux and etc.). In the first day of my PhD, when you suggested me to use Ubuntu (I guess it was the first time that I heard its name), it was so strange for me (because upto that time, I always thought every one use Windows). Later, I saw how necessary was to learn it, especially for our field. I also want to thank Sebastian for the lovely dinner that he provided.

I would like to thank Nicolai for his contribution in my first journal paper. I also had a chance to participate in your lectures, for parallel computing. These lectures provided me the necessary skills that later I used in my work on Ant Colony. I am also grateful to Reynier. It happened several times that you offered help even before I asked. I am thankful to Michael (W), the youngest person in spirit in the group with the highest energy. Your extensive knowledge about different topics makes you an encyclopedia which I benefited from during lunch time. Indeed, photos that you took in SUNDIAL meetings is a good source to remember great memories.

I would like to thank the members of the assessment committee, professor Manfred Opper, professor Fabrice Rossi and professor Georgi Gaydadjiev for the review of my thesis.

At the beginning of my PhD, Kerstin asked Sreejita to help me in case of having any question about some administrative issues. After that whenever I had a question, she was the first one that I asked the question. Thanks for your patience in answering my questions. In the third year of my PhD, I was lucky that Abolfazl joined our group. After that he became a close friend, and we spend a lot of time together: from discussing about our work in Ant Colony to playing tennis and to buying bike. Michiel, being in the same office had given me the chance to enjoy our discussions, and indeed playing cards on the train (on our trip to Mittweida) was a great memory to remember. Thus, I want to thank you both for your friendship.

When I started my PhD in the Intelligent Systems group, I immediately understood the meaning of *diversity*. Having people with different scientific backgrounds (from Electrical engineering to core computing science) and from different cultural background made this group a great place to spend time and do research. Thus, I would like to thank all of them, including Aleke, Caroline, Mohammad (B), Simon, Ahmad, Maria, Nicola, Jiwoo, Xiaoxuan, Hyoyin, Samira, Lauras, Petra, Jarvin, Godliver, Estefania, Jiapan, Elisa and Janis.

As a part of SUNDIAL network, I got the chance to visit big telescopes, in La Palma. Most importantly, I have got to know some wonderful people, such as Johan, Eija, Heiki, Sven and Marlies. I had also spent great time with my colleagues in this project: Alan, Thanh, Marco, Michele, Bahar, Shivangee, Angela, Alex, Teymour and Nushkia. Even though the only chance that I met some of you was during annual meeting, it was fun to go out during these short periods.

At the end, I would like to express my special thank to my family. My grand

mother, who was the first one in encouraging me for learning. My parents, who are patient and supportive on all decisions that I made. Malihe, my wife, her presence has changed my life and created new dimensions to it.

Mohammad Mohammadi
Groningen
December 20, 2021

List of Figures

2.1	An illustration of a curved manifold. In contrast to x_2 the points x and x_1 lie on the curved manifold. The Euclidean distance (dashed) to both x_1 and x_2 is equal (see panel a). As shown in panel b the distance to the tangent (dashed) reflects the closeness to the manifold much better.	14
2.2	The effect of (hyper-)parameters on the quality of the recovered subspaces: a) the effect of the number of samples N within the neighborhood ($\sigma = 0.01, r = 1$), b) the role of the noise level σ ($N = 1000, r = 1$), and c) the impact of neighborhood radius r ($N = 1000, \sigma = 0.01$).	24
2.3	1D manifold consisting of 2000 data points disrupted by Gaussian noise ($N(0, 0.1)$) depicted as gray points in panel a and b. The 120 points with highest density (using a Gaussian kernel) and pheromone values (resulting from Eq. (2.20)) are highlighted in red in a) and b) respectively. The development of the pheromone values over iterations t for points close to the manifold is steeply increasing (as shown in panel c), while it is abruptly dropping to zero for points far away (see panel d). Panel e) shows the evolution of the mean value of weights $a^{(t)}$ (see Eq. 2.29) and f) the KL- divergence from the stationary to the pheromone distribution over the iterations t	29
2.4	a) approximation of a piece of non-linear manifolds with a linear patch. b) the linear patch with an external node (o), simulating points outside the neighborhood.	33

2.5	2 views of the noisy s-curve data set (a,b), radius selection via average log likelihood for different r (c), and 1200 points with highest stationary distributions values for the MCs defined by Eq. (2.11) (e,h) and Eq. (2.12) (f,i), using $r = 0.6$	38
2.6	The impact of the new similarity measures on the t-SNE: Top row COIL20: a) using original t-SNE (perplexity 20) and b) using Eq. (2.11) ($d = 1, k = 10$). Middle row USPS: c) using original t-SNE (perplexity = 30) and d) using Eq. (2.12) ($k = 20$). Bottom row: average QGC measures for COIL20 and USPS (panel e and f). g) QGC curves for four classes of toy cars in COIL20 and h) of digits 3 and 5 in USPS.	40
2.7	Panel a): 1D circular manifold embedded in uniform background noise. Best results of: b) MD, c) MBMS, d) LLD, and e) LLPD (containing 2000 samples, i.e. 35%). Panel f): denoised M3A result where gray and red points show 2700 (45%) and 780 (%13) samples with $r = 3$ and $r = 2$	42
2.8	5 clusters formed by 0.02% of samples with highest pheromone values resulting from M3A ($r = 0.1, N_{\text{steps}} = 5 \times 10^6$) applied to a window in the sky between right ascension ($ra \in]180, 220[$ and declination ($dec \in]15, 30[$. b-e) the application of spectral clustering on the photometric data of two globular clusters: b,d) using Gaussian kernel as a similarity measure. c,e) using the transition probability in Eq. (2.14) ($r = 0.5$).	44
2.9	a) the noisy spiral data set, b-l) the contour curves for the density estimators: b) PW, c) MPW, d) GMM+rand, e) GMM+kmeans, f) SMM, g) Goldberger, h) FPW, l) M3A+FPW: neighborhood radius $r_{rw} = 0.13$ and balls' size $r_{\text{ball}} = 0.21$	46
2.10	a) two circular manifolds with variation in density and noise level, b-l) the contour curves for the compared eight density estimators. Note that black diamonds represent the means of the Gaussian distributions.	48
2.11	The simulated Jellyfish galaxy (viewpoint a and b). c-f) contour curves of density estimators projected by the first two principal directions of Gaussian co-variance matrices for zoomed regions a) (left) and b) (right) using Algorithm 2.3 (c,e) and GMM initialized by k-means (d,f).	50
3.1	Real ants' behavior in forming cemeteries (Theraulaz et al. 2003). They display four consecutive images of the plate.	55

3.2	Two views of the data set (a,b) and the results of EM3A (c,d). The fitness and population of losing (e,f) and winning (g,h) strategies. (i) the number of visited points with different dimensionalities. (j) computation time with number of processors used.	66
3.3	Gray points: a 1D manifolds varying in density and noise level. Red points: the denoised manifold, using: a) MD, b) LLD, c) SAF, d) MBM and e) EM3A. f) EM3A ⁺ : $N_{\text{iter}} = 6$. g) evolution of the MSE (to the ground truth) per iteration for values of $\rho \in \{0.1, 0.5, 1\}$	67
3.4	comparison of the original noisy basket data (panel a) to the results after applying b) MD: $\delta t = 0.02, k = 80, N_{\text{iter}} = 80$. c) SAF: $r = 0.6, \mu = 0.5, N_{\text{iter}} = 4$. d) MBMS: $k = 20, \sigma = \infty, N_{\text{iter}} = 4$. e) EM3A. f) EM3A ⁺ : $N_{\text{iter}} = 7$. g) the number of visited samples in given dimensionalities per iteration (for EM3A).	68
3.5	a) N-body cosmological simulation, b) SAF, c) EM3A ⁺	70
3.6	a) Fast Parzen Window (FPW), b) EM3A+FPW, c) EM3A ⁺ +FPW, and contour curves for the likelihood function of FPW, EM3A+FPW and EM3A ⁺ +FPW in panel d), e) and f) respectively.	71
4.1	The HR diagram of the globular cluster NGC 5024. Important groups are marked by RGB: Red Giant Branch, HB: Horizontal Branch, MS: Main Sequence.	78
4.2	Left: Magnitude distribution of a window of 0.5 degree centering at globular cluster NGC5024. Right: Magnitude distribution of a random window.	78
4.3	Summary of preprocessing: 1) divide the sky into smaller windows, 2) extract sources centered around the region of largest density and 3) compute 3D-KDE of the extracted data.	79
4.4	Comparison of densities using Euclidean distance and Dynamic Time Warping.	80
4.5	Number distribution of all the stars in Gaia DR1 with $10 < G < 20.7$. Boxes marked three selected regions with the positions: (1) $0^\circ < \alpha < 80^\circ$ and $-90^\circ < \delta < 0^\circ$, (2) $120^\circ < \alpha < 246^\circ$ and $-2^\circ < \delta < 60^\circ$, and (3) $310^\circ < \alpha < 360^\circ$ and $-90^\circ < \delta < 0^\circ$	85
4.6	Workflow of the process: at first, the preprocessing step is done (section 4.3.1), then we use two different approaches: Nearest neighbor retrieval (section 4.4.1) and Novelty detection (section 4.4.2). At the end, DoG (section 4.6) is used for the candidates obtained by two methods.	87

4.7	The average votes for the candidate using 40 NN retrieval and the circles mark interesting regions with the color indicating the similarity to known GCs.	89
4.8	Two candidates found at positions $(40, -34.5)$ and $(330.78, -51.3)$. Left: the distribution of stars and gray levels representing the magnitude values of sources from GAIA DR1. Right: stacked responses of spatial Difference of Gaussian filter at different magnitudes.	90
5.1	PCA-projection of the data colouring the background galaxies, UCDs/GCs and foreground stars (blue, red and green, classes 1-3), respectively (panel a). Corresponding Eigenvalue profile (panel b) and position of the objects in the sky in right ascension (RA) and declination (DEC) (panel c).	97
5.2	Panel (a) shows the test confusion matrix for the R_{L_2} model and panel (b) shows the corresponding ROC curve of minority class 2 vs. all the other classes and the corresponding AUC value of 0.99387.	104
5.3	Wrong (a) and correct (b) classification of two minority class observations. Each shows the distances of all samples projected using the local Ω^j of each prototype, highlighting the samples with corresponding label j in colour.	104
5.4	a) Manifold charting projection of the data by LGMLVQ (R_{L_2}) and b) conventional color-color diagram used in photometric selection by Astronomy.	105
5.5	Panel (a-c): class-wise feature relevance profiles of LGMLVQ R_{L_2} marking the top 6 in green and their corresponding percentage of contribution to the respective class and (d) the Random Forest (RF) feature relevance profile.	106
5.6	Pairwise plots of the R_{L_2} top 6 features for the UCDs/GCs class 2 (red markers) with focus on the area covered by that class.	107
5.7	Corresponding violin plots with prototype positions relative to the feature value distribution for minority class 2.	107
5.8	Difficult UCD/GCs example (bottom) in contrast with the the closest correct prototype (middle) and corresponding relevance profile (top). The 6 most relevant features for UCD/GCs LGMLVQ classification are marked green.	109

List of Algorithms

2.1	OneAnt	19
2.2	Manifold Alignment Aware Ants (M3A)	19
2.3	Partitioning the data space	45
3.1	One Agent	60
3.2	EM3A	62
3.3	One Strategy	63
3.4	EM3A ⁺	64
3.5	Covering data space with hyper-balls	69

Chapter 1

Introduction

Astronomy is one of the oldest fields of science involving the study of the universe beyond the Earth's atmosphere by asking fundamental questions about the evolution of our cosmos. Modern astronomy highly depends on data ranging from gamma rays and X-rays to radio bands, offering an enormous field exhibiting a huge demand for large scale data analysis and automated discovery of astronomical knowledge. Due to technical developments, the sensitivity of detectors continuously improves, and as a result, the new generation of telescopes produces images with higher resolution. For instance, while Sloan Digital Sky Survey (SDSS) ¹ collects 200GB per night ², the Thirty Meter Telescope (TMT), launching in 2022, will produce 90TB per night of visible spectrum (Kremer et al. 2017). Thus, the amount of data acquired a decade ago can now be collected in one night (Kremer et al. 2017). For any field including astronomy, entering in the big-data era arises one common question: how to extract valuable knowledge buried inside big datasets?

There are two general approaches for astronomers to study the formation and evolution of the universe: a) observations and b) simulations ³. A telescope can be considered a "time machine" due to the finite speed of light (Janz 2012). Thus, the more distant we observe, the longer we look back in time (Janz 2012). Therefore, we may observe different snapshots of the same process by looking at different distances. However, due to limitations of observations, astronomers use computers, as laboratories, to test and to investigate different scenarios. In both cases, since the number of observations and the number of particles ⁴ in simulations increase tremendously, we need some automated approaches to analyze them. In this thesis, we present several methods, applicable under different circumstances, to facilitate this process.

¹It is started in 2000.

²It is a multi-spectral imaging and spectroscopic redshift survey.

³While observations can present real cases, because of their limitations, astronomers also use simulations to study the universe.

⁴Note that each particle represents a group of stars.

1.1 Scope of this thesis

The current theories for galaxy formation propose that large-scale galaxies are hierarchically created from the merger of low-mass galaxies. The presence of stellar structures, such as streams; shells; and globular clusters, around large galaxies is considered as remnants of such mergers. Therefore, to uncover the formation and evolution of galaxies, it is necessary to study these structures. However, they are typically hidden in large-scale datasets, and robust automated techniques to extract them are highly desired. In this thesis, the goal is to present automated approaches to detect and/or classify such structures. Since different equipment provides different types of data, we present several algorithms to handle this diversity.

- two new algorithms to extract and clean the stellar structures when 3D positional information is available,
- two methods to uncover dense structures if only 2D positional information and 1D magnitude values are available,
- two techniques, from explainable AI, to classify astronomical objects using multi-wavelength imaging data,

As mentioned before, astronomical datasets may come from two major sources. Therefore, we divide the thesis into two parts⁵. Part I focuses on astronomical simulations where the 3D positional information, in addition to other features such as velocities and color information, is available. In part II, we consider datasets collected by telescopes where due to the viewpoint 2D positional information of objects in the sky, i.e. right ascension and declination, and some infrared and optical bands are available. Note that although we separate simulations from observations, the techniques developed for simulations can also be used for observations. For instance, we show in chapter 2 how to use Manifold Alignment Aware Ants (M3A) technique on GAIA survey.

1.2 Outline

In this section, we give an overview of the thesis, which is divided into two parts. In part I, the focus is on astronomical simulations where we propose two new algorithms to extract and denoise low-dimensional structures. Part II deals with observational datasets where we tackle several challenges, which are common for many

⁵Although we divide the thesis into two parts, we can still apply the techniques presented in part I on some astronomical observations, such as the GAIA data set.

surveys, such as the presence of imbalanced classes and the need for interpretable models.

The presence of stellar structures, such as streams and globular clusters, around large-scale galaxies is a common event ⁶. However, since they are surrounded by fore- and background-stars, an algorithm is needed to extract such structures. In chapter 2, we propose the *Manifold Alignment Aware Ants (M3A)* algorithm to extract stellar structures, with the emphasis on the ones with lower dimensionality, also called manifolds. We release a group of agents, who are equipped with a local alignment measure ⁷, in the data set where they search for stellar structures. During the walk, they deposit *pheromone* on the visited samples. Later, pheromone re-enforces the subsequent agents to follow the same route. The theoretical and empirical analyses demonstrate the ability of the strategy not only in removing background stars but also in highlighting the low-dimensional structures. In the experiment section, we also show that the resulting pheromone values can be used to build sparser probabilistic models to describe the distribution of particles in simulations.

In manifold-rich datasets such as cosmic web simulations, it is common that the low dimensional structures exhibit several difficulties: the structures are very noisy, vary in dimensionality and the density changes a lot along each manifold and across different ones. This makes the analysis so challenging especially since there exist millions of particles. The aim of chapter 3 is to clean the manifolds via pushing data points towards their respective central structure. We propose a new algorithm, called *Evolutionary Manifold Alignment Aware Agents (EM3A)*, where several agents walk in the data space and bring the data points closer to the manifolds. Moreover, principles from Evolutionary Game Theory are applied to find the appropriate value for the main hyper-parameter, i.e. neighborhood size. To evaluate the performance of EM3A, we apply the algorithm to a cosmic web simulation. Since the data depicts all the difficulties mentioned before, the denoising task becomes much more challenging. We demonstrate in several synthetic data sets and one astronomical simulation that the algorithm successfully not only denoises low-dimensional manifolds but also keeps high-dimensional structures unchanged.

To study our home galaxy, i.e. the Milky Way, a new mission, called GAIA, started in 2013. The GAIA DR1 provides 3-dimensional data, including 2D positional information in the sky and 1D G-band magnitude. In chapter 4, the goal is to extract stellar structures in the halo of the Milky Way from GAIA-generated data. While previous methods need the full phase-space or color information, which is expensive to acquire, the absence of such information in GAIA DR1 makes this task even more challenging. We tackle this problem by using magnitude distribution.

⁶They can be remnants of some mergers

⁷The agents use the measure to detect and to approximate manifolds.

We utilize 3D *Kernel Density Estimation* to approximate the magnitude distribution of stars. Then two different strategies, i.e. *nearest neighbor retrieval* and *anomaly detection*, are used to extract dense stellar structures. It is shown that both strategies are successful in recovering the known globular clusters, as well as potential candidates for further investigation.

In order to study the evolution of galaxy clusters, telescopes continuously collect images from different parts of the universe. In chapter 5, we focus on the Fornax galaxy cluster, the second closest galaxy cluster to us, and our aim is to find globular clusters (GCs) and ultra-compact dwarfs (UCDs) in its neighborhood. However, these structures in the Fornax cluster look like stars (point-sources); thus, differentiating them from foreground stars and background galaxies through imaging is a challenging task, and spectroscopic data is needed. However, spectroscopy for all point-sources is not possible, because of the need for longer exposure times. In this chapter, we would like to use the multi-wavelength images to extract GCs and UCDs from the collected data for the Fornax cluster. In order to do that, we need to address two challenges in our dataset: a) noisy features, and b) imbalanced classes. To tackle the first problem, we use *Localized Generalized Matrix Learning Vector Quantization (Localized-GMLVQ)* where it uses an adaptive metric to reduce the effect of noise. For the second problem, we use *Synthetic Minority Over-sampling (SMOTE)* to increase the number of samples in the minority class. The experiments demonstrate the success of both strategies in improving the classification performance especially with respect to the class of interest. In addition to achieving a good accuracy, the Localized-GMLVQ also provides feature relevance and representative samples for each class which are highly desirable for further investigation. Thus, in this chapter we implement a method which not only provides an excellent performance but also is highly interpretable⁸.

⁸Not only it provides the typical behavior within each class, but also it specifies the importance of features for each class.

Part I

Astronomical Simulations

Based on:

M. Mohammadi, Peter Tino and K. Bunte. – “Manifold Alignment Aware Ants: a Markovian process for manifold extraction.” *Neural Computation* (accepted).

Chapter 2

Manifold Alignment Aware Ants: a Markovian process for manifold extraction

Abstract

The presence of manifolds is a common assumption in many applications including astronomy and computer vision. For instance, in astronomy low-dimensional stellar structures, such as streams, shells and globular clusters, can be found in the neighborhood of big galaxies such as the Milky Way. Since these structures are often buried in very large data sets, an algorithm which can not only recover the manifold, but also remove the background noise (or outliers) is highly desirable. While other works try to recover manifolds either by pushing all points towards manifolds or by down-sampling from dense regions, aiming to solve one of the problems, they generally fail to suppress the noise on manifolds and remove background noise simultaneously. Inspired by the collective behavior of biological ants in food-seeking process, we propose a new algorithm which employs several random walkers who are equipped with a local alignment measure to detect and denoise manifolds. During the walking process the agents release pheromone on data points which reinforces future movements. Over time the pheromone concentrates on the manifolds, while it fades in the background noise due to an evaporation procedure. We use the Markov Chain (MC) framework to provide a theoretical analysis of the convergence of the algorithm and its performance. Moreover, an empirical analysis, based on synthetic and real world data sets, is provided to demonstrate its applicability in different areas, such as: a) improving the performance of t-distributed stochastic neighbor embedding (t-SNE) and spectral clustering using the underlying MC formulas, b) recovering astronomical low-dimensional structures and c) improving the performance of the Fast Parzen Window density estimator.

2.1 Introduction

New technological developments facilitate the collection of large amounts of high-dimensional data in different fields, such as astronomy, sensor networks, medical science, and computer vision. A typical challenge in dealing with high-dimensional

data is the Curse of Dimensionality, where the data space is sparse, such that data points are far from their neighbors. However, in practice, the high dimensional data is often generated by a system governed by a small number of underlying components (Dixit 2019). In other words, the high-dimensional data lies on a lower-dimensional topological structure called a manifold. Therefore, many dimensionality reduction methods, such as (Roweis and Saul 2000, Belkin and Niyogi 2003, Donoho and Grimes 2003, Zhang and Zha 2003, Coifman and Lafon 2006), aim to identify this underlying manifold from the data. However, such approaches might be tremendously impeded in their performance because of the presence of noise and outliers (Wang and Carreira-Perpinán 2010). This led to the development of methods aiming to deal with noisy manifolds and/or outliers.

There are different strategies to decrease the impact of noise on manifold learning algorithms. The technique proposed in (Little et al. 2020) denoises data sets via down-sampling and picking samples in high dense regions, whereas others suppress the noise level via pushing all instances towards region with higher density. While the first category can deal with background noise (or outliers not belonging to any manifold), the basic assumption in the second category is that the noisy samples belong to a manifold, and their deviations from the manifold are caused by measurement noise. An example technique is Manifold Denoising (MD) (Hein and Maier 2006), where the denoising process is modeled as a diffusion process on a neighborhood graph using the Laplacian to denoise the manifold. (Wang and Carreira-Perpinán 2010) proposes Manifold Blurring Mean Shift (MBMS), where the mean shift directions are only allowed to be parallel to manifold normals. Although increasing the number of iterations in the above algorithms may cause a better result, they eventually partition the manifold into local clusters. To overcome this problem, Locally Linear Denoising (LLD) (Gong et al. 2010) proposes a non-iterative method, assuming that a manifold can be explained as a set of overlapping linear patches. While it denoises each patch separately, it simultaneously uses the graph Laplacian to achieve a smooth manifold. However, such denoising methods may fail in applications, such as astronomy, where noisy manifolds are buried inside point clouds. Thus, a method capable of handling both types of noise, along with a manifold and background, is highly desirable.

Natural systems, such as ant colonies, have motivated many swarm algorithms in computer science. For instance, the cooperation among ants in the food-seeking process has inspired many solutions for optimization problems (Dorigo et al. 1991, Dorigo 1992, Dorigo et al. 1996, Stützle and Hoos 2000, Maniezzo 1999, Blum et al. 2001) and clustering (Tsai et al. 2004, Runkler 2005). In nature, when an ant leaves its nest to find food it deposits a chemical substance called pheromone on its path, which serves as information for other ants which are attracted to it. Each ants' deci-

sion is influenced by the pheromone: the higher the concentration is on a particular route, the more likely it is chosen as a path to follow. This seemingly uncoordinated local behavior in collection assists the hive to find the shortest path to the food source. Following this strategy, several algorithms were proposed and applied to optimization problems. Their simplicity and flexibility make these methods desirable for many optimization tasks that include graphs, such as vehicle/internet routing (Rizzoli et al. 2004) and water distribution systems (Gil et al. 2011).

Unfortunately, strategies explained above fail to denoise data comprising of noisy manifolds contained in point clouds, which are often faced in application domains involving particle simulations, such as astronomy. In this chapter, we propose a new algorithm to uncover noisy manifolds buried in high-dimensional background noise. Motivated by ant colony optimization strategies, our method employs multiple artificial ants who jump from point to point based on defined preferences and release pheromone. The pheromone serves as positive feedback that accumulates and concentrates on the manifolds, leaving the noisy points with relatively less pheromone and hence highlighting the structures.

With empirical experiments we demonstrate the capability of our algorithm to highlight manifolds in synthetic and real-world applications. Moreover, we show that the pheromone distribution can be used to improve the performance of the Fast Parzen Window density estimator. In addition to an empirical analysis we provide a thorough theoretical analysis to verify the ability of pheromone to encode information about distances of data points to the underlying manifold. First we only focus on a linear manifold and use the Markov Chain framework to study the behavior of the algorithm. In order to extend our analysis to non-linear manifolds we assume that a manifold can be approximated by locally linear patches, and we analyze how the pheromone distribution asymptotically behaves. Our analysis shows that the pheromone sorts data points according to their distances to the linear patch, i.e. the closer a data point is to the linear patch, the higher pheromone level it has. In summary, we propose an algorithm that provides valuable information, called pheromone, about how far points are from the underlying manifolds. This information is beneficial not only to extract noisy manifolds buried in point clouds, but also to build more effective density estimators.

The organization of this chapter is as follows. In section 2.2, we provide background information to study random walks in general. Section 2.3 contains three alternative transition probabilities for random walks and our proposed method to extract manifolds from a point cloud is exhibited in section 2.4. A theoretical analysis of the role of the pheromone and its convergence is presented in section 2.5. Section 2.6 contains an empirical analysis on synthetic data sets and real-world application examples, demonstrating the performance in recovering manifolds and

improving visualization and density estimation techniques. Finally, we conclude this chapter by summerizing the results 2.7,

2.2 Background and notation

Let $X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ be the data matrix storing N D -dimensional data points in its columns. Given a number $r > 0$, we define the neighborhood of any point $\mathbf{x}_i$ as follows:

$$B_r(\mathbf{x}_i) = \{\mathbf{x}_j \mid \|\mathbf{x}_i - \mathbf{x}_j\|^2 < r, 1 \leq j \leq N\}, \quad (2.1)$$

where $\|\cdot\|$ denotes the Euclidean norm. This definition of neighborhood has an advantage of being geometrically motivated (Belkin and Niyogi 2003), which also inspires our theoretical analysis. Alternatives are methods such as k-nearest neighbor and mutual k-nearest neighbor (Von Luxburg 2007). In the neighborhood graph nodes correspond to the data points from X and $(\mathbf{x}_i, \mathbf{x}_k)$ represents the directed edge connecting $\mathbf{x}_i$ to $\mathbf{x}_k$, with the associated weight $w(\mathbf{x}_i, \mathbf{x}_k) \geq 0$. The usual way to describe a Markov Chain (MC) is to use a weighted graph where the nodes denote the states of the MC, and its weights represent the probability of transitions between the states. Thus, from the above neighborhood graph one can construct an MC by row-normalization, i.e.:

$$p_{ij} = \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k \in B_r(\mathbf{x}_i)} w(\mathbf{x}_i, \mathbf{x}_k)} \quad (2.2)$$

where p_{ij} is the transition probability of jumping from $\mathbf{x}_i$ to $\mathbf{x}_j$. A popular MC process is the random walk where a walker follows the transition probabilities to move on the graph. We will now review some of the key results and notions from the MC theory that will be needed in our study.

Assume $S = \{1, 2, \dots, N\}$ represents the state space of a system. Let X_n denote the random variable of the state of the system in the n -th time step. In an MC, we assume that the next state of the system only depends on the current state, and the system's behavior is described through transition probabilities:

$$p_{ij} = P(X_1 = j \mid X_0 = i), \quad \forall i, j \in S \quad (2.3)$$

with X_0 being the initial state of the system. In this section, we only consider homogeneous MCs, where the transition probabilities are independent of the time n ¹. Now, let $P = [p_{ij}]$ be a matrix containing the transition probabilities. While P

¹While the proposed MC in this chapter is non-homogeneous (because of changes in pheromone vector), the MC can still be consider homogeneous between two updates of the pheromone.

includes the transition probabilities for one time step, it can be extended to arbitrary number of steps:

Theorem 1. For a MC with the transition probability matrix P , we have

$$P(X_n = j | X_0 = i) = p_{ij}^{(n)}$$

where $p_{ij}^{(n)}$ is (i, j) -th element of the n -th power of the matrix P , i.e. $P^n = \underbrace{P \times \cdots \times P}_{n \text{ times}}$ (Kulkarni 1999).

This distribution is used to study an MC in a limited time interval. Let $N_j(n)$ denote the number of visits state j accumulates over $(n + 1)$ steps including the initial state. To study the behavior of a chain in this interval the occupancy time is defined as follows:

Definition 2.2.1. For any state j and initial state i , the occupancy time of state j up to n is the expected number of times the system spends in state j in a random walk of $n + 1$ steps starting in i , i.e.:

$$m_{ij}(n) = E[N_j(n) | X_0 = i] ,$$

where $E[\cdot]$ denotes the expectation operator.

Let $M(n) = [m_{ij}(n)]$ denote the occupancy time matrix. The following theorem connects the occupancy time to the power of transition matrix (Kulkarni 1999):

Theorem 2. Given an MC, its occupancy time matrix is

$$M(n) = \sum_{q=0}^n P^q . \quad (2.4)$$

Proof. Let fix i and j and define the following random variable:

$$Z_n = \begin{cases} 1 & X_n \neq j \\ 0 & \text{otherwise} \end{cases}$$

Then, we have

$$N_j(n) = Z_0 + Z_1 + \cdots + Z_n$$

From the definition of $m_{ij}(n)$, it can be written:

$$\begin{aligned} m_{ij}(n) &= E(Z_0 + Z_1 + \cdots + Z_n | X_0 = i) = \sum_{q=0}^n E(Z_q | X_0 = i) \\ &= \sum_{q=0}^n E(Z_q = 1 | X_0 = i) = \sum_{q=0}^n E(X_q = j | X_0 = i) \\ &= \sum_{q=0}^n p_{ij}^{(q)} \end{aligned}$$

The matrix form of the latter equality is Eq. 2.4. □

While the above theorem helps to study an MC in a finite number of time steps, we would like to investigate the long-term behavior of the chain, which is encoded in the stationary distribution:

Definition 2.2.2. For an MC with the transition probability matrix P the distribution $\pi = [\pi_1, \dots, \pi_N]$ is called its stationary distribution if the following balance equation holds:

$$\pi P = \pi . \quad (2.5)$$

Here, we only focus on a specific class of Markov Chains called *ergodic chains*².

Definition 2.2.3. An MC is called ergodic if the following condition holds:

$$\exists T \in \mathbb{N} : \forall n > T, \forall i, j \in S \quad P(X_n = j | X_0 = i) > 0 .$$

In other words, for sufficiently large number of time steps n the system can be found in any state. Following theorem shows that the stationary distribution captures all information about the long-term behavior of an ergodic MC.

Theorem 3. For an ergodic MC its stationary distribution π is unique with:

$$\forall i \in S; \quad \lim_{n \rightarrow \infty} P(X_n = j | X_0 = i) = \pi_j , \quad (2.6)$$

$$\lim_{n \rightarrow \infty} P(X_n = j) = \pi_j , \quad (2.7)$$

$$\text{and the expected return time to state } j \text{ is } \mu_j = \frac{1}{\pi_j} . \quad (2.8)$$

Due to the lengthy proof, we refer the interested reader to (Cinlar 2013, Kulkarni 1999).

²Note that the way the neighborhood is defined in (2.1), the ergodicity condition is equivalent to having a connected neighborhood graph, which is a mild condition.

2.3 Homogeneous Markov Chain

An important class of manifold learning techniques relies on weighted graph representations of data. Markov Chains constructed on graphs offer a powerful tool to represent non-linear manifolds and are used in several dimensionality reduction techniques. This is exemplified by a well-known family of dimensionality reduction techniques called "kernel eigenmap methods", which includes local linear embedding (Roweis and Saul 2000); Laplacian Eigenmaps (Belkin and Niyogi 2003); Hessian Eigenmaps (Donoho and Grimes 2003); local tangent space alignment (Zhang and Zha 2003) and diffusion map (Coifman and Lafon 2006). The general idea is that the eigenvectors of Markov transition matrices are used to project high-dimensional data to a lower dimensional Euclidean space preserving the main structures of the data (Coifman and Lafon 2006). Therefore, applying MCs in the context of manifold learning has been employed in several techniques.

Although manifold learning techniques assume that the data lies on a lower-dimensional topological structure, in practice, it is rarely the case due to the presence of noise. One way to improve their performance is to use "less noisy data" closely aligned with the underlying manifold. In this chapter, the goal is to perform a random walk that places emphasis on such manifold aligned sub-samples. In the following, we introduce three MCs that can recover a manifold by highlighting sample points closer to it.

2.3.1 Weights Based on Kernels

A common way to construct an MC on a data set is to use kernel functions (Berry and Sauer 2016), i.e. a map $K : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ that fulfills:

- $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^D \quad K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})$
- $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^D \quad K(\mathbf{x}, \mathbf{y}) \geq 0$,

and quantifies a "similarity" between pairs of data points. The most popular kernel function is the Gaussian, which for two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as:

$$K_\sigma(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) , \quad (2.9)$$

with the Euclidean distance $\|\cdot\|$ and scale parameter $\sigma > 0$. Based on the Gaussian kernel the transition probability reads:

$$p_{ij} = \frac{K_\sigma(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k \in \mathcal{N}_i} K_\sigma(\mathbf{x}_i, \mathbf{x}_k)}$$

Figure 2.1: An illustration of a curved manifold. In contrast to x_2 the points x and x_1 lie on the curved manifold. The Euclidean distance (dashed) to both x_1 and x_2 is equal (see panel a). As shown in panel b the distance to the tangent (dashed) reflects the closeness to the manifold much better.

with $\mathcal{N}_i$ being the set of point x_i 's neighbors. This definition of transition probabilities reinforces random walkers to spend more time in denser regions.

2.3.2 Weights Based on Tangent Spaces

Assuming the presence of a manifold the noise level of a data point can be related to its distance from the manifold. Thus, to form a MC that encourages random walkers to spend more time on the manifold the walkers need to know how far data points are from the manifold. As depicted in Fig. 2.1 a) the Euclidean distance fails to reveal this information. There, if a random walker resides on point x the Gaussian kernel gives the walker the same chance to stay on the manifold (point x_1) or to leave it (jumping to point x_2). In order to overcome this drawback, we need to define a favorable measure that improves the approximate distances to the manifold.

A manifold can be approximated locally at a point x by its tangent space (Tu 2011). Hence, we propose to use the tangent space to estimate the distance of x 's neighbors to the manifold (see Figure 2.1b). Let $\mathcal{N}$ represent the set of x 's neighbors, for example enclosed in $B_r(x)$. The tangent space at x is typically approximated by Principal Component Analysis (PCA) on $\mathcal{N}^3$, yielding a set of eigenvalues and unit orthogonal eigenvectors of the local co-variance matrix, $\{(\lambda_k, u_k)\}_{k=1}^D$. Without loss of generality, we assume that the eigenvalues are in descending order with $\lambda_1 \geq$

³Note that methods, such as (Lerman et al. 2015), provide better estimation for the tangent spaces. However, since in this chapter we consider big data sets we use the less costly PCA.

$\lambda_2 \geq \dots \geq \lambda_D \geq 0$ ⁴ and normalized to $\sum_k \lambda_k = 1$. Then for a d -dimensional manifold the subspace spanned by the columns of $U_1 = [u_1, u_2, \dots, u_d]$ provides an estimation for the tangent space and locally for the manifold.

Since we consider noisy data sets, we only have access to a noisy version of the tangent space $\hat{U}_1$. Hence, depending on the noise level the difference between U_1 and $\hat{U}_1$ may be small or high. For more details, we refer the reader to (Kaslovsky and Meyer 2014) and (Little et al. 2017). In the following we introduce two ways to define MCs highlighting samples close to manifold structures.

A manifold with known intrinsic dimensionality: Consider a d -dimensional manifold in $\mathbf{x}$'s neighborhood. For any $\mathbf{x}_i \in \mathcal{N}$ one can estimate its distance to the tangent space by:

$$\delta_i^{\mathcal{M}} = \|(I - \hat{U}_1 \hat{U}_1^T)(\mathbf{x}_i - \mathbf{x}')\|, \quad (2.10)$$

where $\mathbf{x}'$ determines the place where the tangent space touches the manifold in the neighborhood. Based on these quantities, we define the weight values to be used in the calculation of transition probabilities Eq. (2.2) as follows:

$$w_d(\mathbf{x}, \mathbf{x}_i) = \begin{cases} 1 - \frac{\delta_i^{\mathcal{M}}}{\alpha} & \text{if } \alpha \geq \Delta_i \\ 0 & \text{if } \alpha < \Delta_i \end{cases}. \quad (2.11)$$

where α is a factor to ensure the weight values are positive. Here, we determine α such that only p percent of neighbors have non-zero weights. This definition constructs a random walk favoring jumps closer to the manifold. In other words, the random walker observes a flat region and it is more likely to move to a point close to it.

A manifold with unknown intrinsic dimensionality In most applications the dimensionality of a manifold d is unknown and eigenvalues are typically used to estimate it. As suggested in (Wang et al. 2008) the intrinsic dimensionality may be estimated by:

$$\hat{d} = \arg \max_d S_d,$$

where $S_d = d \cdot (\lambda_d - \lambda_{d+1})$, $d \in \{1, \dots, D-1\}$ and $S_D = D \cdot \lambda_D$. However, the performance of these types of criteria highly depend on the neighborhood size which is related to the noise level and the manifold curvature. In order to tackle this problem, we consider all possible values of d instead of picking a specific value. The

⁴In order to simplify the explanation, we make this assumption, and it is not needed in practice.

basic idea is to compute the weight values for any $d \in \{1, \dots, D\}$, according to Eq. (2.11), followed by the calculation of their mean:

$$w(\mathbf{x}, \mathbf{x}_i) = \sum_{d=1}^D S_d \cdot w_d(\mathbf{x}, \mathbf{x}_i) , \quad (2.12)$$

where the eigengap S_d indicates the importance of each intrinsic dimensionality. Since $S_d \geq 0$ and $\sum_{d=1}^D S_d = 1$, we may interpret S_d as the “probability” of the manifold being d -dimensional and the weight value w as the expected weight with respect to this distribution over manifold dimensionalities.

2.4 Ant Colony: Non-Homogeneous Markov Chain

The group behavior of decentralized natural systems, such as ant colonies and bird flocking, has inspired many methods in computer science and is often summarized under the keyword Swarm Intelligence (SI). For instance, the biological behavior of ants in the food-seeking process has motivated several methods in combinatorial optimization and clustering. In the process two mechanisms, one behavioral and one environmental, help to find the shortest path to a food source:

- **Deposition:** when a bio-ant walks to (and from) a food source it releases a substance called pheromone on the ground. Ants are attracted by it when they choose a path to follow. The more ants use a route the more pheromone accumulates increasing the chance for the path to be selected by subsequent ants as well.
- **Evaporation:** the pheromone evaporates over time and hence less attractive trails with less pheromone eventually decay.

Thus, although there is no central authority to control the ants’ behavior, their indirect form of local interactions, via pheromone, helps them to find the shortest path to the food.

In this chapter, we propose a new algorithm that aims to recover manifolds from noisy samples. The basic idea is that a set of ant-like agents are released in the data space to search for manifolds based on the following elements:

- O1: an MC that highlights underlying manifolds in the data set, and
- O2: deposition and evaporation mechanisms to update the amount pheromone on data points.

The aim is that ants are more likely to visit points close to manifolds and deposit pheromone. Higher pheromone levels function as a positive feedback mechanism reinforcing further visitations. Eventually, by extracting points with more pheromone potential manifolds can be uncovered. In the following we formally define and illustrate the two elements outlined above.

2.4.1 Manifold Alignment Aware Ants (M3A)

Inspired by swarm intelligence, M3A uses several ants who walk in the data space and search for manifolds. During the search process, the ants interact with one another through pheromone and with the environment. In section 2.3, we have already proposed possible formulations of transition probabilities for the random walk. However, they consider only the distribution of data points in the environment and do not provide any form of communication among ants. Thus, we modify the transition probabilities by adding a pheromone factor f . Given any data point $\mathbf{x}_i$, let f_i denote the amount of pheromone on it and $\mathcal{N}_i$ represents the set of its neighbors. The new transition probability is defined as:

$$p_{ij} = \frac{(w_{ij})^{1-\gamma} (\hat{f}_j)^\gamma}{\sum_{\mathbf{x}_k \in \mathcal{N}_i} (w_{ik})^{1-\gamma} (\hat{f}_k)^\gamma}, \quad (2.13)$$

where $w_{ij} = w(\mathbf{x}_i, \mathbf{x}_j)$. To ensure that the weights w and the pheromone f are in the same scale, i.e. $[0, 1]$, the pheromone is normalized within the neighborhood by $\hat{f}_j = \frac{f_j}{\sum_{\mathbf{x}_k \in \mathcal{N}_i} f_k}$. The $\gamma \in [0, 1]$ effectively controls how much ants are attracted to points already visited or are able to explore the data space for new regions. For instance, if we set γ to zero, then there would be no interaction among ants and, as a result, every ant could independently explore the data space. Hence, the new transition probability takes into account both types of interactions O1 and O2.

Pheromone is an indirect form of communication among artificial ants who release it on the visited points. Therefore, the amount of pheromone on data points vary and it should be updated over time. In order to make the algorithm suitable for parallelization the pheromone values are updated when the ants finish their walks of a predefined number of steps n , called *one round*⁵. Thus, the transition probability

⁵In one round, we can distribute the agents among several processors and update the pheromone after n steps. Otherwise, there should be communications between processors to synchronize pheromone values in every step, which causes unnecessary overhead.

for the $(t + 1)$ -th round can be re-written as:

$$p_{ij}^{(t+1)} = \frac{(w_{ij})^{1-\gamma} \left(\hat{f}_j^{(t)} \right)^\gamma}{\sum_{k \in \mathcal{N}_i} (w_{ik})^{1-\gamma} \left(\hat{f}_k^{(t)} \right)^\gamma}, \quad (2.14)$$

where $\hat{f}_j^{(t)}$ denotes the amount of pheromone on x_j after t rounds. Since the amount of pheromone on a data point depends on the number of visits and the extent of evaporation, the updating rule consists of two parts:

- deposition, based on the number of times a point has been visited, and
- evaporation, controlled by the hyper-parameter ρ , represent how much ants are allowed to explore new points.

Therefore, the pheromone level on any sample x_i can be updated as follows:

$$f_i^{(t+1)} = \frac{c}{M} \sum_{a=1}^M \frac{N_i^a(n)}{n+1} + (1-\rho) \cdot f_i^{(t)}, \quad (2.15)$$

where M is the number of ants and $N_i^a(n)$ is the number of times the a -th ant visits x_i over $n + 1$ steps (including initial state) in the $(t + 1)$ -th round. Note that the deposition term is divided by $M(n + 1)$ to prevent unlimited increase of the pheromone level (especially for big M or n). The constant $c > 0$ specifies the amount of pheromone a single ant deposits on a point in a single visit. The pseudo code of the M3A is provided in Algorithm 2.2 and the implementation is provided at <https://github.com/mohammadimathstar/M3A.git>.

In summary, we can highlight the following points:

- The proposed ant algorithm uses transition probabilities (Eq. 2.14) based on local tangent alignment and pheromone to highlight manifold structures, with the latter also reinforcing the agents to stay close.
- In contrast to previous works with the ant colony, the pheromone is deposited on the nodes, instead of edges. This vastly reduces the number of pheromone values that need to be stored ⁶. Moreover, the associated pheromone to each data point can be used to extract points close to manifolds.
- The amount of pheromone is updated after a pre-defined number of steps n . In practice, this allows to distribute the computations among several processors and parallelize the random walks. Then, at the end of each iteration, the master branch can receive the visitation profiles and update the pheromone vector.

⁶Since a graph is expected to have lower number of nodes, in comparison to the number of edges.

Algorithm 2.1 OneAnt

Input: data set X , pheromone vector $\mathbf{f}$, radius r , number of steps n
Output: number of visits $\mathbf{N}_v$
function ONEANT($\mathbf{f}$, X , r , n)
 $\mathbf{N}_v = [0, 0, \dots, 0]$
 Randomly select a node as the initial position (say x_i)
 $\mathbf{N}_v[i] = \mathbf{N}_v[i] + 1$
 for $s \leftarrow 1, n$ **do**
 randomly select its next destination based on weights and pheromone values, following Eq. (2.14) (say x_j)
 $\mathbf{N}_v[j] = \mathbf{N}_v[j] + 1$
 end for
end function

Algorithm 2.2 Manifold Alignment Aware Ants (M3A)

Input: data set X , radius r , number of steps n , number of ants M and number of iterations N_{iter}
Output: The pheromone vector $\mathbf{f} = [f_1, f_2, \dots, f_N]$
function M3A(X , r , n , M , N_{iter})
 Initialize the pheromone vector $\mathbf{f} = [f_1^{(0)}, f_2^{(0)}, \dots, f_N^{(0)}]$
 for $t \leftarrow 1, N_{\text{iter}}$ **do**
 $\mathbf{N}_v = [0, 0, \dots, 0]$
 for $a \leftarrow 1, M$ **do**
 $\mathbf{N}_v^a = \text{OneAnt}(\mathbf{f}, X, r, n)$
 end for
 $\mathbf{N}_v = \sum_{a=1}^M \mathbf{N}_v^a$
 Update pheromone by Eq. (2.15)
 end for
end function

2.4.2 Complexity analysis

As a preprocessing step for the M3A algorithm, we implement two operations on each data point: a) finding its neighbors and b) performing PCA on its neighborhood. To perform the neighbor search we need to calculate the distances of all pairs of N samples, which leads to the complexity of $\mathcal{O}(N^2)$. However, there are approximate nearest neighbor search strategies to reduce the complexity. For instance, the k-d tree algorithm (Bentley 1975) with the complexity $\mathcal{O}(N \log N)$ can significantly decrease the computational costs. Besides the neighbor search we perform local

PCA using Singular Value Decomposition. In the worst case, its implementation for any point x scales cubic with the dimensionality of the data and quadratic with the size of its neighborhood $|\mathcal{N}_x|$. Nonetheless, approximate strategies decrease the computational complexity (Golub and Van Loan 2012).

In addition to speeding up the preprocessing step via approximate strategies, the execution of the M3A can be accelerated by parallelization. In order to do that, we can distribute ants (*OneAnt* function in algorithm 2.2) among several processors with access to shared memory. Let assume there are one master and several worker branches, and the workers have access to the weight values coming from the preprocessing step. At the beginning of each round, the master branch sends the current pheromone vector to all workers, and then workers start doing the random walk. After finishing the walks, they send back their visitation profiles (i.e. N_v). In the end, the master branch, after receiving all messages, updates the pheromone vector and sends the updated version of the pheromone vector to all workers for the next round. This process repeats for any given number of times.

2.5 Theoretical Analysis

Many metaheuristic methods suffer from the lack of theoretical analysis. Therefore, their behavior and the effects of parameters on them are typically investigated only empirically. In this section, we provide an analysis of the convergence and the impact of hyper-parameters of the proposed algorithm in exemplary situations. More precisely, we consider a data set containing a noisy d -dimensional manifold and we use Eq. (2.11) to compute the weight values with α determined such that $p = 100\%$ of neighbors having non-zero weight values. First, we study the effect of noise on the performance of PCA and, as a result, on the M3A algorithm. Then, we examine the pheromone distribution and demonstrate its convergence to the stationary distribution and its capability in recovering a linear manifold. Finally, we study the performance of the algorithm on non-linear manifolds under the assumption that they can be approximated by local linear patches. We concentrate our analysis on a single patch and consider the other patches as conceptually grouped in a single state representing “the outside”.

2.5.1 Spectral Analysis

An important subject in perturbation theory is to study the effect of noise on the eigenvalues and eigenvectors of matrices (Kaslovsky and Meyer 2014). Since we use principal directions of co-variance matrices to define the MCs, it is important to

know the effect of noise on the eigenvectors of co-variance matrices. Let $\mathcal{M}$ be a d -dimensional vector subspace embedded in a higher dimensional space⁷. In the case of noise-free samples it can be exactly recovered via PCA. However, in the presence of noise the recovered subspace $\hat{\mathcal{M}}$ is perturbed. Here, the goal is to see the impact of noise on $\hat{\mathcal{M}}$.

Without loss of generality, let us consider the first d coordinates span the subspace $\mathcal{M}$. We assume the set $\{\mathbf{l}_i\}_{i=1}^N$ contains N realizations of $\mathcal{M}$, such that their first d coordinates $(l_i^{(1)}, \dots, l_i^{(d)})$ are uniformly distributed within $B_r(\mathbf{0})$. Thus, every point $\mathbf{l}_i$ on the subspace $\mathcal{M}$ has the form $(l_i^{(1)}, \dots, l_i^{(d)}, 0, \dots, 0)$. Now, suppose these realizations are disrupted via Gaussian noise with mean zero and standard deviation σI_D . If $\mathbf{x}_i$ denotes a noisy observation, then we have:

$$\mathbf{x}_i = \mathbf{l}_i + \mathbf{e}_i ,$$

where $\mathbf{e}_i$ is the noise vector. In this setting, the design matrix X can be described as:

$$X = L + E , \tag{2.16}$$

where the columns of L and E keep the noise-free realizations and noise vectors, respectively. Let us denote the centered version of a matrix $H = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N]$ as:

$$\tilde{H} = H - \mu_h \mathbf{1}_N ,$$

where $\mu_h = \frac{1}{N} \sum_i \mathbf{h}_i$ and $\mathbf{1}_N = [1, 1, \dots, 1]$ with N entries. Then the data co-variance matrix can be written as

$$\frac{1}{N} \tilde{X} \tilde{X}^T = \frac{1}{N} \tilde{L} \tilde{L}^T + \Delta$$

with the matrix Δ representing the perturbation caused by the noise:

$$\Delta = \frac{1}{N} (\tilde{L} \tilde{E}^T + \tilde{E} \tilde{L}^T + \tilde{E} \tilde{E}^T) .$$

From Eigen-decomposition, we obtain:

$$\frac{1}{N} \tilde{L} \tilde{L}^T = U \Lambda U^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \end{bmatrix}^T ,$$

⁷The assumptions are made since we only focus on linear manifolds in the next subsection. For a more general case, i.e. non-linear manifold, we refer to (Kaslovsky and Meyer 2014) where the effect of curvature on the covariance matrices is studied.

where Λ_1 is a $d \times d$ diagonal matrix containing non-zero eigenvalues in descending order, and U_1 includes their corresponding eigenvectors. Note that U_2 can be any orthogonal basis for the last $D - d$ coordinates. Similarly, we can write:

$$\frac{1}{N} \tilde{X} \tilde{X}^T = \hat{U} \Lambda \hat{U}^T = \begin{bmatrix} \hat{U}_1 & \hat{U}_2 \end{bmatrix} \begin{bmatrix} \hat{\Lambda}_1 & 0 \\ 0 & \hat{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \hat{U}_1 & \hat{U}_2 \end{bmatrix}^T.$$

The subspace $\mathcal{M}$ is spanned by U_1 and the recovered subspace $\hat{\mathcal{M}}$ is spanned by $\hat{U}_1$. The orthogonal projectors onto $\mathcal{M}$ and $\hat{\mathcal{M}}$ are derived as follow:

$$Q = U_1 U_1^T \quad \hat{Q} = \hat{U}_1 \hat{U}_1^T.$$

Here, the Frobenius distance $\|Q - \hat{Q}\|_F$ is used to compare two subspaces because it corresponds to the sum of the squared sines of the principal angles between $\mathcal{M}$ and $\hat{\mathcal{M}}$ (Kaslovsky and Meyer 2014). If the number of samples N and the probability constants ε and ε_λ satisfy the following inequalities:

$$N > 4 \left(\max(\sqrt{d}, \sqrt{D-d}) + \varepsilon \right), \quad \varepsilon < 0.7\sqrt{d(D-d)}, \quad \varepsilon_\lambda < \frac{3}{\sqrt{d+2}}\sqrt{N} \quad (2.17)$$

then the following theorem offers a bound on the angle between $\mathcal{M}$ and $\hat{\mathcal{M}}$.

Theorem 4. (Kaslovsky and Meyer 2014) Let

$$\delta = \frac{r^2}{d+2} \left(1 - \frac{1}{\sqrt{N}} \zeta_1(\varepsilon_\lambda) \right) - \sigma \frac{1}{\sqrt{N}} \zeta_2(\varepsilon_\lambda) - \sigma^2 \left(\sqrt{d} + \sqrt{D-d} + \frac{1}{\sqrt{N}} \zeta_3(\varepsilon) \right)$$

and

$$\beta = \frac{1}{\sqrt{N}} \left[\sigma \sqrt{d(D-d)} \eta(\varepsilon, \varepsilon_\lambda) + \frac{1}{\sqrt{N}} \zeta_{\text{numer}}(\varepsilon) \right].$$

Additionally, if the following conditions hold:

- (Condition 1) $\delta > 0$,
- (Condition 2) $\beta < \frac{1}{2}\delta$

then

$$\|Q - \hat{Q}\| \leq \frac{2\sqrt{2}\beta}{\delta} \quad (2.18)$$

with probability greater than

$$1 - 2de^{-\varepsilon_\lambda^2} - 9e^{-\varepsilon^2} \quad (2.19)$$

over the joint random selection of the sample points and random realization of the noise, where the following definitions have been made to ease the presentation:

$$\begin{aligned}
\zeta_1(\varepsilon_\lambda) &= \frac{2}{\sqrt{N}} - \frac{1}{N^{\frac{3}{2}}} + (1 - \frac{1}{N})\varepsilon_\lambda\sqrt{8(d+2)} , \\
\zeta_2(\varepsilon, \varepsilon_\lambda) &= \frac{2rd}{\sqrt{d+2}} \left(1 + \varepsilon_\lambda \frac{5\sqrt{d+2}}{\sqrt{N}} \right) \left(1 + \frac{6\varepsilon}{5d} \right) , \\
\zeta_3(\varepsilon) &= \frac{5}{2} \left(\sqrt{d} + \varepsilon\sqrt{2} \right) \left(\sqrt{D-d} + \varepsilon\sqrt{2} \right) , \\
\eta(\varepsilon, \varepsilon_\lambda) &= \left(1 + \frac{6}{5} \frac{\varepsilon}{\sqrt{d(D-d)}} \right) \left[\sigma + \frac{r}{\sqrt{d+2}} \left(1 + \varepsilon_\lambda \frac{5\sqrt{d+2}}{\sqrt{N}} \right) \right] , \\
\zeta_{\text{numer}}(\varepsilon) &= \sigma^2 \sqrt{d(D-2)} \left(1 + \frac{6}{5} \frac{\varepsilon}{\sqrt{d(D-d)}} \right) \left(\sqrt{D-d} + \varepsilon\sqrt{2} \right) .
\end{aligned}$$

This theorem provides an upper bound for the difference between the true manifold $\mathcal{M}$ and the recovered subspace $\hat{\mathcal{M}}$. Therefore, it is safe to say that the definition of weights, in Eq. (2.11), is reasonable in keeping a random walker close to the $\mathcal{M}$ (via encouraging to stay close to $\hat{\mathcal{M}}$). Moreover, the above theorem can help us to explain the relation between the quality of the recovered subspace $\hat{\mathcal{M}}$ and the hyper-parameters N , σ and r . An empirical example demonstrates the relationship in practice. We generate samples from a noisy one-dimensional linear manifold embedded in $\mathbb{R}^{10}$ for various (hyper-) parameter settings. Their influence is demonstrated in Figure 2.2. Panel (a) shows that the approximation performance increases with a growing number of samples N . On the other hand, there is a direct connection between the noise level σ and the approximation error as shown in panel (b). Moreover, Figure 2.2 (c) demonstrates that a bigger neighborhood radius r recovers an increasingly accurate subspace. However, this is not the case for non-linear manifolds where the curvature prevents using big radius (see 2.5 c)⁸.

2.5.2 Pheromone distribution for a linear manifold

The proposed M3A algorithm associates a pheromone value with each sample. In this section, we show that the pheromone values encode distances of the samples to the reconstructed subspace $\hat{\mathcal{M}}$. Let the data set fulfill the conditions explained in the previous subsection, i.e. the noisy data points are uniformly distributed along a linear manifold inside $B_r(\mathbf{0})$. In order to simplify our analysis we make the following assumptions in constructing the MC:

⁸In section 2.6.1, a method is proposed to pick the best radius value.

Figure 2.2: The effect of (hyper-)parameters on the quality of the recovered subspaces: a) the effect of the number of samples N within the neighborhood ($\sigma = 0.01$, $r = 1$), b) the role of the noise level σ ($N = 1000$, $r = 1$), and c) the impact of neighborhood radius r ($N = 1000$, $\sigma = 0.01$).

- The radius r is big enough such that all pairs of samples are neighbors and any sample belongs to its neighborhood. Therefore, instead of using local PCA for each point we apply PCA on the whole data set.
- In Eq. (2.10) we set x' to the origin 0 . Thus, for any point x_i its distance to $\hat{\mathcal{M}}$ is:

$$\delta_i^{\mathcal{M}} = \|(I - \hat{U}_1 \hat{U}_1^T)x_i\| ,$$

and with Eq. (2.11) the weight values are computed as:

$$w_{ki} = 1 - \frac{\delta_i^{\mathcal{M}}}{\alpha} = w_i \quad 1 \leq k \leq N . \quad (2.20)$$

We define $w_i = w_{ki}$ since w_{ki} does not depend on k .

- We remove γ from the transition probabilities resulting in:

$$p_{ji}^{(t+1)} = \frac{w_i f_i^{(t)}}{\sum_k w_k f_k^{(t)}} \quad (2.21)$$

- The pheromone values are uniformly initialized by:

$$f_i^{(0)} = \frac{1}{N} \quad \forall i . \quad (2.22)$$

Note that since the pheromone values are multiplied by weight values (see Eq. 2.21), we do not initialize them with zero.

- In the updating rule of pheromone Eq. (2.15) we set the constant c to the evaporation rate ρ :

$$f_i^{(t+1)} = \frac{\rho}{M} \sum_{a=1}^M \frac{N_i^a(n)}{n+1} + (1-\rho)f_i^{(t)} . \quad (2.23)$$

Without loss of generality we re-label the data points by $\{1, \dots, N\}$ such that

$$w_1 \geq w_2 \geq \dots \geq w_N . \quad (2.24)$$

Thus, the new labels sort samples according to their closeness to the subspace $\hat{\mathcal{M}}$.

Definition 2.5.1. The pheromone values across the samples form a probability distribution called *pheromone distribution*, denoted by $\mathbf{f}^{(t)} = [f_1^{(t)}, f_2^{(t)}, \dots, f_N^{(t)}]$.

In this setting we show that:

- the pheromone values on sample points are sorted by their distances to the subspace $\hat{\mathcal{M}}$, and
- the pheromone distribution $\mathbf{f}^{(t)}$ converges as $t \rightarrow \infty$.

Our analysis is based on the Markov Chain framework and we consider two cases. First, we assume there is only one ant who walks for an unbounded number of steps in every round (i.e. $n \rightarrow \infty$). Second, we assume an unbounded number of ants (i.e. $M \rightarrow \infty$) who walk for n steps each in every round. While the former gives enough time to a single walker to explore the data set, the latter employs many agents to search in the space in a limited time. In other words, we study the long-term and the short-term behavior of random walks, respectively.

Single ant, unbounded path length

We assume a single ant (i.e. $M = 1$) performing $n \rightarrow \infty$ steps in each round. From theorem 3, the long-term fraction of time spending in a point x_i is equal to its stationary distribution value π_i . Therefore, for any point x_i , its pheromone value is updated according to:

$$\begin{aligned} f_i^{(t+1)} &= \frac{\rho}{M} \sum_{a=1}^M \pi_i^{(t+1)} + (1-\rho)f_i^{(t)} \\ &= (1-\rho)f_i^{(t)} + \rho \cdot \pi_i^{(t+1)} , \end{aligned} \quad (2.25)$$

where $\pi^{(t+1)}$ is the stationary distribution associated to the transition probability matrix in the $(t + 1)$ -th round. The stationary distribution can be computed via the balance equation:

$$\begin{aligned}
\pi_i^{(t+1)} &= \sum_j \pi_j^{(t+1)} p_{ji}^{(t+1)} \\
&= \sum_j \pi_j^{(t+1)} \frac{w_i f_i^{(t)}}{\sum_k w_k f_k^{(t)}} \\
&= \frac{w_i f_i^{(t)}}{\sum_k w_k f_k^{(t)}} \\
&= p_{ji}^{(t+1)} \quad \forall j .
\end{aligned} \tag{2.26}$$

From (2.22) and (2.24), it can be shown (by induction) for any $t > 0$:

$$\pi_1^{(t)} \geq \pi_2^{(t)} \geq \dots \geq \pi_N^{(t)} \tag{2.27}$$

and by (2.25):

$$f_1^{(t)} \geq f_2^{(t)} \geq \dots \geq f_N^{(t)} . \tag{2.28}$$

Since the weight values encode the distance of samples to the subspace $\hat{\mathcal{M}}$, see Eq. (2.11), the following corollary holds.

Corollary 4.1. *The pheromone values are sorted according to their closeness to the linear manifold $\hat{\mathcal{M}}$.*

In the next step, we would like to prove the convergence of the pheromone distribution, but we need to first establish the following lemma:

Lemma 5. The sequence $\{a^{(t)}\}_{t=0}^{\infty}$ with

$$a^{(t)} = \sum_k w_k f_k^{(t)} \tag{2.29}$$

- is monotonically increasing (i.e. $a^{(t+1)} \geq a^{(t)}$) and
- is convergent (i.e. $a^{(t)} \rightarrow a$ for some $a \geq 0$).

Proof. a) To demonstrate that the sequence $\{a^{(t)}\}_{t=0}^{\infty}$ is monotonically increasing, we need to show:

$$I^{(t)} = a^{(t+1)} - a^{(t)} = \sum_k w_k \cdot \Delta^{(t)} f_k \geq 0 ,$$

where $\Delta^{(t)} f_i = f_i^{(t+1)} - f_i^{(t)}$. We can write:

$$\begin{aligned} \Delta^{(t)} f_i &\stackrel{\text{Eq.(2.25)}}{=} \rho(\pi_i^{(t+1)} - f_i^{(t)}) \\ &\stackrel{\text{Eq.(2.26)}}{=} \rho \left(\frac{w_i}{\sum_k w_k f_k^{(t)}} - 1 \right) f_i^{(t)} \\ &= \frac{\rho}{a^{(t)}} (w_i - a^{(t)}) f_i^{(t)} . \end{aligned} \quad (2.30)$$

For every fixed t we have: $\{f_i^{(t)}\}_{i=1}^N$ and $\{w_i - a^{(t)}\}_{i=1}^N$ ⁹ that are monotonically decreasing sequences. Thus, the sequence $\{\Delta^{(t)} f_i\}_{i=1}^N$ will be decreasing as well. Therefore, we have:

$$I^{(t)} = \sum_{k \leq k^*} w_k \cdot \overbrace{\Delta^{(t)} f_k}^{\geq 0} + \sum_{k > k^*} w_k \cdot \overbrace{\Delta^{(t)} f_k}^{< 0}$$

where $k^* = \max\{k : \Delta^{(t)} f_k \geq 0\}$ and we define the following values:

$$w = \min_{k \leq k^*} w_k \quad , \quad w' = \max_{k > k^*} w_k .$$

Since $\{w_i\}_{i=1}^N$ is decreasing we have $w > w'$:

$$\begin{aligned} I^{(t)} &\geq w \cdot \sum_{k \leq k^*} \overbrace{\Delta^{(t)} f_k}^{\geq 0} + w' \cdot \sum_{k > k^*} \overbrace{\Delta^{(t)} f_k}^{< 0} \\ &\stackrel{w > w'}{>} w \cdot \sum_{k \leq k^*} \Delta^{(t)} f_k + w \cdot \sum_{k > k^*} \Delta^{(t)} f_k \\ &= w \cdot \sum_k \Delta^{(t)} f_k = w \cdot \sum_k (f_k^{(t+1)} - f_k^{(t)}) \\ &= w \cdot \left(\sum_k f_k^{(t+1)} - \sum_k f_k^{(t)} \right) = 0 . \end{aligned}$$

b) We know that:

$$0 \leq w_i \leq 1, \quad 0 \leq f_i^{(t)} \leq 1$$

and therefore

$$0 \leq a^{(t)} \leq N .$$

Since $a^{(t)}$ is an increasing and bounded sequence it is convergent. $\square$

⁹From inequality (2.24) we know that $\{w_i\}_{i=1}^N$ is a decreasing sequence. Therefore, the sequence $\{w_i - a^{(t)}\}_{i=1}^N$ is also decreasing, since $a^{(t)}$ is a constant number for a fixed t .

The following theorem states that the pheromone values are convergent over time.

Theorem 6. *For any point x_i , its pheromone value $f_i^{(t)}$ is convergent as $t \rightarrow \infty$.*

Proof. Since we know that the pheromone values $f_i^{(t)}$ are bounded from below and above for all t , it is enough to show that $\{f_i^{(t)}\}_{t=1}^{\infty}$ is a monotonic sequence. From Eq. (2.30) we just need to show that $w_i - a^{(t)}$ becomes only positive or only negative after some round T . Let us define the sequence $\{b^{(t)}\}$ as $b^{(t)} = w_i - a^{(t)}$. From lemma 5 we know sequence $a^{(t)}$ is increasing and $a^{(t)} \rightarrow a$ (as $t \rightarrow \infty$). Thus, $b^{(t)}$ will be a decreasing sequence and convergent to $w_i - a$:

- If $w_i \geq a$, then $b^{(t)} \geq 0$ for any t . Thus, from Eq. (2.30) we get $\Delta^{(t)} f_i \geq 0$ and then $f_i^{(t)}$ is a monotonically increasing sequence.
- If $w_i < a$ then there exists $T \in \mathbb{N}$ such that

$$\forall t > T : a^{(t)} > w_i \implies b^{(t)} < 0 ,$$

so $f_i^{(t)}$ is a decreasing sequence for $t > T$.

□

Corollary 6.1. *From the convergence of pheromone values and the updating rule in Eq. (2.25) it is clear that $\pi_i^{(t)}$ is convergent (as $t \rightarrow \infty$) for any $i \in \{1, 2, \dots, N\}$.*

In addition to the convergence of stationary and pheromone distributions the following theorem shows that the pheromone distribution converges to the stationary distribution of the MC.

Theorem 7. *The pheromone distribution $f^{(t)}$ converges to the stationary distribution $\pi^{(t)}$ as $t \rightarrow \infty$, i.e. for any $\epsilon > 0$, there exists $T \in \mathbb{N}$ such that*

$$\forall i \in \{1, \dots, N\}, \forall t > T \implies |\pi_i^{(t)} - f_i^{(t)}| < \epsilon .$$

Proof. Theorem 6 states that the pheromone distribution converges (i.e. $\forall i \ f_i^{(t)} \rightarrow f_i$ as $t \rightarrow \infty$). Therefore, the update rule can be written as follows:

$$f_i = (1 - \rho)f_i + \rho \cdot \lim_{t \rightarrow \infty} \pi_i^{(t)} \implies \lim_{t \rightarrow \infty} \pi_i^{(t)} = f_i .$$

Thus, the pheromone and stationary distribution converge towards the same distribution.

□

Figure 2.3: 1D manifold consisting of 2000 data points disrupted by Gaussian noise ($N(0, 0.1)$) depicted as gray points in panel a and b. The 120 points with highest density (using a Gaussian kernel) and pheromone values (resulting from Eq. (2.20)) are highlighted in red in a) and b) respectively. The development of the pheromone values over iterations t for points close to the manifold is steeply increasing (as shown in panel c), while it is abruptly dropping to zero for points far away (see panel d). Panel e) shows the evolution of the mean value of weights $a^{(t)}$ (see Eq. 2.29) and f) the KL-divergence from the stationary to the pheromone distribution over the iterations t .

In order to demonstrate the above results with an example we generate a 1D manifold such that 2000 data points are uniformly distributed on a line and disrupted by Gaussian noise $N(0, 0.1)$ ¹⁰ (gray points as shown in Figure 2.3a and b). In panel (a) the Gaussian kernel with $\sigma = 1$ and $r = 0.1$ ¹¹ is used to detect 120 samples with the highest density highlighted in red. Panel (b) shows the samples with the highest pheromone values using the weights defined in Eq. (2.20). With the Gaus-

¹⁰Other noise models, such as uniform noise (“tubes” around the manifold) or the Laplace distribution (strongly concentrating points along the manifold) can be alternative scenarios dependent on the process with which the manifolds are sampled. However, here we concentrate on Gaussian noise that is suitable for our applications to keep the paper more concise.

¹¹Note that we select r such that the red points are more distributed along the manifold. For bigger value for r , the red points are more concentrated around $x = 0$ and fail to recover the 1D manifold.

sian kernel failing we clearly see the superiority of the new algorithm in recovering the linear manifold nearly perfectly, since it uses PCA in the creation of the MC. Panels (c) and (d) depict the evolution of pheromone values of samples close to and far from the underlying 1D manifold, respectively. Closer samples to the manifold exhibiting lower noise levels receive more and more visits in smaller time intervals (from theorem 3) and thus accumulate more and more pheromone over time. Panel (e) displays that the mean of weight values $a^{(t)} = \sum_k w_k f_k$ is monotonic increasing and convergent, as explained in Lemma 5. Finally, panel (f) uses the Kullback Leibler (KL) divergence to show that the pheromone distribution converges towards the stationary distribution, confirming theorem 7.

Unbounded number of ants, fixed bounded path length

Since our algorithm allows employing multiple ants we study its performance in the presence of a group of ants, which walk for a limited number of steps n in every round. Let M denote the number of ants and $N_j(n)$ represents the number of times a random walker visits point $\mathbf{x}_j$ during n steps on $(t + 1)$ -th round. For a specific case $M = 1$ we have (according to Definition 2.2.1):

$$E[N_j(n)] = E_{P(X_0=\mathbf{x}_i)}[m_{ij}(n)] \quad (2.31)$$

and assuming uniform distribution over the initial states (sample points),

$$E[N_j(n)] = \frac{1}{N} \sum_{i=1}^N m_{ij}(n) . \quad (2.32)$$

For $M > 1$ we still have the above equality for each ant since pheromone values are kept fixed over one round. Hence, in each round an ant walks independently and others do not have any impact on it. Initializing each ant on one of the sample points with probability $P(X_0 = \mathbf{x}_i)$ we obtain by the Law of Large Numbers:

$$\frac{\sum_{a=1}^M N_j^a(n)}{M} \rightarrow E[N_j(n)], \quad \text{as } M \rightarrow \infty . \quad (2.33)$$

As the number of ants M goes to infinity and from Eq. (2.32) and (2.33) the updating rule, in Eq. (2.23), can be re-written:

$$f_j^{(t+1)} = (1 - \rho) f_j^{(t)} + \frac{\rho}{N} \sum_{i=1}^N \frac{m_{ij}(n)}{n + 1} . \quad (2.34)$$

The following corollary says that, like the weight values, the pheromone values form a decreasing sequence.

Corollary 7.1. *For each t the sequence of pheromone values $\{f_i^{(t)}\}_{i=1}^N$ is monotonically decreasing.*

Proof. We use induction on t to show that the following statement holds for any arbitrary pair of points x_i and x_j with $i < j$:

$$f_i^{(t)} \geq f_j^{(t)} . \quad (2.35)$$

For simplicity we drop the exponent t in the following computations. In the first round $t = 1$ we have:

$$f_i^{(0)} = f_j^{(0)} = \frac{1}{N} \xrightarrow{w_i \geq w_j} p_{ki} \geq p_{kj} \quad \forall k ,$$

and

$$p_{ki}^{(2)} = \sum_l p_{kl} p_{li} \geq \sum_l p_{kl} p_{lj} = p_{kj}^{(2)} ,$$

where $p_{ki}^{(2)}$ is the (k, i) -the element of the second power of the transition matrix P . Similarly, it can be generalized for any q :

$$p_{ki}^{(q)} \geq p_{kj}^{(q)} \quad \forall k .$$

As a result of theorem 2, we derive:

$$m_{ki}(n) \geq m_{kj}(n) \quad \forall k .$$

Thus, the updating rule (see Eq. (2.34)) shows that the inequality (2.35) holds in the first round. Now, let the statement hold up to t -th round (i.e. $f_i^{(t)} \geq f_j^{(t)}$). In the $(t + 1)$ -th round we have:

$$p_{ki} = \frac{w_i f_i}{\sum w_k f_k} \xrightarrow{w_i \geq w_j} \frac{w_j f_j}{\sum w_k f_k} = p_{kj} .$$

Similar to the first round, we obtain for any q :

$$\forall k \quad p_{ki}^{(q)} \geq p_{kj}^{(q)} \implies m_{ki}(n) \geq m_{kj}(n) \implies f_i^{(t+1)} \geq f_j^{(t+1)}$$

□

Therefore, the pheromone values, similar to weights, encode information about how close the individual sample points are to the subspace $\hat{\mathcal{M}}$. Analogously to Theorems 6 and 7 the following theorem guarantees the convergence of the pheromone and the stationary distributions.

Theorem 8. *The following statements hold for the pheromone distribution:*

- (a) *For any point $\mathbf{x}_i$ its pheromone values $f_i^{(t)}$ converge in t (i.e. $f_i^{(t)} \rightarrow f_i$ as $t \rightarrow \infty$).*
- (b) *The pheromone distribution $\mathbf{f}^{(t)}$ converges to the stationary distribution $\pi^{(t)}$ as $t \rightarrow \infty$, i.e. for any $\epsilon > 0$ there exists $T \in \mathbb{N}$ such that:*

$$\forall i \in \{1, \dots, N\}, \forall t > T \rightarrow |\pi_i^{(t)} - f_i^{(t)}| < \epsilon .$$

Proof. The proof of (a) follows that of theorem 6.

b) From the convergence of pheromone values and the updating rule (2.34) we can write:

$$f_i = \frac{1}{N(n+1)} \sum_{i=1}^N \lim_{t \rightarrow \infty} m_{ji}^{(t)}(n) .$$

From (2.26) we know :

$$\forall i, j; \quad p_{ji}^{(t)} = \pi_i^{(t)} \implies p_{ji}^{(t)(q)} = \pi_i^{(t)} \quad \forall q , \quad (2.36)$$

where $p_{ji}^{(t)(q)}$ is the (j, i) -th element of $(P^{(t)})^q$ in the t -th round. Therefore, we have

$$\begin{aligned} f_i &= \lim_{t \rightarrow \infty} \left(\frac{1}{N(n+1)} \sum_{i=1}^N m_{ji}^{(t)}(n) \right) \stackrel{\text{theo. (2)}}{=} \lim_{t \rightarrow \infty} \left(\frac{1}{N(n+1)} \sum_{i=1}^N \sum_{q=0}^n p_{ji}^{(t)(q)} \right) \\ &\stackrel{\text{Eq. (2.36)}}{=} \lim_{t \rightarrow \infty} \frac{1}{N(n+1)} \sum_{i=1}^N \sum_{q=0}^n \pi_i^{(t)} = \lim_{t \rightarrow \infty} \pi_i^{(t)} . \end{aligned}$$

□

Hence, it can be said that both scenarios ($n \rightarrow \infty$ or $M \rightarrow \infty$) result in the same pheromone structure and asymptotic:

- pheromone values are sorted according to their closeness to the recovered linear manifold, and finally
- they are convergent to the stationary distribution of the Markov Chain.

2.5.3 Pheromone distribution for a non-linear manifold

For non-linear manifolds we consider a simplified scenario for our analysis. Although it is slightly different from the algorithm's definition, we use it as an example to explain the success of M3A in recovering non-linear manifolds. Motivated

Figure 2.4: a) approximation of a piece of non-linear manifolds with a linear patch. b) the linear patch with an external node (o), simulating points outside the neighborhood.

by the assumption that a non-linear manifold can locally be approximated by linear patches (see Fig. 2.4 a), we focus on a small part of the manifold (red circle), and we model all points outside via a single state called o . Therefore, we assume that the random walker can jump from any point inside the red circle to o with a probability proportional to the fixed-number ν and vice versa with a probability proportional to the fixed-number ω (see Fig. 2.4 b). Consequently, the transition probabilities are defined as:

$$p'_{oj}(t+1) = \begin{cases} \frac{\omega}{N} & \text{if } j \neq o \\ 1 - \omega & \text{if } j = o \end{cases} \quad \text{and} \quad p'_{i(\neq o)j}(t+1) = \begin{cases} \frac{\nu}{\sum_{k \neq o} w_k f_k^{(t)} + \nu} & \text{if } j = o \\ \frac{w_j f_j^{(t)}}{\sum_{k \neq o} w_k f_k^{(t)} + \nu} & \text{if } j \neq o \end{cases},$$

where N is the number of data points inside the red circle. Note that in order to differentiate the transition probabilities and the stationary distribution in this section from the previous one, we use the prime symbol ($'$).

From the balance equation we can compute the stationary distribution. Thus, for any state j ($\neq o$) we have:

$$\begin{aligned}
\pi_j'^{(t+1)} &= \sum_{i \neq \mathbf{o}} \pi_i'^{(t+1)} p_{ij}^{(t+1)} + \pi_{\mathbf{o}}'^{(t+1)} p_{\mathbf{o}j}^{(t+1)} \\
&= \sum_{i \neq \mathbf{o}} \pi_i'^{(t+1)} \frac{w_j f_j^{(t)}}{\sum_k w_k f_k^{(t)} + \nu} + \pi_{\mathbf{o}}'^{(t+1)} \frac{\omega}{N} \\
&= \frac{w_j f_j^{(t)}}{\sum_k w_k f_k^{(t)} + \nu} \sum_{i \neq \mathbf{o}} \pi_i'^{(t+1)} + \pi_{\mathbf{o}}'^{(t+1)} \frac{\omega}{N} \\
&= \left(\frac{w_j f_j^{(t)}}{\sum_k w_k f_k^{(t)}} \right) \left(\frac{\sum_k w_k f_k^{(t)}}{\sum_k w_k f_k^{(t)} + \nu} \right) (1 - \pi_{\mathbf{o}}'^{(t+1)}) + \pi_{\mathbf{o}}'^{(t+1)} \frac{\omega}{N}, \quad (2.37)
\end{aligned}$$

In the previous section we found $p_{ij}^{(t+1)} = \frac{w_j f_j^{(t)}}{\sum_k w_k f_k^{(t)}}$ and hence we can re-write the terms above as follows:

$$\begin{aligned}
\pi_j'^{(t+1)} &= p_{ij}^{(t+1)} \left(\frac{\sum_k w_k f_k^{(t)}}{\sum_k w_k f_k^{(t)} + \nu} \right) (1 - \pi_{\mathbf{o}}'^{(t+1)}) + \pi_{\mathbf{o}}'^{(t+1)} \frac{\omega}{N} \\
&= a p_{ij}^{(t+1)} + b, \quad (2.38)
\end{aligned}$$

where

$$a = \left(\frac{\sum_k w_k f_k^{(t)}}{\sum_k w_k f_k^{(t)} + \nu} \right) (1 - \pi_{\mathbf{o}}'^{(t+1)}) > 0$$

and

$$b = \pi_{\mathbf{o}}'^{(t+1)} \frac{\omega}{N} > 0.$$

From Eq. (2.26) we obtain:

$$\pi_j'^{(t+1)} = a \pi_j^{(t+1)} + b, \quad (2.39)$$

where $\pi_j^{(t+1)}$ denotes the stationary distribution of the random walk in the previous section. Thus, this equation connects the stationary distributions of the new MC π' to π in the previous section. In other words, the new distribution is monotonically increasing with respect to the previous one.

The theorem 3 connects the long-term behavior of a random walk to its stationary distribution. From (2.27) and (2.39) we see that the stationary distribution (and similarly the pheromone distribution) is sorted according to the distances of data

points (inside the neighborhood) to the linear patch constructed by PCA. From theorem 2, we know that the occupancy time (i.e. the short-term behavior) is directly connected to the transition probabilities of the MC. For any i and j ($\neq o$), we have:

$$p_{ij}^{(t+1)} = \frac{w_j f_j}{\sum_k w_k f_k + \nu} = \left(\frac{w_j f_j}{\sum_k w_k f_k} \right) \left(\frac{\sum_k w_k f_k}{\sum_k w_k f_k + \nu} \right) = ap_{ij}^{(t+1)}$$

Thus, the new transition probabilities are monotonic increasing function of the previous transition probabilities. Therefore, similar results as in section 2.5.2 hold.

By simplifying the scenario, in this section, we show that the algorithm sorts data points according to their distances to the linear patches (as estimators for a manifold). From (Little et al. 2017), we can conclude that the performance of the new algorithm depends on several factors including selecting an appropriate neighborhood size r . Note that r should be big enough to include a sufficient number of samples for recovering tangent spaces, and small enough to prevent high-curvature. Therefore, in the presence of an appropriate r value, our analysis shows that the pheromone distribution contains valuable information in order to highlight manifolds.

2.6 Experiments

In this section we use synthetic and real data sets to investigate the performance of M3A in different scenarios. At first we discuss the influence and specification of its hyper-parameters. Furthermore, complementary to our analysis in section 2.5.1 we empirically examine the impact of the new MC formulations for denoising and visualization of non-linear manifolds. Moreover, we show how the proposed algorithm helps to discover manifolds and to build better probabilistic models in the sense of sparseness, descriptiveness, and preservation of structural details.

2.6.1 Strategy for automatic hyper-parameter selection

Our analysis in section 2.5 shows that the neighborhood size r , the number of ants M , and the number of steps n in one round play major roles for the performance of M3A. Thus, an automatic strategy to find appropriate values for these quantities is highly desirable. In the following we present such strategies for practical application.

In subsection 2.5.1, we demonstrate that a proper value for r depends on the noise level σ of the manifold. We observe that a bigger r helps to reduce the effect of the noise and achieve a better approximation for the linear manifold. However, as it is shown in (Kaslovsky and Meyer 2014) and (Little et al. 2017) for non-linear

manifolds, the presence of curvature encourages to use a smaller radius r , such that the manifold looks almost linear within the local neighborhood. Hence, the selection of r is a trade-off between the noise level and curvature of the manifolds. Here, to specify a suitable value for r we use a mixture of Gaussian models. First, we place the centers of Gaussian distributions on points with higher pheromone values, and then their co-variance matrices are computed based on the local neighbors. Finally, we use the likelihood function to determine the best r (see Figure 2.5c).

In subsection 2.5.2 we investigate two strategies: a) using a few ants walking many steps in every round, and b) using many ants walking a limited number of steps. As a rule of thumb to determine a suitable number of steps and ants (n and M) we assume they fulfill the following inequality:

$$n \cdot M \geq z \cdot N \quad (2.40)$$

for a constant (big) z . Thus, one may use a small number of workers M (or processors/cores) with a big number of steps n or use many workers with a small number of steps in each round. In our experimentation, we set the number of steps and the number of ants to $n = N$ and $M = z = 50$, respectively ¹².

The pheromone determines how much the ants are reinforced to prefer points frequently visited before. And the evaporation rate ρ and parameter γ control the impact of the pheromone on the random walk process, and thus their determination is connected to M and n . If M and n are both small, it means the algorithm has neither enough ants nor enough time to highlight manifolds; thus, our analysis in subsection 2.5.2 is not valid any more. In this case, we suggest using smaller ρ and γ to prevent highlighting background noise. In this chapter, we set them to $\rho = \gamma = 0.1$ for all the experiments.

Furthermore, the maximum amount of pheromone c added on a sample per round is fixed to 2^{13} , and the factor α for the weight w_d , in Eq. (2.11), is determined such that only $p = 50\%$ of neighbors have non-zero values. Finally, for any sample $\mathbf{x}_i$, we approximate the point $\mathbf{x}'$ that touches the underlying manifold by $\mathbf{x}' = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{x}_k \in \mathcal{N}_i} \mathbf{x}_k$, (see Eq. (2.10)).

2.6.2 Homogeneous Markov Chain

In this section we extend our theoretical analysis with an empirical analysis on non-linear manifolds. First, we demonstrate the denoising capability of the MCs defined

¹²Note that the selection of z , M and n depends on the amount of available computational resources. For instance, if we do not have a restricted limitation on either time or number of processors, we can fix z to a large number. Then, based on the number of processors and their available time, M and n are set such that to fulfill Eq. (2.40).

¹³It is just saturated at 2 to prevent big changes on pheromone vector in one round.

in Eq. (2.11) and (2.12) on highlighting the underlying manifold. Second, we investigate their impact on the performance of t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton 2008), a widely used dimension reduction and visualization technique.

Denoising the manifold: To check the success of the weighting methods (section 2.3) in keeping ants close to the manifold, we need to study the short and long-term behavior of the random walk defined by Eq. (2.2). According to theorems 2 and 3 the short and long-term behaviors of MCs are directly connected to the power series of the transition probability and to the stationary distribution, respectively. Here, we use a synthetic data set to investigate the constructed MCs.

In Figure 2.5, panels (a,b) show 6000 samples generated from the s-curve manifold disrupted by Gaussian noise $N(0, 0.2)$. In order to select r we follow the strategy introduced before (in section 2.6.1). We apply M3A with different r using 20% of the samples (1200 points) with the highest pheromone values as means for Gaussian distributions and their neighbors to determine the co-variance matrices. The quality of the models is computed by the average log-likelihood (ALL) function and the results are shown in Figure 2.5c. Panels (d-l) show 1200 points with the highest stationary distribution values for different MCs. As it can be seen the MC based on the Gaussian kernel function fails to recover the manifold (see panels d and g). However, the other methods are successful in keeping random walkers close to the manifold (see panels e, f, h and l). Note that panels (e, f) also show that, since we use our knowledge about the intrinsic dimensionality of the manifold in Eq. (2.11), it gives a slightly better result than the MC defined by Eq. (2.12). A similar result is achieved for the short-term behavior of the MCs where the two formulation of Eq. (2.11) and (2.12) outperforms the Gaussian kernel function.

Manifold aligned similarities in t-SNE: A specific case of dimensionality reduction is visualization where high dimensional data are embedded to a two or three dimensions. A well-known non-linear tool to visualize high-dimensional data is t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton 2008). It transforms pairwise similarities of data points to probability distributions of the high-dimensional data as well as the low-dimensional embedding and then minimizes the Kullback-Leibler divergence between them. While the original t-SNE is based on a Gaussian kernel function Eq. (2.9) we replace it with the tangent space dissimilarity measure defined in Eq. (2.11) and (2.12). Since high dimensional data spaces are typically sparse we use a k- nearest neighbours, instead of radius neighborhood, and we set $p = 100\%$ (i.e. we normalize the distances in Eq. (2.11) by dividing them to their maximum values). Furthermore, we compare the results

Figure 2.5: 2 views of the noisy s-curve data set (a,b), radius selection via average log likelihood for different r (c), and 1200 points with highest stationary distributions values for the MCs defined by Eq. (2.11) (e,h) and Eq. (2.12) (f,l), using $r = 0.6$.

to the original t-SNE, using the Euclidean distance to compute the neighbourhood probabilities, and a t-SNE version using Mahalanobis distance instead. The latter is based on the covariance of the data and can therefore align with global directions of major variance. Therefore the locally aligned formulation, as proposed in M3A, is compared to “no alignment” and “global alignment”. We demonstrate the strategy using two real non-linear dimensionality reduction benchmark datasets: 1) COIL20: consists of images (with $32 \times 32 = 1024$ pixels) of 20 objects rotated 72 times (5 degrees per image) and 2) USPS: contains 9298 grey-scale images (with 16×16 pixels)

of handwritten digits. As a preprocessing step we apply PCA to decrease the dimensionality to 20. The different t-SNE outputs are compared and evaluated using the quality of group compactness (QGC) (Gorban and Zinovyev 2010) as a measure of how close samples from the same class remain in the embedding. Let $c(i; k)$ denote the number of points in the k -neighbourhood of x_i with the same label, then the compactness of a class is defined as:

$$QGC_k(l) = \frac{1}{k \cdot N(l)} \sum_{y_i=l} c(i; k)$$

where $N(l)$ is the number of samples with the label l . If the QGC for a class is close to 1 the class is compact and well-separated from others. Since t-SNE is non-convex it generates different results for each run and therefore we repeat the experiment 10 times and report the average quality. Figure 2.6 shows example t-SNE embeddings for COIL20 and USPS (panel a and c), accompanied by results using Eq. (2.11) with $d = 1$ and $k = 10$ and Eq. (2.12) with $k = 20$ (panel b and d). Panel e and g depict the average QGC curves for the two data sets COIL20 and USPS and the different similarity measures, respectively. It can be seen that the new weight values outperform others, especially for COIL20 if we use our knowledge about the intrinsic dimensionality of manifolds $d = 1$. For the USPS, it is clear that it does a better job for more global structures, i.e. clusters, visible for $k > 200$. Since in the COIL20 represents different viewpoints of rotated objects, one can expect that each class forms a closed loop 1D manifold. Since the new weights only consider the distance to the estimated manifold it can recover more closed loops with less distortion (see Figure 2.6 a and b). Moreover, the four objects, corresponding to four types of toy cars, cannot be separated using the original t-SNE, because it maps cars with the same orientations close to each other. As seen in Figure 2.6 (g) the new weights do a better job embedding them. Not knowing the intrinsic dimensionality of manifolds in the handwritten digit dataset we use the dimension weighted version Eq. (2.12) to define a similarity measure for the t-SNE. Although t-SNE with the Gaussian kernel function can recover most classes, a strong overlap between samples from digit 3 and 5 is visible. However, due to taking into account the shape of manifold in the neighborhood, the new weights can successfully separate digit numbers, even the numbers 3 and 5, and increase QGC measures, as seen in Figure 2.6 panels d and h. In summary, the locally aligned formulation, proposed here, improves the performance of t-SNE, in comparison to "no alignment" (i.e. Euclidean distance) and "global alignment" (i.e. Mahalanobis distance) formulations, in separating classes.

Figure 2.6: The impact of the new similarity measures on the t-SNE: Top row COIL20: a) using original t-SNE (perplexity 20) and b) using Eq. (2.11) ($d = 1, k = 10$). Middle row USPS: c) using original t-SNE (perplexity = 30) and d) using Eq. (2.12) ($k = 20$). Bottom row: average QGC measures for COIL20 and USPS (panel e and f). g) QGC curves for four classes of toy cars in COIL20 and h) of digits 3 and 5 in USPS.

2.6.3 Manifold Alignment Aware Ants (M3A)

In this section we analyze the capability of M3A in recovering structures surrounded by background noise. First, we compare it to related state-of-the-art techniques for denoising low-dimensional manifolds. And second, we demonstrate the performance of our algorithm on a real-world astronomical data set, namely the GAIA DR2 catalog (Collaboration et al. 2018), to extract stellar structures.

Extracting low-dimensional manifolds: Typically, manifold learning assumes that data points are lying on a low-dimensional manifold embedded in a higher-dimensional space. However, in practice, manifolds are disrupted by high-dimensional noise, for example stemming from equipment or the presence of samples that do not belong to the manifolds. In those cases, the goal is to extract the manifolds by removing the background noise and simultaneously suppressing the noise level on the manifolds. To compare the performance of denoising techniques we create a synthetic 1D manifold forming a circle ($r = 6$), which misses a part of its arc. Furthermore, we randomly select 3000 points from the manifold and add Gaussian noise $N(0, 0.3)$. To model background noise, 3000 samples are generated following a uniform distribution in the square $[-15, 15] \times [-10, 20]$ (see Figure 2.7a).

In the following we compare four denoising techniques to the proposed M3A. We use: a) MD with hyper-parameters settings $\delta t = 0.1$, $N_{\text{iter}} = 20$ and $k = 40$, b) MBSM with $k = 40$ and $N_{\text{iter}} = 40$, and c) LLD with $\lambda = 0.01$, $k = 40$ and $\sigma = 1$. Lastly, LLPD is deployed using a threshold of 0.15. Moreover, in order to select r in M3A we follow a two step strategy. First, we apply the algorithm using $r = 3$ to remove the background noise. Note that since the density of the background is lower than the manifold it can be done using a wide range of r . Then, we use the strategy explained in section 2.6.1 to find the best r for the remaining points.

Some techniques, such as MD, MBMS and LLD, are not designed to deal with the presence of the background noise¹⁴ and are therefore not directly suitable for the given situation, as shown in Figure 2.7. Since the Manifold Denoising (MD) algorithm pushes samples towards dense regions it has two disadvantages: first, due to the presence of background noise it creates some new dense structures in data space, and second, because of small variation on the density along the manifold the manifold becomes discontinuous after some iterations (see panel b). Since Manifold Blurring Mean Shift (MBMS) limits the movement of samples to be parallel to the manifold normals, it can prevent discontinuity on the manifold. However, it creates many artificial 1D manifolds and fails to recover the missing arc in the circle structure of interest (see panel c). Similar to MBMS, the presence of background

¹⁴Since they assume all data points belong to the manifold which is not true for the background noise.

Figure 2.7: Panel a): 1D circular manifold embedded in uniform background noise. Best results of: b) MD, c) MBMS, d) LLD, and e) LLPD (containing 2000 samples, i.e. 35%). Panel f): denoised M3A result where gray and red points show 2700 (45%) and 780 (13) samples with $r = 3$ and $r = 2$.

noise highly influences the result of the LLD algorithm, which also fails to recover the manifold (see panel d). In contrast to previous methods, LLPD considers the presence of the background noise and extracts the samples in dense regions recovering the noisy circle. Although it does not create new structures, it can not denoise the manifold. It also fragments the structure into smaller clusters (see panel e). On the other hand, our method encourages random walkers to stay close to the manifold and overcomes all three problems: 1) it recovers the circle from the background noise, 2) it preserves the missing part in the arc without creating artificial structures, and 3) it reduces the manifold noise by subsampling based on the highest pheromone level (see panel f).

Detecting dense structures in real world data: An important task in astronomy is to extract stellar structures, such as galaxies and globular clusters. Here, we apply M3A to a part of the GAIA DR2 catalog (Collaboration et al. 2018) containing

1,071,714 light sources. The data set contains seven features, including five denoting positional information (right ascension, declination, motion along right ascension and declination, and parallax), as well as photometric information, that is G-band magnitude and (B-R) color. As a preprocessing step, we normalize the data set and use k nearest neighbor (with $k = 20$) to speed up the computation. First, we use the positional information to detect stellar structures. Figure 2.8a shows that M3A detects five of the known globular clusters (GC): 1) NGC4147 2) NGC5024 3) NGC5053 4) NGC5272 5) NGC5466. Second, we would like to cluster the members of a GC. To extract stars of a GC, we use objects which are not further than 0.05 degree from the center of the GC. Then, we use their photometric information since it reveals at which stage of life a star is (for more details see (Mohammadi et al. 2019)). Panels b and d display the clusters detected by spectral clustering using a Gaussian kernel (Von Luxburg 2007). However, the resulting clusters do not correlate to any astrophysical meaning. If M3A is applied on the extracted light sources, we obtain the transition probability matrix P . After symmetrization $P' = \frac{1}{2}(P + P^T)$, the spectral clustering can be implemented based on P' . From panels c and e, we see that the resulting clusters coincide with the three known groups called: Main Sequence (MS), Horizontal Branch (HB) and Red Giant Branch (RGB) (see (Mohammadi et al. 2019)). It can be seen that MS is separable since it is a 2D manifold while HB and RGB are 1D manifolds. Moreover, since HB is almost a line perpendicular to RGB, the weight values between these two clusters are small, and as a result, they are separable. In summary, M3A provides: 1) a pheromone distribution that can be used to find stellar structures, and 2) a transition probability matrix that can be used to group stars according to the stage of their life.

2.6.4 Improved density estimation

A common task in machine learning is to estimate the underlying probability density function (pdf) given a limited number of data points. A typical way to model it is to use a finite mixture model

$$\hat{f}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^K p(k) \hat{f}(\mathbf{x}; \theta_k) ,$$

where $p(k)$ and θ_k are the mixture weight and the parameter values of the k -th component, respectively, while $\boldsymbol{\theta}$ denotes the set of all parameters. If the number of models (K) is much smaller than the number of samples (N) it provides a sparse representation for the data set. A well-known algorithm is the Gaussian Mixture Model (GMM) (Bishop 2006a), which uses a mixture of full-rank Gaussian distributions to model the pdf. Its parameters are learned via optimizing the likelihood

Figure 2.8: 5 clusters formed by 0.02% of samples with highest pheromone values resulting from M3A ($r = 0.1$, $N_{\text{steps}} = 5 \times 10^6$) applied to a window in the sky between right ascension (ra) $\in]180, 220[$ and declination (dec) $\in]15, 30[$. b-e) the application of spectral clustering on the photometric data of two globular clusters: b,d) using Gaussian kernel as a similarity measure. c,e) using the transition probability in Eq. (2.14) ($r = 0.5$).

function through an iterative process, called Expectation-Maximization (EM). It is known that EM is sensitive to the initialization step, and it may get stuck in local optima. Moreover, the time needed for EM is dependent on the number of components and increases for large data sets with many samples and dimensions. One way to avoid this problem is to use a non-parametric method, such as Parzen Window (PW), which is a special case of the finite mixture model with $K = N$, and it only has one parameter σ :

$$\hat{f}(\mathbf{x}; \sigma) = \frac{1}{N} \sum_{k=1}^N \mathcal{K}(\mathbf{x}; \mathbf{x}_k, \sigma)$$

where $\mathcal{K}$ is the Gaussian kernel function. However, if the data points are distributed along a low-dimensional manifold the spherical Gaussian kernel is often not an optimal choice (Vincent and Bengio 2003). Therefore, (Vincent and Bengio 2003) proposed the Manifold Parzen Window (MPW) which allows the Gaussian distributions to have elliptical shapes instead of spherical. Whereas PW and MPW may provide a reliable estimation, they both suffer from high computational costs in the test

Algorithm 2.3 Partitioning the data space

Input: The pheromone vector $\mathbf{f} = [f_1, f_2, \dots, f_N]$ and the balls' radius r_{ball}

Output: $\mathcal{S}$, containing the center of balls

function FINDCENTERS($X, \mathbf{f}, r_{\text{ball}}$)

Set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathcal{S} = \phi$

while $\mathcal{D} \neq \phi$ **do**

$l = \arg \max_{\mathbf{x}_i \in \mathcal{D}} f_i$

 find l 's neighbors $\mathcal{N}_l = B_{r_{\text{ball}}}(\mathbf{x}_l)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{l\}$

$\mathcal{D} \leftarrow \mathcal{D} - \{l\} - \mathcal{N}_l$

end while

end function

phase, due to the large number of Gaussian models. To prevent this problem several methods, such as Simplifying mixture models (SMM) (Zhang and Kwok 2010) and Hierarchical clustering of a mixture model (HCMM) (Goldberger and Roweis 2005) were proposed. They start with a large mixture model and then construct a simpler model, such that the distance between the original model and the simplified one is minimized. In contrast to these approaches, (Wang et al. 2009) proposed a new method called Fast Parzen Window (FPW), which partitions the data space via hyper-balls positioned randomly with fixed radii r_{ball} . Then, it fits a full-rank Gaussian distribution for each ball only. Extending the latter idea we use the pheromone values obtained by M3A to find the best position for the center of the hyper-balls. We start with the whole data set $\mathcal{D}$ and pick the point $\mathbf{x}_l$ with the highest pheromone as the center of a ball. Then we remove all points within the ball from $\mathcal{D}$. Then, we continue picking the next point in $\mathcal{D}$ with the highest pheromone as the new center and remove its neighbors from $\mathcal{D}$. Following this strategy, we obtain a more compact list of centers. Its pseudo-code is summarized in Algorithm 2.3.

To demonstrate the performance of the new strategy, we use three data sets:

- 1) a synthetic spiral shape manifold with Gaussian noise $N(0, 0.04)$ for which 1000 samples are generated as a training set (see Figure 2.9a), and another 20000 samples as a test set.
- 2) two intersected circular manifolds with radius 2: a) in the above circle the density varies and the noise level is 0.2; b) at the bottom the noise level varies between 0 to 0.3 (see Fig. 2.10a). We generated 3350 and 16750 points for training and testing, respectively. And
- 3) an astronomical simulation of a jellyfish galaxy that contains 3D position in-

Figure 2.9: a) the noisy spiral data set, b-l) the contour curves for the density estimators: b) PW, c) MPW, d) GMM+rand, e) GMM+kmeans, f) SMM, g) Goldberger, h) FPW, i) M3A+FPW: neighborhood radius $r_{rw} = 0.13$ and balls' size $r_{ball} = 0.21$.

formation of 58531 particles.

To evaluate the performance, we use the average log-likelihood (ALL) to measure how well the adapted model can describe the data set. We furthermore use 10-fold cross-validation on the training set to find the best hyper-parameter values.

Table 2.1 reports the results of the above algorithms on the spiral shape manifold. From cross-validation, hyper-parameters are found where m , k , and σ are the num-

Table 2.1: Comparison of density estimators for the spiral data set.

Algorithm	Parameters	ALL (std)	
PW	$\sigma = 0.03$	0.3130	(0.0)
MPW	$k = 45$	0.3003	(0.0)
FPW	$r_{\text{ball}} = 0.18$ ($m = 20.8$)	0.2851	(0.0092)
SMM	$m = 95$	0.2762	(0.0116)
HCMM	$m = 35, \sigma = 0.06$	0.2832	(0.0049)
GMM	$m = 20$	0.2944	(0.0028)
GMM ^k	$m = 15$	0.2802	(0.0049)
M3A+FPW	$r_{rw} = 0.13, r_{\text{ball}} = 0.21$ ($m = 17.1 \pm 1$)	0.3101	(0.0052)

ber of components and neighbors, and the scale of Gaussian kernel, respectively, and r_{rw} is the neighbor size for M3A. Note that GMM^k is the Gaussian mixture model when k-means is used to initialize components. We repeat the experiment for M3A 10 times. While ALL for the M3A+FPW (i.e., algorithm 2.3) is comparable with PW, it provides a much sparser representation for the data set and speeds up the computation in the testing phase. Moreover, Figure 2.9 shows the contour curves of the pdfs. We observe that most methods represent the center of the spiral fairly well, with the exception of PW and SMM, which provide the most noisy model. Strikingly, PW, SMM, and HCMM discontinue the manifold in the less dense tail on the right side of the spiral, dividing it into several small clusters. Our strategy M3A+FPW, on the other hand, not only provides a compact model but also successfully tracks the underlying structure and stays very close to the original manifold as compared to others.

In Table 2.2 we present the outputs of the studied algorithms on the two circular manifolds. as in the previous examples, cross-validation is used to tune hyperparameters, and then the performance of the algorithms on the test set is reported. While the FPW+M3A yields a sparse representation for the data, it also outperforms (in terms of ALL) others, except for the GMMs methods. Since the M3A can be parallelized, it is more suitable for big datasets, in comparison to GMM techniques. Moreover, we display the contour curves of pdfs in Fig. 2.10 where the true manifold (without noise) is shown in red. It can be seen that the M3A helps FPW to be more compact and more successful in tracking the manifolds, as indicated by the black diamond in the plot. In comparison to GMMs, FPW+M3A uniformly distributes Gaussian models on the manifolds, even in less dense regions. It is especially important in some applications such as astronomy where low-density streams have high importance in studying the evolution of astronomical structures. Therefore,

Figure 2.10: a) two circular manifolds with variation in density and noise level, b-l) the contour curves for the compared eight density estimators. Note that black diamonds represent the means of the Gaussian distributions.

similarly to previous experiments, we conclude that FPW+M3A builds a compact model, which is more successful in following the manifolds, regardless of their den-

Table 2.2: Comparison of density estimators for the two circular manifolds.

Algorithm	Parameters	ALL (std)	
PW	$\sigma = 0.09$	-2.668	(0.0)
MPW	$k = 140$	-2.689	(0.0)
FPW	$r_{\text{ball}} = 0.75$ ($m = 31$)	-2.682	(0.0116)
SMM	$m = 800$	-2.671	(0.0007)
HCMM	$m = 130, \sigma = 0.15$	-2.700	(0.0014)
GMM	$m = 20$	-2.620	(0.0036)
GMM ^k	$m = 20$	-2.621	(0.0034)
M3A+FPW	$r_{rw} = 0.6, r_{\text{ball}} = 0.95$ ($m = 20.6 \pm 1$)	-2.650	(0.0027)

sities.

Astronomical simulations are often used to study the evolution of astronomical objects through time. Since a simulation often includes many Millions of particles, it is not possible to track each particle individually. Therefore, it is desirable to encode the distribution of particles via a sparse pdf and follow its evolution instead. Since our goal in this section is to compare different density estimators, we only use one time snapshot of a simulated jellyfish galaxy, as shown in Figure 2.11 a and b. This data set is challenging since the density changes significantly in the data space. While the head of the jellyfish is very dense the outer parts depict much lower density. Those parts contain some low-dimensional structures called streams, as highlighted by boxes in panels a and b. During the evolution process these structures are more affected by other astronomical objects and change over time. Thus, in addition to the dense regions a suitable density estimator should capture these lower density regions well.

We compare our M3A+FPW algorithm with the result of GMM initialized by k-means visually and additionally report the quality of all density estimators as before. From Table 2.3 it can be seen that the GMM outperforms the other methods in terms of average log-likelihood (ALL), which is expected since it aims to optimize the likelihood function. However, its optimization process is time-consuming for big data sets, which makes it challenging to be applied in big simulations, as indicated by large training times reported in the last column of Table 2.3¹⁵. PW as before provides an unnecessary complex model, which is not very desirable for this application. Alternatively, the M3A+FPW strategy presents a sparse model while preserving a comparable ALL.

¹⁵Here, we distribute the M3A ants among 10 processors (i.e. 5 ants per CPU) and restrict the number of step in each round to $n = 10000$.

Figure 2.11: The simulated Jellyfish galaxy (viewpoint a and b). c-f) contour curves of density estimators projected by the first two principal directions of Gaussian covariance matrices for zoomed regions a) (left) and b) (right) using Algorithm 2.3 (c,e) and GMM initialized by k-means (d,f).

Table 2.3: Comparison of density estimators for the simulated jellyfish galaxy.

Algorithm	Parameters	ALL (std)		d_H (std)		time in s (std)	
PW	$\sigma = 0.08$	-3.775	(0.018)	0.264	(0.040)	0.2	(0.0)
FPW	$r = 0.4, m = 949$	-3.855	(0.023)	0.633	(0.071)	9.2	(0.4)
SMM	$m = 4100$	-3.815	(0.021)	0.699	(0.115)	144.8	(10.5)
SMM	$m = 600$	-4.249	(0.032)	0.393	(0.053)	63.0	(5.2)
HCMM	$\sigma = 0.17, m = 600$	-3.967	(0.014)	0.697	(0.091)	12.4	(0.5)
GMM	$m = 300$	-3.772	(0.025)	2.916	(0.600)	1244.4	(257.4)
GMM ^k	$m = 300$	-3.767	(0.024)	3.060	(0.654)	909.3	(63.1)
M3A+FPW	$r_{rw} = 0.4, r_{ball} = 0.5,$ $m = 566$	-3.838	(0.025)	0.737	(0.109)	165.5	(2.1)
M3A+FPW	$r_{rw} = 0.4,$ $r_{ball} = 0.65, m = 309$	-3.882	(0.024)	0.823	(0.072)	151.2	(11.9)

In addition to ALL we also report the Hausdorff distance

$$d_H(A, B) = \max \left\{ \max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b) \right\}$$

where d denotes the Euclidean distance, and A and B are the set of particles and the set of centers of Gaussian models, respectively.

It is informative since it shows how much the model captures the original low-density structures. To compare our method to GMM we increase the balls' radius to $r_{\text{ball}} = 0.65$. From the table it is clear that the Hausdorff distance of the proposed method is much lower than the one achieved by GMM. To visually see its success we added the contour plots (see panels c-f). It can be seen that the new method is more successful in fitting the pdf along with the low-dimensional structure. In summary, if selecting a model is a trade-off between simplicity (sparsity) and capturing structural details, the M3A+FPW strategy, outlined in Algorithm 2.3, could successfully improve the performance of FPW algorithm in terms of both ALL value and sparsity, i.e. while it reduces the number of components by 40%, it can still increase the ALL value.

2.7 Conclusion

In some applications, such as astronomy, it is common to have low-dimensional structures buried inside big data sets. Therefore, it is desirable to have a method that extracts these structures of varying density while their continuity is kept unchanged. Although there are several methods, they often fail to either extract manifolds or keep their continuity. Inspired by the ant colony algorithm we propose a new method where a value, called pheromone, is assigned to each sample. Later these quantities can be used to reveal manifolds without the undesirable effects. To study the behavior of the algorithm and the effect of its hyper-parameters, we provide a theoretical analysis using the Markov Chain framework, where we consider a noisy manifold and apply the algorithm examining two cases: a) a random walker with an unbounded number of steps, and b) an unbounded number of walkers with a fixed number of steps. We show for both cases that the pheromone distribution captures information about the distances of data points to the underlying manifold. In addition to the theory, we empirically analyze the algorithm using synthetic and real data sets, demonstrating three different scenarios: a) denoising and clustering manifolds, b) visualizing data, and c) density estimation. In all investigated scenarios and application examples strategies based using M3A exhibit comparable or superior results in suppressing the noise, capturing the manifolds and providing sparser models.

Published as:

M. Mohammadi and K. Bunte. – “Multi-agent Based Manifold Denoising.” In International Conference on Intelligent Data Engineering and Automated Learning 2020, Springer, Cham, pp. 12-24.

Chapter 3

Multi-Agent based Manifold Denoising

Abstract

The existence of the gravity causes formation of variety of structures, from large-scale structure, such as galaxy clusters, to smaller ones such as streams and shells. To study this process, astronomers use simulations to model matter distribution and study formation and evolution of such structures, however, it is difficult to infer from them because of the distribution of particles. One way to facilitate the inference process is to clean structures via attaching each particle to its underlying structures. However, the diversity in density and dimensionality of structures make this task much more challenging. In this chapter, we propose a novel (swarm) algorithm to clean these structures of potentially varying density and dimensionalities. Inspired by colonial insects this method employs multiple agents with different strategies moving through the data space in parallel. During this process, they use local information to reconstruct the manifolds and then move data objects close to them. Moreover, principles of evolutionary game theory are used to encourage agents to select better strategies and hence optimize the hyper-parameters automatically. While other denoising techniques can be seen as single-agent approaches, the new algorithm is a multi-agent approach which makes it more flexible and suitable for scenarios including multiple manifolds. In the experiments, we simulate several situations from a simple manifold with a specific noise level, to more complex one, N-body cosmological simulation, where there are variations on the density, noise level and dimensionalities. Furthermore, we demonstrate the improvement of the proposed algorithm for the performance of the Fast Parzen Window (FPW) density estimator.

3.1 Introduction

Nowadays computers have a central role on developments of astronomy. Their role on performing large-scale simulations has a huge impact on our knowledge about the universe. In cosmology, N-body simulations, modeling matter distribution, are used to study formation and evolution of structures in the universe. The presence of gravity forms a variety of structures, from spherical shape with

high density clusters to 1D filaments with lower density. Therefore, investigating outputs of large-scale simulations is a challenging task, and it is needed to simplify this process via cleaning and extracting the underlying manifolds. In addition to astronomical simulation, cleaning manifolds is valuable in applications such as 3D mesh smoothing in the computer graphics community (Taubin 1995). Hence, multiple cleaning methods have emerged as preprocessing strategy to suppress the noise.

Many strategies to denoise manifolds have been proposed, which are typically applied as a preprocessing step on data sets to reduce the noise before solving the Machine Learning (ML) task. For instance, Manifold Denoising (MD) (Hein and Maier 2006), Mean Shift (MS) (Xiang and Chen 2016) and Manifold Blurring Mean Shift (MBMS) (Wang and Carreira-Perpinán 2010) decrease the noise level, iteratively, via pushing data points towards the underlying manifold. Unlike these iterative algorithms, Locally Linear Denoising (LLD) (Gong et al. 2010) optimizes a cost function fulfilling two objectives: a) preserving local linear structures and b) a smooth manifold. However, its computation and memory costs make LLD undesirable for big data sets¹. In general, the success of these methods highly depends on using proper values for their hyper-parameters which often require information of the manifold and the noise itself. Thus it is highly desirable for a manifold denoising algorithm to make fewer presumptions about the data and the nature of noise.

Swarm intelligence (SI) is a self-organized system of multiple agents performing individual actions leading to intelligent collective behaviour. A fascinating example of SI is the well-known and studied ant colony. Several clustering techniques are inspired by the ability of ants to cluster corpses and sort larvae (Deneubourg et al. 1991, Lumer and Faieta 1994, Tsai et al. 2004, Shelokar et al. 2004, Chu et al. 2004, Runkler 2005). They modeled this capability by two simple rules called picking up and dropping down, where several ants walk in the data space, pick up data objects and then drop them down somewhere else (see Fig. 3.1). Although no central authority controls them, their local interactions recover underlying clusters by moving similar objects close to each other. While typically less efficient than classical clustering methods, swarm strategies proved captivating and beneficial through their flexibility, simplicity, and robustness.

In this chapter, we propose a multi-agent evolutionary algorithm to denoise manifolds. In the new algorithm, agents with different capabilities walk in the data space and move samples close to manifolds. While the previous works need to be tuned with appropriate hyper-parameters, which require prior knowledge of the manifold and noise, we use the Evolutionary Game Theory (EGT) to find the best hyper-parameters automatically. Using multiple agents equipped with different ca-

¹In addition to applying local PCA for every data point, LLD needs to compute the inverse of an $n \times n$ matrix with the complexity of $O(n^3)$ (n : number of data points).

Figure 3.1: Real ants' behavior in forming cemeteries (Theraulaz et al. 2003). They display four consecutive images of the plate.

pabilities makes the novel formulation suitable for multi-manifold scenarios where manifolds may have priorly unknown with variation in dimensionality and noise.

This chapter is organized as follows: we first detail the problem setting and introduce the MBMS (Wang and Carreira-Perpinán 2010), which is the most related work to our algorithm, and EGT (section 3.2). In section 3.3 we explain our novel algorithm. The performance is then demonstrated and compared with empirical analysis by means of several example scenarios in section 3.4. Finally, we conclude in section 3.5.

3.2 Related work

3.2.1 Problem setting

We assume that the data is distributed along a d -dimensional smooth manifold embedded in a D dimensional space, with $D > d$. Let N samples $\{z_i \in \mathbb{R}^D : i = 1, \dots, N\}$ be randomly selected from the manifold. Due to noise ϵ the observations x_i do not lie exactly on the manifold, such that $x_i = z_i + \epsilon_i$. The goal of denoising is to recover z_i from noisy samples x_i . If we assume the underlying manifold is linear and the noise ϵ follows a Gaussian distribution, then Principal Component Analysis (PCA) can be used to denoise the manifold. From PCA, we obtain the parameter set $\{(\lambda_i, u_i)\}_{i=1}^D$, where λ_i and u_i represent the i -th eigenvalue and its corresponding eigenvector. Without loss of generality we assume the eigenvalues are sorted in descending order² and re-scaled, such that $\sum_{i=1}^D \lambda_i = 1$. Defining $U_i = u_i u_i^T$ the covariance matrix C can be re-written as:

$$C = \sum_{i=1}^D \lambda_i U_i = S_1 U_1 + \frac{1}{2} S_2 (U_1 + U_2) + \dots + \frac{1}{D} S_D (U_1 + \dots + U_D) \quad (3.1)$$

where $S_i = i \times (\lambda_i - \lambda_{i+1})$ ($\forall i = 1, \dots, D-1$) and $S_D = D \cdot \lambda_D$. As suggested in (Wang et al. 2008, Mordohai and Medioni 2005) one can estimate the intrinsic dimensionality of the manifold by:

$$\hat{d} = \arg \max_i S_i. \quad (3.2)$$

Then the first $\hat{d}$ eigenvectors $\{u_1, \dots, u_{\hat{d}}\}$ can be used to recover the linear manifold and x_i is denoised via projecting on the reconstructed space.

3.2.2 Manifold Blurring Mean Shift (MBMS)

The primary goal of any denoising algorithm is to repress noise while simultaneously keeping the important details. One way to achieve it is to move the data points towards the manifold. Thus, for a given sample x_i the update is: $x_i^{\text{new}} = x_i^{\text{old}} + \delta x_i$, where δx_i represents the direction of the motion. Since the position of samples on the manifold reveals important details, the motion should not take place along the manifold (Wang and Carreira-Perpinán 2010). Thus, MBMS restricts motions to be parallel to the manifold normals. Although the manifold may be non-linear, it can be

²In practice, since we compute eigenvalues recursively, from the biggest to the smallest, this assumption does not cause any extra computational costs.

locally estimated by linear patches using eigenvectors derived by PCA. Therefore, the update term is:

$$\delta \mathbf{x}_i = (I - UU^T)(\boldsymbol{\mu} - \mathbf{x}_i) \quad (3.3)$$

where U 's columns contain the first d eigenvectors and $\boldsymbol{\mu}$ is the kernel average of $\mathbf{x}_i$'s neighbors. Our novel strategy keeps the idea of limiting the motion to be parallel to manifold normals. However, to avoid fixing the dimensionality d of the manifold beforehand it is estimated locally as integral part of our strategy.

3.2.3 Evolutionary Game Theory (EGT)

Game theory studies strategic interaction such as competition or cooperation of players. While classical game theory assumes that all players make rational decisions EGT replaces it by biologically inspired notions, such as natural selection (Hofbauer et al. 1998). Let a population contain n strategies (or types) E_1 to E_n with frequencies p_1 to p_n , and $\mathbf{p} = (p_1, \dots, p_n)$ denote the distribution of strategies within the population. Let f_i represent the fitness of strategy i which reflects its reproductive success. If the size of the population is very large, then its evolution can be modelled as a differential equation. For a given strategy i the rate of change in its frequency (i.e. $\frac{\dot{p}_i}{p_i}$) measures its evolutionary success. Following Darwin's doctrine the replicator dynamics expresses evolutionary success as the difference between the fitness of the strategy and the average fitness of the population:

$$\frac{\dot{p}_i}{p_i} = f_i(\mathbf{p}) - \bar{f}(\mathbf{p}) \quad (3.4)$$

Thus, if a strategy has a higher fitness than the average value, its population share increases and vice versa. Using finite difference approximation, the above equation can be re-written as:

$$p_i^{(t+1)} = p_i^{(t)}(f_i(\mathbf{p}) - \bar{f}(\mathbf{p})) + p_i^{(t)}$$

where $p_i^{(t)}$ is the frequency of i -th strategy in t -th generation.

3.3 A Multi-Agent Paradigm for Manifold Denoising

In this chapter we combine aspects of EGT with principles of the above mentioned SI clustering techniques resulting in a novel strategy for manifold denoising. The basic idea is to release several ant-like agents in the data space, which can, such

as their biological counterparts, pick up data samples and bring them closer to the manifold.

To achieve this goal we need to define individual behaviour comprising of the following objectives:

- O1: an approximation strategy to recognize the manifold structure,
- O2: a good policy to keep agents close to the manifold,
- O3: and an update rule bringing samples close in proximity of the respective manifold (decreasing the noise).

In the following subsections we explain how we address the above challenges, describing the strategy for one agent first followed by the generalization to several agents. While in the first case an agent with a specific capability is introduced, the second case takes advantage of having several agents with different abilities (or strategies). For the latter we use EGT to automatically determine the winning strategies and therefore avoiding the necessity to fix them beforehand.

3.3.1 Manifold Alignment Aware Agent

While a linear manifold can be approximated by PCA it is challenging to estimate non-linear or multiple manifolds. Hence, non-linear manifolds are commonly estimated by local linear patches, namely given a data point x a manifold in x 's neighborhood can be approximated by the tangent space at x . A way to estimate tangent spaces is to recover them by the eigenvectors derived from PCA. Given a radius r the set of x 's neighbors is defined as $\mathcal{N}_x = \{x_i : \|x_i - x\| < r\}$. Using PCA on $\mathcal{N}_x$ one can use (3.2) to approximate the dimensionality of the manifold $\hat{d}$ and then recover it by the first eigenvectors $\{u_1, \dots, u_{\hat{d}}\}$. Thus, a non-linear manifold is described by a set of (overlapping) linear patches.

When the manifold is approximated the denoising can be translated to bringing samples closer to it. Thus, our aim is to define a random walk which reinforces an agent to walk through the data cloud and bring data objects closer to the underlying manifold. Let an agent, with a specific view range r , be on a data point x . It uses local PCA to approximate the manifold and its intrinsic dimensionality. Hence, the distance of samples to the manifold is estimated by:

$$\delta^{\mathcal{M}}(i) = \|(I - UU^T)(\mu - x_i)\| \quad (3.5)$$

where $\|\cdot\|$ is the Euclidean norm and $\mu = \frac{1}{|\mathcal{N}|} \sum_{k: x_k \in \mathcal{N}_x} x_k$. Then for every $x_i \in \mathcal{N}_x$

we define the following weight values:

$$w(\mathbf{x}, \mathbf{x}_i) = \begin{cases} 1 - \frac{\delta^{\mathcal{M}}(i)}{b} & \text{if } \delta^{\mathcal{M}}(i) \leq b \\ 0 & \text{if } \delta^{\mathcal{M}}(i) > b. \end{cases} \quad (3.6)$$

where b is selected such that only $p = 50\%$ of neighbors have non-zero weights. Next the agent selects its next destination by the following probability:

$$P_{\text{rw}}(\mathbf{x}, \mathbf{x}_i) = \frac{w(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^k w(\mathbf{x}, \mathbf{x}_j)} \quad (3.7)$$

Thus, using only local information the manifold is approximated and then agents are encouraged to stay close to it fulfilling objectives O1 and O2.

During the random walk agents pick up and later drop down some of the data points in their neighborhood aiming to reduce the noise of the manifold. To fulfill objective O3, in every step an agent randomly selects one (or more) data point $\mathbf{x}_i$ for picking up with the following probability:

$$P_{\text{pick}}(\mathbf{x}_i) = \frac{1 - w(\mathbf{x}, \mathbf{x}_i)}{\sum_{j=1}^k (1 - w(\mathbf{x}, \mathbf{x}_j))}. \quad (3.8)$$

Hence, the further $\mathbf{x}_i$ is from the tangent space the more probable it is to be picked up. Then the main goal of the agent is to drop it down closer to the manifold. Motivated by MBMS, it is achieved by the following rule:

$$\mathbf{x}_i^{\text{new}} = \mathbf{x}_i^{\text{old}} + \eta(I - UU^T)(\boldsymbol{\mu} - \mathbf{x}_i^{\text{old}}), \quad (3.9)$$

where $\eta > 0$ is the learning rate which controls how much change an agent can do in a single step. Moreover, to prevent unnecessary change, we introduce a threshold τ , such that agents only change the neighborhood if the mean distance of neighbors to the tangent space (quantified by $\sum_{i=d+1}^D \lambda_i$) is bigger than τ . In other words, a neighborhood only changes when the effect of noise and curvature is large enough. Thus, we can decrease the chance of over-smoothing and shrinkage.

In summary, an agent is released on the data set performing a random walk following the local preference (3.7). During this process it picks up a data object according to (3.8) and in the following carry the sample closer to the manifold using Eq. (3.9). The pseudo code for the method is given in Algorithm 3.1. In addition to denoising the data set, the agent also counts the number of changes it does and the number of samples it visits in any dimensionality. These outputs provide useful information about the manifold.

Algorithm 3.1 One Agent

Input: data set X , initial position x_0 , radius r , learning rate η , number of steps N_s and threshold τ

Output: denoised dataset (X^{new}), number of changes in the environment (n), number of visited samples in different dimensionalities (ξ)

function ONEAGENT($X, x_0, r, \eta, N_s, \tau$)

$\xi \leftarrow \mathbf{0}_{D \times 1}, n \leftarrow 0$

$x_c \leftarrow x_0$

for $s \leftarrow 1, N_s$ **do**

find x_c 's neighbors $\mathcal{N}$

$(\mu, \lambda, U) \leftarrow \text{PCA}(\mathcal{N})$

approximate the intrinsic dimensionality $\hat{d}$ in $\mathcal{N}$ by (3.2)

$\xi[\hat{d}] \leftarrow \xi[\hat{d}] + 1$

if $\sum_{i=\hat{d}+1}^D \lambda_i > \tau$ **then**

$n \leftarrow n + 1$

select a neighbor to pick up by Eq. (3.8)

bring the selected neighbor closer to the manifold using Eq. (3.9)

end if

use (3.7) to select the next place x_i and set $x_c \leftarrow x_i$

end for

end function

3.3.2 Evolutionary Manifold Alignment Aware Agents (EM3A)

As it is shown in (Kaslovsky and Meyer 2014) the performance of PCA and the method above highly depend on the neighborhood size r . While high curvatures encourage the use of smaller radii, it should be large enough to suppress the effect of noise and contain enough samples (Little et al. 2011). Hence, one needs extra information over the data in order to pick a proper value for r . In the lack of such extra information and for increased flexibility, we propose an optimization combining SI strategies with concepts of EGT, to automatically adapt such parameters.

EGT studies how a population of strategies evolves, where the population share of better strategies increases over time. Thus, if we define r as a strategy, EGT can be used to find appropriate values for it. Since r is a continuous variable, we assume $r \in [R_{\min}, R_{\max}]$, and discretize by division into m smaller intervals $([r_i, r_{i+1}), 0 \leq i < m)$ where each interval is viewed as a strategy. Let each agent randomly select a strategy following a population share distribution $\mathbf{p}^{(t)} = [p_k^{(t)}]_{m \times 1}$, where $p_k^{(t)}$ denotes the population share of the k -th strategy in the current generation. Given an agent with strategy k , r is chosen from $[r_k, r_{k+1})$ uniformly. Next, a queen (or

master branch) sends a copy of the data set to all agents, for them to separately do the random walk as explained in Algorithm 3.1. After finishing the walk with N_s steps, each agent sends back its output to the queen. The queen updates the data set taking the average over all copies received from the agents and then it computes the fitness of the strategies. Let S_k denote a set of agents with strategy k , then $\sum_{i \in S_k} n_i$ denotes the number of times they change the data set. The fitness is defined as:

$$f_k = \frac{\sum_{i \in S_k} n_i}{|S_k| \cdot N_s} \quad (3.10)$$

where $|S_k|$ is the number of agents with strategy k . The queen uses the replicator equation to adapt the population share of each strategies for the next generation:

$$p_k^{(t+1)} = p_k^{(t)} + p_k^{(t)} (f_k^{(t)} - \bar{f}^{(t)}) \quad (3.11)$$

In the next generation agents use the distribution $\mathbf{p}^{(t+1)} = [p_k^{(t+1)}]$ to select their own strategy. The resulting method is detailed in Algorithm 3.2. Note that among the parameters in EM3A the most important one is the learning rate η which controls the size of changes. Since agents have different view range r the term $(\mathbf{x}_i - \boldsymbol{\mu})$ in Eq. (3.9) is in different scales for different r . Therefore, we re-define $\eta = \frac{\rho}{\sqrt{r}}$ with constant ρ . The selection of ρ has a direct effect on the smoothness of manifolds, and it should be $\rho \in (0, 1]$. Moreover, the number of steps N_s and agents N_a should be big enough to ensure all parts of the data set are reachable. One can select it such that $N_s \times N_a > c \cdot N$ for a constant c .

3.3.3 Specific case (EM3A⁺)

While the EM3A algorithm is flexible in terms of applicability in multi-manifold scenarios, its application on big datasets is challenging since it needs every agents to find its neighbors, i.e. data points within a given distance, in every step.

To decrease the computational cost, we propose a specific case of the EM3A algorithm where we set $N_s = 1$ and $N_a = N \cdot m$. More precisely, for each strategy we use exactly N agents who are distributed such that one agent per point, and each agent only allowed to update the point where it is placed on. In other words, we define the picking up probability for an agent on point $\mathbf{x}_i$ as follows:

$$P_{\text{pick}}(\mathbf{x}_j) = \delta_{ij} \quad (3.12)$$

where δ is the kronecker delta. Since in every iteration all points can be updated, to decrease the speed of changes, we use kernel mean in Eq. (3.9), instead of a simple

Algorithm 3.2 EM3A

Input: data set X , number of strategies m and their boundaries $(R_{\min}, R_{\max})$, learning rate η , number of agents N_a , number of steps N_s , threshold τ , stopping criteria: N_{iter}

Output: denoised data set (X^{new}) , fitness values f , population share p

function EM3A($X, m, R_{\min}, R_{\max}, \eta, N_a, N_s, \tau, N_{\text{iter}}$)

$X^{\text{new}} \leftarrow X$

$p^{(0)} \leftarrow \frac{1}{m} \mathbf{1}_{D \times 1}$

for $t \leftarrow 1, N_{\text{iter}}$ **do**

$n \leftarrow \mathbf{0}_{D \times 1}, s^{(t)} \leftarrow \mathbf{0}_{m \times 1}, \xi^{(t)} \leftarrow \mathbf{0}_{D \times 1}$

for $i \leftarrow 1, N_a$ **do**

following $p^{(t)}$, select its strategy (say k), and $s^{(t)}[k] \leftarrow s^{(t)}[k] + 1$

select a neighborhood size $r \in [r_{k-1}, r_k]$

Randomly select a node x_0 as an starting point for the agent

$[X_i, n_i, \xi_i] = \text{OneAgent}(X^{\text{new}}, x_0, r, \eta, N_s, \tau)$

$n[k] \leftarrow n[k] + n_i$

end for

$X^{\text{new}} \leftarrow \frac{1}{N_a} \sum_{i=1}^{N_a} X_i, \xi^{(t)} \leftarrow \sum_i \xi_i$

compute $f_k^{(t)}$ and $p^{(t+1)}$ by Eq. (3.10) and (3.11), respectively.

end for

end function

mean value, i.e.:

$$\mu = \sum_{k: \mathbf{x}_k \in \mathcal{N}_i} \frac{K_\sigma(\mathbf{x}_i, \mathbf{x}_k)}{\sum_{l: \mathbf{x}_l \in \mathcal{N}_i} K_\sigma(\mathbf{x}_i, \mathbf{x}_l)} \mathbf{x}_k$$

where $K_\sigma(\cdot, \cdot)$ is the Gaussian kernel function with scale parameter σ . Note that the mean value $\mu = \frac{1}{|\mathcal{N}_i|} \sum_{k: \mathbf{x}_k \in \mathcal{N}_i} \mathbf{x}_k$ is a specific case of the kernel mean as $\sigma \rightarrow \infty$. Since now the number of agent is equal for all strategies, $p^{(t)}$ does not represent the population share distribution anymore, but it reflects the importance of each strategy. It means the higher fitness a strategy has the higher impact it has on the output. Therefore, instead of a simple averaging, the queen computes a weighted mean from the received copies, i.e.:

$$X^{\text{new}} = \sum_{i=1}^m p_i^{(t)} \cdot X_i \quad (3.13)$$

To increase the speed, we distribute strategies among processors (instead of agents), so agents with similar strategy present on the similar processor and use the same

copy of data. The modified version of EM3A can be seen in 3.4.

Algorithm 3.3 One Strategy

Input: data set X , radius r , learning rate η and threshold τ
Output: denoised dataset (X^{new}), number of changes in the environment (n), number of visited samples in different dimensionalities (ξ)
function ONESTRATEGY(X, r, η, τ)
 $\xi \leftarrow \mathbf{0}_{D \times 1}, n \leftarrow 0$
 for $i \leftarrow 1, N$ **do**
 find x_i 's neighbors $\mathcal{N}$
 $(\mu, \lambda, U) \leftarrow \text{PCA}(\mathcal{N})$
 approximate the intrinsic dimensionality $\hat{d}$ in $\mathcal{N}$ by (3.2)
 $\xi[\hat{d}] \leftarrow \xi[\hat{d}] + 1$
 if $\sum_{k=\hat{d}+1}^D \lambda_k > \tau$ **then**
 $n \leftarrow n + 1$
 bring x_i closer to the manifold using Eq. (3.9)
 end if
 end for
end function

3.3.4 Complexity analysis

Every agent should perform two operations: a) detecting its neighbors and b) applying PCA on the neighbors. The neighborhood search needs to compute the pairwise distances of the N samples of the data set, resulting in a complexity $O(N^2)$. Approximate neighbor search strategies can reduce the computation, for example using a k-d tree reduces the complexity to $O(N \log N)$ (Bentley 1975). In addition to the neighbor search the agents perform PCA which is done by Singular Value Decomposition. This complexity depends on the specific implementation and in the worst case it scales cubic with the dimensionality of the data and quadratic with the number of samples in the neighborhood $|\mathcal{N}_x|$. Again approximate strategies can reduce the computational complexity (Golub and Van Loan 2012). Moreover, the execution of the EM3A (and EM3A⁺) is parallelizable via distributing agents (or strategies) among several processors. In the following section, we show the acceleration of the process.

Algorithm 3.4 EM3A⁺

Input: data set X , number of strategies m and their boundaries $(R_{\min}, R_{\max})$, learning rate η , threshold τ , stopping criteria: N_{iter}

Output: denoised data set (X^{new})

function EM3A⁺($X, m, R_{\min}, R_{\max}, \eta, \tau, N_{\text{iter}}$)

$X^{\text{new}} \leftarrow X$

$p^{(0)} \leftarrow \frac{1}{m} \mathbf{1}_{D \times 1}$

for $t \leftarrow 1, N_{\text{iter}}$ **do**

$\mathbf{n} \leftarrow \mathbf{0}_{D \times 1}, \mathbf{s}^{(t)} \leftarrow \mathbf{0}_{m \times 1}, \boldsymbol{\xi}^{(t)} \leftarrow \mathbf{0}_{D \times 1}$

for $k \leftarrow 1, m$ **do**

select a neighborhood size $r \in [r_{k-1}, r_k)$

for $i \leftarrow 1, N$ **do**

$[X_i, n_i, \xi_i] = \text{OneStrategy}(X^{\text{new}}, r, \eta, \tau)$

$\mathbf{n}[k] \leftarrow \mathbf{n}[k] + n_i$

end for

end for

$X^{\text{new}} \leftarrow \sum_{i=1}^m p_i^{(t)} \cdot X_i, \boldsymbol{\xi}^{(t)} \leftarrow \sum_i \xi_i$

compute $f_k^{(t)}$ and $\mathbf{p}^{(t+1)}$ by Eq. (3.10) and (3.11), respectively.

end for

end function

3.4 Experiments

In this section, we demonstrate and investigate several aspects of the proposed method on different data sets and application scenarios, including:

- handling variation in density, noise, and dimensionality and
- improving the result of density estimators.

In subsection 3.4.1, we use four synthetic datasets and one real dataset to simulate different situations, and we use mean square error (MSE) to evaluate outputs. Section 3.4.2 demonstrates the effect of EM3A used as preprocessing step before density estimation, exemplified by Parzen Window (PW) (see Table 3.2).

3.4.1 Denoising manifolds

To see the strengths and weaknesses of different denoising algorithms, we use three synthetic data sets³: a) an s-shaped manifold, b) a circular manifold where the noise and density along a one dimensional manifold is varied, and c) a combination of

³Code and supplementary material: <https://github.com/mohammadimathstar/EM3A>

multiple manifolds of different dimensions. Here, we set $R_{\min} = 0$, $R_{\max} = 2$ and $m = 10$ with $\tau = 0.01$. To ensure $N_a \times N_s > 2N$, we set $N_s = 100$ and $N_a = 100$. While we use $\rho = 1$ and $\sigma = \infty$ for EM3A, they are set to $\rho = \sigma = 0.5$ to limit amount of changes over an iteration for the EM3A⁺. Moreover, we apply EM3A⁺ algorithm on a cosmological n-body simulation to extract the underlying structures.

Simple manifold

To investigate how EGT helps to achieve proper values for r , we extract 4000 data points from the s-curve manifold exhibiting Gaussian noise $\mathcal{N}(0, 0.15)$. The panels c) and d) in Fig. 3.2 display an example output of EM3A, where it successfully decreases noise, especially in less curved regions. Panels (e, f) show the fitness values and population sizes of the losing strategies for every iteration, while panels (g,h) report the corresponding values for the winning strategies. One may divide the strategies into three categories. The first group, including strategies 1 and 2, contains agents with low view range r where noise has a major effect on the eigenvalues. Hence, agents with these strategies rarely observe the low-dimensional manifold and their fitness values become small and their population decreases over time. On the other hand, the second category strategies 7-10, includes agents with large view range. The major obstacle for them in observing 2D manifold is curvature. Particularly, high curvature regions prevents them to detect the manifold so their populations continuously reduces as well. Finally, the winning strategies 3-5 involve agents with medium view range where r is big enough to suppress the effect of noise and small enough not to be defected by curvature. Thus, they survive while the rest disappears. Moreover, the algorithm provides information over the dimension of the manifold as learned in the iterations (panel i). At first, since the noise level is high, most agents see a 3D object. Since the noise level decreases over time and agents pick better strategies, agents observe the 2D manifold more often. Panel j) exemplifies the achieved speed up for this experiment if agents are distributed among several processors. Because of the transfer time adding more than 3 cores does not accelerate the computation further for this experiment. Note that optimizing parallelization is possible but not in the focus of this chapter.

Manifold with varying noise

Fig. 3.3 (a) displays a circular manifold which contains a sharp change in its density, and is disrupted by Gaussian noise with varying $\sigma \in (0, 0.3)$ along the manifold. To apply denoising techniques, we set their hyper-parameters as follow: 1) MD: $\delta t = 0.01, k = 80, N_{\text{iter}} = 200$. 2) LLD: $k = 200, \lambda = 0.01, \sigma = 0.02, d = 1$. 3) SAF: $r = 0.5, \mu = 0.2, N_{\text{iter}} = 3$. 4) MBMS: $k = 60, \sigma = \infty, N_{\text{iter}} = 2, d = 1$.

Figure 3.2: Two views of the data set (a,b) and the results of EM3A (c,d). The fitness and population of losing (e,f) and winning (g,h) strategies. (i) the number of visited points with different dimensionalities. (j) computation time with number of processors used.

The Fig. 3.3 shows their outputs and Table 3.1 reports MSE. To report the MSE for EM3A, we run it 5 times. From the table, EM3A outperforms others since it limits the motion of samples to be parallel to the normal vectors, similar to MBMS. However, unlike MBMS, which updates all samples in every iteration, EM3A tries to only update points which are further from the underlying circle. Similar to MBMS, EM3A⁺ updates all samples in every iteration, but since MBMS is tuned with appropriate values for its hyper-parameters, it is expected to give a better result. The effect of the learning rate ρ (and η) on the reconstruction error is shown in Fig. 3.3g. Smaller values lead to more stable results but need more iterations to run. Thus, the selection of $\rho \in (0, 1]$ represents a trade-off between the iterations necessary and the stability of the result, and it needs to be small for more complex structures. In summary, EM3A can handle variations in density and noise of the manifold very well, while not needing to make presumptions about the manifold beforehand.

Multi-manifold example

Often real data sets contain several manifolds with different dimensionalities. For demonstration, we generate manifolds forming a basket-like structure shown in Fig. 3.4 (a). The top manifold contains 3600 data points uniformly sampled from a hemisphere, and the bottom manifold includes 1000 data points generated from a semi-circle. Then, Gaussian noise with $\sigma = 0.3$ is added. Since LLD and MBMS needs to know the intrinsic dimensionality d , they can not be used directly on a data set containing multiple manifolds. In order to compare MBMS we cheat and separate

Figure 3.3: Gray points: a 1D manifolds varying in density and noise level. Red points: the denoised manifold, using: a) MD, b) LLD, c) SAF, d) MBM and e) EM3A. f) EM3A⁺: $N_{\text{iter}} = 6$. g) evolution of the MSE (to the ground truth) per iteration for values of $\rho \in \{0.1, 0.5, 1\}$.

the two manifolds to be able to apply it. On the other hand MD, SAF, EM3A and EM3A⁺ do not need to know d and can be used in a realistic scenario. As seen in Fig. 3.4b MD depicts some undesired effects: it creates artificial holes and discontinuity in the hemisphere and circle structures, respectively, and does not completely clean the 1D basket handle. Although MBMS provides comparable clean manifolds as EM3A and EM3A⁺, it needs to denoise them separately. EM3A⁺ and EM3A outperforms others without any preprocessing (see Fig. 3.4c) and additionally provides information about dimensions of the manifolds as visualized in panel e.

Cosmological N-body simulations

An important tool to study the evolution of large-scale astronomical structures is N-body simulations where it is modeled as interaction among particles following physical rules. The presence of gravity causes the accumulation of particles which later form different types of structures, including walls (plane-like structures), filaments (1D structures which created from crossing two walls) and clusters (spherical structures). Since positions of galaxies on these structures influence their evolution tracks, it is important to have a good estimation for these structures (Taghribi et al. 2020). Here, we show how we can use our algorithm to reconstruct the low-

Figure 3.4: comparison of the original noisy basket data (panel a) to the results after applying b) MD: $\delta t = 0.02, k = 80, N_{\text{iter}} = 80$. c) SAF: $r = 0.6, \mu = 0.5, N_{\text{iter}} = 4$. d) MBMS: $k = 20, \sigma = \infty, N_{\text{iter}} = 4$. e) EM3A. f) EM3A⁺: $N_{\text{iter}} = 7$. g) the number of visited samples in given dimensionalities per iteration (for EM3A).

Table 3.1: Average distances to the ground truth

Method	Dataset	
	Circle	Basket
Noisy	0.2498 (0.0000)	0.4782 (0.0000)
MD	0.1941 (0.0000)	0.4343 (0.0000)
LLD	0.1781 (0.0000)	-
SAF	0.1771 (0.0000)	0.4459 (0.0000)
MBMS	0.1628 (0.0000)	0.3720 (0.0000)
EM3A	0.1617 (0.0002)	0.3640 (0.0007)
EM3A ⁺	0.1779 (0.0002)	0.3614 (0.0004)

dimensional filaments from the simulation while clusters stay unchanged.

Fig. 3.5 (a) depict an N-body simulation containing 115697 particles. Recovering the underlying manifolds is challenging since there are variation on both dimensionality and density. Particularly, the density significantly changes in dataspace, even along a single filament. To apply SAF, we checked several values for its hyperparameters, and we used the best values for the experiment: $r = 0.5, \mu = 0.4, N_{\text{iter}} = 4$. Moreover, since using EM3A is time consuming, we only apply EM3A⁺, and we decrease the number of its strategies $m = 4$ and set $R_{\text{min}} = 0.2, R_{\text{max}} = 1$. Fig. 3.5 (b) shows the output of SAF. Not only it ruins clusters, but also it discretizes filaments

via pushes particles towards denser regions. On the other hand, Fig. 3.5 (c) displays result of EM3A⁺. While it keeps clusters unchanged, it successfully recover the underlying manifolds of filaments without causing discontinuity.

3.4.2 Manifold denoising as a preprocessing step

Density estimation

One major problem in machine learning is to approximate the underlying probability density function (pdf) of a data set. If there is no prior knowledge about the form of the pdf, the Parzen Window (PW), a well-known non-parametric density estimation technique, is appealing. However, for larger data sets, this method becomes computationally expensive and strategies have been proposed to reduce its burden. For instance, Fast Parzen Windows (FPW) (Wang et al. 2009) decreases the amount of computation by two steps:

1. partition the data set via hyper-balls with fixed radii r (see algorithm 3.5) and
2. fit a Gaussian distribution on each ball, separately.

In order to find the centres, it randomly selects a sample as a centre and then removes all its neighbors closer than r and continues this process on the remaining samples until only centres remain. Although it provides a sparse representation

Algorithm 3.5 Covering data space with hyper-balls

Input: data set $\mathcal{D}$, radius r .

Output: set $\mathcal{S} = \{s_1, \dots, s_q\}$ containing centers of balls, and the partition $\{\mathcal{S}_1, \dots, \mathcal{S}_q\}$

procedure COVERING($\mathcal{D}, r$)

$\mathcal{S} \leftarrow \emptyset, \mathcal{T} \leftarrow \emptyset, j \leftarrow 0$

while $\mathcal{D} \setminus \mathcal{T} \neq \emptyset$ **do**

$j \leftarrow j + 1$

Randomly select a point $x_i \in \mathcal{D} \setminus \mathcal{T}$ and add it to the partition set $\mathcal{S}$ as well as to the set $\mathcal{T}$

for every data point $x \in \mathcal{D} \setminus \mathcal{T}$ **do**

if $\|x - x_i\| \leq r$ **then**

add x to the partition $\mathcal{S}_j$ centered at $s_j = x_i$ and add x to $\mathcal{T}$

end if

end for

end while

end procedure

for the pdf, its performance can be degraded by noise. We demonstrate the benefit

Figure 3.5: a) N-body cosmological simulation, b) SAF, c) EM3A⁺.

Figure 3.6: a) Fast Parzen Window (FPW), b) EM3A+FPW, c) EM3A⁺⁺+FPW, and contour curves for the likelihood function of FPW, EM3A+FPW and EM3A⁺⁺+FPW in panel d), e) and f) respectively.

of denoising, using EM3A and EM3A⁺, as preprocessing and perform FPW subsequently on the denoised data set to find the centres. In order to show its impact we use a 2D data set aligned along a spiral structure extracting 2000 samples from the distribution $x_1 = 0.04t \sin(t) + \epsilon_1$, $x_2 = 0.04t \cos(t) + \epsilon_2$, with t distributed uniformly over the interval $(3, 15)$, and ϵ_i being Gaussian with $\mathcal{N}(0, 0.03)$. Here, we set $R_{\max} = 0.2$ and $\rho = 0.1$ (to keep $\eta = \frac{\rho}{\sqrt{r}}$ small) for both EM3A and EM3A⁺. Table 3.2 shows the mean result for 10 independent runs. While the denoising improves the performance of FPW significantly, it also leads to a sparser representation of the pdf.

In addition to the spiral dataset, we also consider the simulated jelly-fish galaxy containing 58530 particle. We use the same 10- folds created in the previous chapter, to check the impact of our denoising approach. Here, we only apply the EM3A⁺ algorithm to the data, and we use the same hyper-parameters as the cosmic web experiment. The result is reported in 3.2. It not only slightly improves the performance but also simplifies the model significantly.

Table 3.2: Density estimation

	FPW	EM3A+FPW	EM3A ⁺ +FPW
ALL	0.3004 (0.13)	0.6012 (0.04)	0.5323 (0.05)
centres	25.90 (1.29)	21.80 (0.74)	22.1 (0.74)
ALL	-3.8557 (0.02)	-	-3.8408 (0.01)
centres	952.8 (9.70)	-	748.1 (6.59)

3.5 Conclusion

The presence of noise can degrade the results of many algorithms, and data cleaning can improve the performance of subsequently applied ML algorithms. Inspired by swarm intelligence EM3A uses colonies of agents with different properties to decrease noise and concepts from evolutionary game theory to control their populations for automatic hyper-parameter optimization. In contrary to our proposed strategy, related manifold denoising techniques, such as LLD and MBMS, require information about the manifolds and noise beforehand. Since every agent in the new algorithm just uses local information for their actions it has several advantages: it is flexible, parallelizable and applicable in different scenarios including multi-manifold cases. Therefore, we applied the new algorithm on an N-body cosmological simulation where it is desirable to have an algorithm which recover low dimensional structures. The results show it successfully clean structures without requiring extra knowledge about the number of manifolds and their intrinsic dimensionalities. Moreover, it can be used as a preprocessing step for density estimators, and the experiments show it can improve the performance of density estimators and provide a sparser model.

Part II

Astronomical Observations

Published as:

M. Mohammadi, N. Petkov, K. Bunte, R.F. Peletier and F.M. Schleif – “Globular cluster detection in the *gaia* survey,” *Neurocomputing*, 2019, 342, pp.164-171.

M. Mohammadi, R.F. Peletier, F.M. Schleif, N. Petkov and K. Bunte – “Globular cluster detection in the *Gaia* survey”, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN2018).

Chapter 4

Globular cluster detection in the halo of the Milky Way

Abstract

Many techniques for the detection of interesting stellar structures in astronomical data sets require full phase-space or color information, which is not always available. The first data release of the GAIA satellite, for example, provided highly accurate positions and magnitudes for more than one billion sources. Therefore the question arises if such structures can also be automatically found without waiting for more detailed information in future data releases. In this chapter we propose and compare two conceptually different strategies to find globular clusters in the GAIA DR1 survey. The first approach is a nearest neighbor retrieval and the second one is an anomaly detection. Both techniques are able to find most of the known globular clusters within our observation frames consistently, as well as potential candidates for further investigation. Furthermore we address approximation approaches to scale the strategy to larger data.

4.1 Introduction

A major interest in astronomy is finding characteristic stellar structures in the sky that are present in large modern astronomical databases, but surrounded by foreground stars posited in our own galaxy. With more powerful sensor technology and data surveys ever increasing in size, the demand for automated techniques to find and investigate such astronomical structures has grown tremendously. The GAIA mission, for example, launched a satellite at the end of 2013 to survey the whole sky for the creation of an extraordinarily precise three-dimensional map for stars in our galaxy (the Milky Way) and beyond. Data produced by this mission provides a massive amount of observational data useful to analyze various important problems related to the origin, structure and evolutionary history of our galaxy.

Good candidates for the study of stellar evolution are structures known as globular clusters (GCs). A globular cluster is a stellar population which assumes that the involved stars were formed at the same time from a single cloud of interstellar gas and dust and therefore being of similar age and chemical compositions, which makes GCs effective tracers of galaxy formation and evolution (Barnes 1989). Moreover, the dynamical interactions of GCs with the galactic potential inferred from effects on stellar density and velocity profiles, can be used to study the mass distribution of the galaxies (Rockosi et al. 2002). However, GCs are comparably small and compact stellar structures and therefore not trivial to find.

The development in observing capabilities enables us to collect different types of observations. In the presence of diversity in measurements, we need different techniques to detect stellar structures. A few algorithms like ROCKSTAR (Behroozi et al. 2012), OPTICS (Fuentes et al. 2017) and M3A identify substructures, such as GCs, from full phase-space information. If only position, magnitudes and color information of the stars is available matched filter methods are typically applied (Tian 2017).

In this chapter we focus on the detection of GCs using the data release 1 (DR1) catalog collected by the GAIA satellite. The first release contains highly accurate positions for more than one billion stars brighter than magnitude 20.7 in the white-light photometric band G (Brown et al. 2016). However, neither the full phase-space information nor the color information of the stars is available in the GAIA DR1 catalog, and therefore the above mentioned methods cannot be used. A very recent contribution proposed the use of the magnitude distribution to detect compact stellar structures such as GCs and dwarf galaxies in GAIA DR1 (Tian 2017), which inspired this work. They introduce a histogram to approximate magnitude distribution. However, using a one-dimensional histogram has two disadvantages: a) a histogram is sensitive to the placement of bins, and b) the sole use of the magnitude distribution disregards the information induced by position.

To date, Computer Science has yielded an impressive amount of machine learning algorithms to collect, store, retrieve, visualize and analyze complex interdisciplinary data, which have repeatedly proven useful in decision support and discovery in many fields (Bishop 2006b). In this chapter, we investigate the automatic detection using machine learning when only limited information is available. For the detection we follow two strategies: a retrieval technique using known GCs to detect similar structures and a novelty detection technique using the fact that most of the sky does not contain any GCs. The main reason to select these strategies is that both methods are suitable in situations where the availability of examples of the interesting pattern is limited. In summary, this chapter provides the following three major contributions:

- A globular cluster detection technique following two independent strategies, namely similarity retrieval and novelty detection.
- 3D Kernel Density Estimation (KDE) as feature vectors in order to capture both position and brightness information.
- Discussion of the computational costs and approximation schemes to scale the approach to larger data sets.

The chapter is structured as follows: in section 4.2, we explain the main properties of globular clusters and why they are important for investigation. The data and preprocessing is addressed in section 4.3. In section 4.4 we explain the retrieval and novelty detection techniques followed by a complexity analysis in section 4.5. The effectiveness of the strategies is demonstrated in detail in the experimental section 4.6, followed by the results in section 4.7 and the chapter is finalized with conclusions.

4.2 Laboratory for stellar evolution

A globular cluster (GC) is a spherically shaped collection of stars bounded by gravity with high stellar densities toward its center. A star is a luminous spheroid of gas, mostly hydrogen and helium, held together by its own gravity. To measure its brightness, the apparent magnitude m is computed by

$$m = -2.5 \log_{10} F + c, \quad (4.1)$$

where F denotes the flux of the star received on Earth and c represents a photometric constant. If the magnitude values of stars in two bands are available, we can calculate their color which are closely related to their surface temperature. The Hertzsprung-Russell (HR) diagram uses the relationship between the apparent magnitude (or luminosity) and color to illustrate the evolution of stars.

Figure 4.1 depicts an example HR diagram containing the magnitude and color information of stars within the globular cluster NGC 5024. Within a GC we can find stars in different stages of their life which makes them good candidates to study stellar evolution. Depending on the distance of the cluster, the position of horizontal branch will shift in the apparent magnitude axis which can be described by the distance modulus formula:

$$m - M = 5 \log_{10} d - 5, \quad (4.2)$$

where m , M and d are apparent magnitude, absolute magnitude and the distance (in parsec), respectively. The absolute magnitude of a star is its intrinsic brightness

Figure 4.1: The HR diagram of the globular cluster NGC 5024. Important groups are marked by RGB: Red Giant Branch, HB: Horizontal Branch, MS: Main Sequence.

Figure 4.2: Left: Magnitude distribution of a window of 0.5 degree centering at globular cluster NGC5024. Right: Magnitude distribution of a random window.

(in magnitude) which is expected to be $M = -0.5$ for a star in HB (Riello et al. 2003). Hence, finding the distances of GCs takes us one step closer to the goal of the GAIA mission of creating a 3-dimensional map of the Milky Way.

4.3 Data and Preprocessing

The GAIA DR1 catalog captures stars of magnitudes up to 20.7 magnitude (mag). We use the magnitude distributions of the stars contained in rectangular windows to detect globular clusters in the GAIA database. As proposed in (Tian 2017) and (Mohammadi et al. 2018) we consider windows of 0.5 degree length in both right ascension (RA) and declination (Dec). The left panel in Fig. 4.2 displays the magnitude distribution as estimated via one dimensional Kernel Density Estimation (KDE)

Figure 4.3: Summary of preprocessing: 1) divide the sky into smaller windows, 2) extract sources centered around the region of largest density and 3) compute 3D-KDE of the extracted data.

based on the magnitudes extracted from a window centered at position (198.23 RA, 18.17 Dec) of globular cluster NGC5024 in GAIA. In contrast to random windows (Fig. 4.2 right panel) the distribution of a window containing a globular cluster will exhibit a bump and then a sharp increase which is expected if we collapse the HR diagram in the color dimension (when color information is not available). By comparing Fig. 4.1 and Fig. 4.2, we see a clear bump at a characteristic magnitude 17, which comes from the horizontal branch (HB). For higher magnitudes, we see a sharp increase towards faint stars, up to the magnitude limit of 20.7 mag in the GAIA survey.

4.3.1 Preprocessing

The preprocessing procedure we perform throughout the experiments is summarized in Fig. 4.3. First we divide the sky into smaller windows of size 0.5×0.5 degree² to ensure each region is taken into account for the analysis. For each bright source, we have three numbers which represent its position (right ascension and declination) and its apparent magnitude. Since the center of GCs are the most dense place in comparison to stars far from the center, we approximate the position of the center finding the most dense place in each window. Then we extract the information (position and apparent magnitude values) of sources contained in the bins of

Figure 4.4: Comparison of densities using Euclidean distance and Dynamic Time Warping.

size 0.17×0.17 degree² around the center.

Suppose $(x_1, \dots, x_n)$ is the extracted magnitude values from a window, we can approximate the magnitude distribution by one dimensional KDE as follows:

$$f_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (4.3)$$

with bandwidth h and the Epanechnikov function as kernel K (Chu et al. 2017). Here, we extend the original proposal in (Mohammadi et al. 2018) to use the full position information. This can be achieved by using three dimensional KDE (Simonoff 2012). Since different coordinates have different units, we need to allow different amount of smoothing in each of the coordinates. Hence, we use 3D KDE which relies upon the product kernels with different bandwidth for each coordinates, as follows:

$$f_H(x) = \frac{1}{nh_1h_2h_3} \sum_{i=1}^n K\left(\frac{\text{ra} - \text{ra}_i}{h_1}\right) K\left(\frac{\text{dec} - \text{dec}_i}{h_2}\right) K\left(\frac{x - x_i}{h_3}\right) . \quad (4.4)$$

The variables $(\text{ra}_i, \text{dec}_i)$ denote the position of i -th star inside the window and $H = (h_1, h_2, h_3)$ depicts the bandwidths along right ascension, declination and magnitude axes, respectively.

4.4 Methods

We compare the magnitude densities extracted from a window regarding each magnitude distribution as a sequence (replacing time by magnitude values) and make use of Dynamic Time Warping (DTW) (Sakoe and Chiba 1978), (Schleif and Tino 2015). Consider the two D -dimensional sequences $X = x_1, x_2, \dots, x_m$ and $Z =$

$z_1, z_2, \dots, z_n$ where $x_i, z_j \in \mathbb{R}^D$. DTW aligns them non-linearly (see right panel in Fig. 4.4) in such a way to minimize the distance between them. The alignment can be represented by a sequence of tuples:

$$F = a_1, a_2, \dots, a_K \quad (4.5)$$

where $a_k = (i_k, j_k)$ contains the indices from X and Z with $i_k \in \{1, 2, \dots, m\}$, $j_k \in \{1, 2, \dots, n\}$. Hence, it assigns an element of X to an element of Z . The sequence F is called warping path, and it should fulfill the following conditions:

- Boundary condition: alignment of the first and last elements of sequence, i.e. $a_1 = (1, 1)$ and $a_K = (m, n)$.
- Monotonicity condition: the assignment can not be backward in time, i.e. if we have $k < l$, then $i_k \leq i_l$ and $j_k \leq j_l$ should hold.
- Step size condition: no element of X and Y can be omitted, and all tuples of F should be distinct.

Given a warping path, we can measure the dissimilarity between them as follow:

$$d(X, Y) = \sum_{k=1}^K \|x_{i_k} - z_{j_k}\| \quad (4.6)$$

where $\|\cdot\|$ is the euclidean distance. The optimal warping path with the lowest distance d can be achieved by warping the time axes, which can be efficiently obtained using dynamic programming (Sakoe and Chiba 1978). However, the resulting dissimilarity does not fulfill all metric properties, but it has the potential to detect similar structures in the densities, such as bumps, despite their shifting location on the magnitude axis.

4.4.1 Nearest Neighbor Retrieval

Nearest Neighbor Search (NNS) finds similar objects of interest in a data base. For a given example, this technique returns a desired number of most similar objects from the data base according to the distance measure used. By using this strategy we aim to find interesting structures in the GAIA survey by retrieving similar densities to those extracted from the known globular clusters using DTW as dissimilarity measure. To compare the extracted distribution with a k nearest neighbors of the known structures, the number of common GCs for each window is counted. The higher the obtained count the more likely it is that there is something interesting in it. This technique can be easily extended to deal with even larger data sets by using

approximate nearest neighbor search approaches, as for example locality-sensitive hashing(Leskovec et al. 2014), best bin first(Beis and Lowe 1997) and balanced box-decomposition tree based search(Arya et al. 1998).

4.4.2 Novelty Detection

Since the sky is mostly empty (of structures) we can regard the search for interesting structures as novelty or outlier detection. We use an approach which was inspired by the Support Vector Classifier called Support Vector Data Description (SVDD) (Tax and Duin 1999). Assume given data $\mathbf{x}_i \in \mathbb{R}^n$ and a fixed chosen kernel $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ which is associated to the feature map Φ : $K(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^t \Phi(\mathbf{z})$.

The motivation for such an embedding comes with the hope that the non-linear transformation of input data into a higher dimensional space allows for using linear techniques. Kernelized methods process the embedded data points in a feature space utilizing only the inner products $\langle \cdot, \cdot \rangle$ (kernel trick) (Shawe-Taylor and Cristianini 2004), without the need to explicitly calculate Φ . The kernel function can be very generic. Most prominent are the linear kernel with $K(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$ where $\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$ is the Euclidean inner product and Φ identity mapping, or the RBF kernel $K_\sigma(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}\right)$, with $\sigma > 0$ as a free scale parameter. In any case, it is always assumed that the kernel function $K(\mathbf{x}, \mathbf{z})$ is positive semi definite (psd).

The goal of SVDD is to find a generalized linear mapping $\mathbf{x} \mapsto \text{sgn}(f(\mathbf{x})) = \text{sgn}(\mathbf{w}^t \Phi(\mathbf{x}) - \rho)$, which defines a separation of the given data to outliers by means of its sign. Where the separation boundary corresponds to a linear separation in the feature space induced by Φ . The problem to find suitable parameters $\mathbf{w}$ and ρ for a given data set of typical points $\mathbf{x}_i$ can be formalized as an optimization problem which aims at a separation of the given data from the origin with maximum margin. This leads to the following primal optimization problem SVDD (primal): $\min_{\mathbf{w}, \rho, \xi_i} \frac{1}{2} \cdot \|\mathbf{w}\|^2 - \rho + \frac{C}{2} \cdot \sum_i \xi_i^2$, such that $\mathbf{w}^t \Phi(\mathbf{x}_i) \geq \rho - \xi_i \quad \forall i$, where $C > 0$ is a fixed constant, and the parameters ξ_i refer to the slack variables to allow for some error tolerance. Using the Karush-Kuhn-Tucker conditions, the Lagrange dual problem becomes

$$\begin{aligned} &\text{SVDD(dual)} \\ &\max_{\alpha_i} \quad -\frac{1}{2} \cdot \sum_{ij} K(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j + \frac{1}{C} \cdot \sum_i \alpha_i^2 \\ &\text{such that} \quad \alpha_i \geq 0 \quad \forall i \\ &\quad \quad \quad \sum_i \alpha_i = 1 \end{aligned}$$

This dual problem can be directly optimized relying on linearly constraint convex quadratic optimization. The solution $\mathbf{w}$ and ρ of the primal problem can then be

recovered from the dual variables α_i . These are non-vanishing for support vectors only and hence result in a sparse description. For large data sets it is possible to use optimized SVDD solvers or to apply the equivalent core vector data description technique (CVDD), which effectively scales to millions of points (Tsang et al. 2006).

The tightness of the data description can be controlled by setting an error tolerance or using outlier examples one desires to detect. However, for this approach we require a positive definite kernel also known as Mercer kernel (Pekalska and Duin 2005), which we cannot guarantee if the DTW dissimilarity is used due to missing metric properties. To obtain a kernel (similarities) from a dissimilarity measure (like DTW) it is possible to use a technique called double-centering (Pekalska and Duin 2005):

$$S = -\mathbf{J}D\mathbf{J}/2 \text{ with } \mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n) \quad (4.7)$$

with D being a (squared) dissimilarity matrix as obtained from a given dissimilarity function and identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. However in case of DTW the obtained kernel will be non-psd.

Non-metric dissimilarity measures and non-psd, or indefinite kernels have recently attracted a lot of research activities, with a summary given in (Schleif and Tino 2015). These measures have often found to be very effective and more flexible than standard metric measures. Most kernel methods are restricted to psd (mercer) kernels and a variety of transformation approaches have been suggested to convert non-psd kernels into psd-kernels (Schleif and Tino 2015).

A natural way to address this problem and to obtain a psd similarity matrix is to correct the eigenspectrum of the original similarity matrix S . Popular strategies include *flipping*, *clipping*, *shift correction*. The non-psd similarity matrix S is decomposed as

$$S = U\Lambda U^\top, \quad (4.8)$$

where U contains the eigenvectors of S and Λ contains the corresponding eigenvalues. The following approaches are commonly used:

- clipping - the negative eigenvalues in Λ are fixed to 0
- flipping - the sign of the negative eigenvalues in Λ is changed to positive
- square - all eigenvalues are squared
- shift - the eigenspectrum (from Λ) is shifted by the most negative eigenvalue value, such that all entries in Λ are non-negative.

Spectrum shift enhances all the self-similarities by the amount of ν and does not change the similarity between any two different data points. Detailed equations, further transformation techniques as well as an in depth analysis of the various methods, is provided in (Schleif and Tino 2015). We approximate a positive definite kernel (psd) based on the DTW dissimilarity following the strategy discussed in (Schleif and Tino 2015) using the *square approach*, which was found to be of good performance in a number of tasks. We follow two steps to convert the DTW pairwise dissimilarity matrix D between magnitude distributions into an appropriate kernel as input to the SVDD:

1. Constructing the dissimilarity matrix by computing DTW for all pairs of magnitude distributions
2. computing similarity matrix S from D via double centering and
3. converting S to a psd kernel by $K = S \cdot S^\top$.

For large scale data this approach can be optimized by using additional methods detailed in (Gisbrecht and Schleif 2015).

4.5 Complexity analysis

The alignment distances can be efficiently calculated using dynamic programming with a worst-case runtime of $\mathcal{O}(L_1 \cdot L_2)$, with L_1 and L_2 being the number of spectrum elements in the first and the second input spectrum, respectively. The conversion from dissimilarities to positive-semi-definite similarities can be achieved with linear cost in $\mathcal{O}(n)$ using an approximation strategy (including an effective out-of sample extension) suggested in (Gisbrecht and Schleif 2015).

The application of the nearest neighbor classifier has linear complexity in $\mathcal{O}(N)$, but can also be reduced to potentially log-linear costs using e.g. k-d-trees (Skiena 1998). The SVDD algorithm has roughly $\mathcal{O}(n^2)$ complexity but using the CVDD approach the complexity can be reduced to $\mathcal{O}(n)$. All together using specific optimization concepts and data structures, the NN approach has $\mathcal{O}(\log n)$ runtime and $\mathcal{O}(n)$ memory complexity in the test phase. With NN a training is not needed. Using the SVDD / CVDD technique we get $\mathcal{O}(n)$ runtime and memory complexity during training and due to sparsity and approximation concepts sub-linear memory and runtime complexity during testing. In the presented results we have used exact approaches only.

4.6 Experiments

Figure 4.5: Number distribution of all the stars in Gaia DR1 with $10 < G < 20.7$. Boxes marked three selected regions with the positions: (1) $0^\circ < \alpha < 80^\circ$ and $-90^\circ < \delta < 0^\circ$, (2) $120^\circ < \alpha < 246^\circ$ and $-2^\circ < \delta < 60^\circ$, and (3) $310^\circ < \alpha < 360^\circ$ and $-90^\circ < \delta < 0^\circ$.

We apply the proposed strategies on three regions around the disk of Milky Way as depicted in Fig. 4.5 (Tian 2017). As explained in section 4.3.1, the sky is divided into smaller windows of 0.5 degree bin width. Then we find the most dense place in each window and extract information of sources in the neighborhood centered at it with the size of 0.17 degree. We use the 3D-KDE with bandwidth $H = (0.1, 0.1, 0.15)$. And we extract the KDE values in a regular grid of size $3 \times 3 \times 500$: 3 along RA and Dec axes and 500 along magnitude axis, respectively. The resulting grid is reshaped as a sequence of vectors ($v_i \in \mathbb{R}^9, i = 1, \dots, 500$).

There are 26 known globular clusters in the selected regions. Since we have a limited magnitude of stars of maximal 20.7 in the GAIA survey, we do not include the more distant known GCs, which cannot be expected to be captured in the data. A total of 17 known GCs remains for the analysis in our experiments (see Table 4.1 (Harris 1996)).

Nearest Neighbor Retrieval For the nearest neighbor retrieval we use the magnitude distributions of the 17 known GCs as example structures. All these examples are then compared with DTW to windows of equal size extracted by sliding with 0.5 degree through the parts of the sky under observation. Using the counting strategy discussed above the promising candidates are identified. To evaluate this technique

Table 4.1: Names and positions of the known GCs in the selected regions.

Name	Position		Region
	right ascension	declination	
NGC 1851	78.53	-40.05	1
47 Tuc	6.02	-72.08	1
NGC 288	13.19	-26.58	1
NGC 362	15.81	-70.85	1
NGC 1261	48.07	-55.22	1
NGC4147	182.53	18.54	2
NGC 5024	198.23	18.17	2
NGC5053	199.11	17.7	2
NGC 5272	205.55	28.38	2
NGC5466	211.36	28.53	2
NGC5904	229.02	-0.11	2
Palomar5	229.64	2.08	2
M72	313.37	-12.54	3
M2	323.36	-0.82	3
M30	325.09	-23.18	3
Palomar 12	326.66	-21.25	3
NGC 7492	347.11	-15.61	3

we use 4-fold cross validation. Due to the relatively small number of GCs (seventeen), we divide them into four non-overlapping folds, such that the method is tested on structures not used for training. Then DTW is used to find the most similar distributions in the test set to queries in the training set.

Novelty Detection For the outlier detection we build 5 psd kernels for cross validation based on the DTW dissimilarities as mentioned before. As a proof of concept we build kernels of 6000 samples (2000 per region) each containing:

- a) a training subset of 4200 random windows taken from GAIA (1400 per region) resembling empty sky¹ and
- b) a test set containing the windows extracted around the known GCs, potential candidates from the retrieval and 600 random windows per region.

¹The probability of accidentally picking a window containing an interesting structure is very low and the SVDD has a hyper-parameter to allow outliers.

Figure 4.6: Workflow of the process: at first, the preprocessing step is done (section 4.3.1), then we use two different approaches: Nearest neighbor retrieval (section 4.4.1) and Novelty detection (section 4.4.2). At the end, DoG (section 4.6) is used for the candidates obtained by two methods.

Therefore, we train 5 SVDD models on different random subsets of empty sky to get models depicting typical uninteresting windows. To evaluate our method, we use the testing set (b) and report the detection rate of known GCs as well as overlap with the retrieval method. In our experiments we investigate the threshold ρ for the decision boundary and therefore the tightness of the data description by an error tolerance E .

Difference of Gaussian filters A difference of Gaussian (DoG) filter (Gecer et al. 2017) is constructed by subtraction of a Gaussian filter from a tighter one as follows:

$$DoG_{\sigma}(x, y) = \frac{\exp\left(-\frac{x^2+y^2}{2(0.5\sigma)^2}\right)}{2\pi(0.5\sigma)^2} - \frac{\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^2}, \quad (4.9)$$

where σ is standard deviation. This filter gives high response to a circular shape. To determine the exact position of a GC and to reduce false positives we use DoG filters to judge the candidates. The work-flow of the process is depicted in Fig. 4.6.

4.7 Results

Both methods retrieve an increasing number of potential candidates with increasing their corresponding hyper-parameter. The summary of the results is given in Table 4.2. Since the retrieval technique uses examples of the GCs, it is more specific in comparison to the outlier detection strategy. To evaluate the retrieval technique, we

Table 4.2: Mean hit number (and std) for different hyper-parameter settings k and E in retrieval and outlier detection respectively.

HP	Candidates		GCs found	
k 10	79.25	(11.18)	8.75	(1.26)
k 20	135.75	(19.00)	11.00	(1.41)
k 50	309.25	(31.06)	12.50	(0.58)
k 100	540.00	(51.20)	14.00	(0.00)
k 250	1174.00	(101.78)	16.00	(0.00)
E 1%	90.20	(52.82)	14.40	(0.55)
E 5%	243.80	(37.93)	15.60	(0.89)
E 10%	305.80	(20.05)	16.80	(0.45)
E 20%	318.80	(27.81)	17.00	(0.00)

use 4-fold cross validation to see whether it can detect the known GCs (see the first 5 rows of Table 4.2). For this technique, 47 Tuc, NGC 362 and Palomar 5 are more difficult. This is due to the fact that the radii of 47 Tuc and NGC 362 are big (even bigger than 0.25 deg), making them difficult to detect using smaller windows (i.e. 0.17 deg). However, we still can detect them if we increase the hyper-parameters or the size of the windows. The average vote for $k = 40$ nearest neighbors and corresponding candidates together with known GCs in the investigated regions are shown in Fig. 4.7. Windows with more than one third of the maximum vote are clustered based on Ra and Dec and hits closer than 0.75° denote the same candidate.

Using the outlier detection strategy models are trained solely on random windows. Therefore the models are not provided with any example of the known structures, but still detect almost all 17 known GCs used for testing when allowed an error tolerance $E \geq 10\%$. Furthermore it is important to note that the outliers found often overlap with candidates detected using the retrieval method.

To reduce false positives, specially for the outlier detection, we use DoG filters (Gecer et al. 2017) in a post processing step. The right panels of figure 4.8 shows filter responses to two potential interesting candidates found by our strategies at different magnitudes stacked on top of each other. On the left side we show original magnitudes of sources at the corresponding RA and Dec position taken from the GAIA DR1 survey. We observe that the filter starts giving high responses in the center of the structure (i.e. a circular shaped group of stars) from some magnitude up to limit magnitude of GAIA, but a weak response (i.e. uniformly distributed stars) at lower magnitude, which can be inferred that there are fewer bright stars

Figure 4.7: The average votes for the candidate using 40 NN retrieval and the circles mark interesting regions with the color indicating the similarity to known GCs.

and more faint stars in these structure.

4.8 Conclusion and Outlook

Techniques to detect interesting stellar structures often require full-phase space or color information, which is not always available especially in early releases of astronomical data. In this chapter two independent strategies are presented to find inter-

Figure 4.8: Two candidates found at positions $(40, -34.5)$ and $(330.78, -51.3)$. Left: the distribution of stars and gray levels representing the magnitude values of sources from GAIA DR1. Right: stacked responses of spatial Difference of Gaussian filter at different magnitudes.

esting stellar structures targeted at astronomical surveys which provide only limited information such as 2D positions and brightness. Previous work (Tian 2017) uses histograms as features and manual inspection to retrieve possible candidates. In comparison we use three dimensional KDEs as features and automatic retrieval and novelty detection techniques to find possible candidates. The methods are demonstrated for the detection of globular clusters in the GAIA DR1 survey. Using a nearest neighbor retrieval method we compare examples extracted from 17 known GCs to other parts of the sky. Furthermore, assuming the sky is mainly empty (of interesting structures with the magnitude values smaller than 20.7) we treat the problem of finding interesting structures as outlier detection problem. Therefore we used some random windows of the sky to train a support vector data description, a kernel-

based anomaly detection technique. Both techniques find most of the 17 known GCs used for testing and also depict overlapping agreement on potential candidates found. Since these methods could give false positive candidate which is the case when their hyper-parameters are big, we use Difference of Gaussian filters on the candidate. However, since the GAIA DR1 does not provide the color information, it is still possible to get false positive. For example, it is possible to find some other types of structures, such as dwarf galaxies because some of them have similar magnitude distributions and also spherical shapes. Recently, the new version of GAIA survey (GAIA DR2) has been released which includes color information and proper motions, and let us to use our proposed technique (M3A) for further investigation.

Based on:

M. Mohammadi, J. Mutatiina, T. Saifollahi, K. Bunte – “Detection of extragalactic Ultra-Compact Dwarfs and Globular Clusters using Explainable AI techniques,” Manuscript submitted.

Chapter 5

Detection of extragalactic Ultra-Compact Dwarfs and Globular Clusters using Explainable AI techniques

Abstract

Compact stellar systems such as Ultra-compact dwarfs (UCDs) and Globular Clusters (GCs) around galaxies are known to be the tracers of the merger events that have been forming these galaxies. Therefore, identifying such systems allows astronomers to study galaxies mass assembly, formation and evolution. However, in the lack of spectroscopic information detecting UCDs/GCs using imaging data is very uncertain. Here, we aim to train a machine learning model that is able to separate these objects from the foreground stars and background galaxies using the multi-wavelength imaging data of the Fornax galaxy cluster in 6 filters, namely u, g, r, i, J and Ks. The classes of objects are highly imbalanced which is problematic for many automatic classification techniques. Hence, we employ Synthetic Minority Over-sampling (SMOTE) to handle the imbalance of the training data. Then, we compare two classifiers, namely Localized Generalized Matrix Learning Vector Quantization (LGMLVQ) and Random Forest (RF). Both methods are able to identify UCDs/GCs with a precision and a recall (sensitivity) of $> 93\%$ and provide relevances that reflect the importance of each feature dimension (colors and angular sizes) for the classification. Both methods detect angular sizes as important markers for this classification problem. While it is astronomical expectation that color indices of $u-i$ and $i-Ks$ are the most important colors, our analysis shows that colors such as $g-r$ are more informative, potentially because of higher signal-to-noise ratio. Besides the excellent performance the LGMLVQ method allows further interpretability by providing the feature importance for each individual class, class-wise representative samples and the possibility for non-linear visualization of the data as demonstrated in this chapter. We conclude that employing machine learning techniques to identify UCDs/GCs can lead to promising results. Especially transparent methods, as presented here, allow further investigation and analysis of importance of the measurements for the detection problem and provide tools for non-linear visualization of the data.

5.1 Introduction

Based on the current theories of galaxy formation and evolution, galaxies are formed hierarchically from the merger of low-mass galaxies that were formed earlier. In this picture, the dense stellar structures such as Ultra-compact dwarfs (UCDs) galaxies and Globular clusters (GCs), which are mostly found around galaxies and the core of galaxy clusters, are known to be the tracers of such merging events (Beasley 2020). However, extragalactic UCDs and GCs around other galaxies than the Milky Way look like stars (point-sources), due to their distance and current limitations of observational equipment. Therefore identifying them through imaging is challenging (Jordán et al. 2009). To find these objects, it is necessary to measure their distances, which is only possible using spectroscopy and measuring their radial velocity. The Hubble-Lemaître Law says that more distant galaxies move faster away from us and thus astronomers measure the distance of galaxies by measuring their radial velocity. The latter can be measured using Doppler shifts of the absorption lines in the spectroscopic observations. Spectroscopy of astronomical objects, however, needs longer exposure times than imaging. In other words, spectroscopy for all the star-like objects (point-sources) is not feasible in practice (Voggel et al. 2020).

The recent advances in astronomical instrumentation and observations, without doubt, have provided us with a large amount of data to explore. Traditionally, the possible UCDs/GCs candidates are identified by multi-wavelength observations in a few optical filters (Cantiello et al. 2018). Once the candidates are found follow-up spectroscopy for selected nominees is carried out to measure the radial velocity and therefore the distance to confirm the identity of the objects (Pota et al. 2018). This implies that a more accurate UCD/GC selection makes the observation time shorter.

In the case of Fornax galaxy cluster, the second closest galaxy cluster to us, recent observations of optical and near-infrared have been available, which makes it possible to identify UCDs/GCs within the galaxy cluster. The optical part of the data has been used earlier to identify GCs in the cluster using various techniques (Cantiello et al. 2020, Prole et al. 2019, Angora et al. 2019). Recently, (Muñoz et al. 2014) has shown that a combination of optical/near infrared filters improves the quality of identifying UCDs/GCs. However, this approach was not used until very recently, mostly because deep observations in the near-infrared are not as easy as the optical.

Due to the sheer amount of astronomical data automatic tools to analyse the data are highly desirable. Therefore, machine learning techniques get more and more attention among Astronomers recently, and they have been popularly explored for astronomical applications. The Random Forest (RF) (Carrasco et al. 2015, Gao et al. 2009) has been used to build a classifier for quasar identification and the Support

Vector Machine (SVM) (Jones and Singal 2017) has been employed to estimate the redshift. Other techniques used are k-nearest neighbor classifier (Li et al. 2008) for active galactic nuclei (AGN) detection, Support Vector Data Description (SVDD) (Mohammadi et al. 2019) for GC detection, and Artificial Neural Networks (ANN) (Ball et al. 2004, Barchi et al. 2020, Xiao et al. 2020), for galaxy morphological classification and AGN detection. Thus, a plethora of machine learning methods have been successfully applied in many astronomical applications and they encourage us to be extended to other cases.

A well-known family of prototype-based classifiers is Learning Vector Quantization (LVQ) that efficiently distinguish high dimensional multi-class problems. One of the main advantages of LVQ classifiers is the interpretability of the adaptive parameters. The learned prototypes, for example, can be investigated as typical representatives of the classes. While the original formulations of the LVQ family, such as *Generalized learning Vector Quantization* (GLVQ) (Sato and Yamada 1995) use the Euclidean distance, more complex extensions, such as Generalized Matrix LVQ (GMLVQ) and the localized version LGMLVQ (Schneider et al. 2009b, Schneider et al. 2009a) employ adaptive distance metrics. The latter algorithms also allow further insights by visualization of the decision boundaries after training (Bunte et al. 2012, Bunte 2011). In this chapter, we use LGMLVQ to classify three astronomical structures, namely foreground stars, UCDs/GCs and background galaxies, based on their optical (u, g, r, i) and near-infrared (J, K_s) measurements of the Fornax Deep Survey, VISTA Hemisphere Survey and ESO/VISTA archive. The method constructs non-linear decision boundaries and allows to evaluate the importance of features for each class individually. One major issue often faced in astronomical problems is the imbalance of the classes. The total number of known UCDs and GCs in the Fornax cluster identified in the data is just over 500, whereas the majority class contains about 5 times more instances. To tackle this challenge we apply over-sampling techniques, such as Synthetic Minority Oversampling (SMOTE) (Chawla et al. 2002) and Borderline SMOTE (Han et al. 2005). In contrast to previous works we have both optical and near-infrared filters in the dataset. We use LGMLVQ to detect the classes of objects in potentially large amounts of high dimensional astronomical data and analyse the results. Moreover, we compare the performance using an ensemble of LGMLVQ classifiers.

The paper is organized as follows: In section 5.2 we describe the dataset, followed by the explanation of classifiers in section 5.3. Afterwards, in section 5.4 we describe the experiments and discuss the results. Finally we conclude in section 5.5 and provide inspirations for future work.

5.2 Astronomical Data and Preprocessing

The data used in this research is extracted from multi-wavelength wide astronomical surveys obtained from a combination of 6 filters, i.e optical filters (u, g, r and i) and near-infrared filters (J and Ks). The optical u, g, r, i data was obtained from Fornax Deep Survey (FDS) using the ESO VLT survey telescope (VST), J from VISTA Hemisphere Survey (VHS, (McMahon et al. 2013)) using the VISTA telescope and Ks from the ESO/VISTA archival data. This data is described in detail in Saifollahi et al. (submitted to Monthly Notices of the Royal Astronomical Society). The data set provides photometric information in the direction of the Fornax galaxy cluster. After excluding larger objects, likely to be galaxies in the Fornax cluster or slightly more distant, any observed object in the dataset belong to one of the following 3 classes:

1. consists of 2826 background galaxies further away than the galaxies in the Fornax cluster,
2. denotes the class of interest consisting of 512 UCDs and GCs, and
3. contains 4399 nearby foreground stars located in our own galaxy, the Milky Way.

These classes are difficult to distinguish for several reasons: (1) the UCD and GC (class 2) observations are faint and ambiguous as they are engulfed between the two other classes, and (2) there is only a small number of confirmed examples of them.

The labelled data consists of 23 features. The coordinates of the objects in the sky are given in right ascension (RA) and declination (DEC) as a degree of point of observation from earth and illustrated in Fig. 5.1(c). Features $FWHM^*g, FWHM^*r, FWHM^*i, FWHM^*u, FWHM^*j$ and $FWHM^*k$, also known as the Full width half maximum, are the proxies for the angular sizes of the objects as seen from the corresponding filters. Followed by $u - g, u - r, u - i, u - J, u - Ks, g - r, g - i, g - J, g - Ks, r - i, r - J, r - Ks, i - J, i - Ks$ and $J - Ks$, which are the colour indices indicating the emission of the astronomical object in logarithmic scale, known as magnitude and correlated to physical properties like age and metallicity. In total, the data consists of 7737 complete observations that are used in the analysis.

Figure 5.1 shows the Eigenvalue profile of the data obtained using the unsupervised Principal Component Analysis (PCA) in panel (b). The two major Eigenvalues explain 88% of the data's variance and the corresponding two-dimensional projection is depicted in panel (a). The large areas of overlap and imbalance of the classes is clearly visible. Especially the latter states a problem for most classification techniques and thus re-sampling techniques, such as Synthetic Minority Over-sampling

Figure 5.1: PCA-projection of the data colouring the background galaxies, UCDs/GCs and foreground stars (blue, red and green, classes 1-3), respectively (panel a). Corresponding Eigenvalue profile (panel b) and position of the objects in the sky in right ascension (RA) and declination (DEC) (panel c).

(SMOTE) (Chawla et al. 2002) and Borderline-SMOTE (Han et al. 2005), are investigated as preprocessing steps. SMOTE increases the population of the minority classes by generating synthetic samples as weighted convex combination between random samples and its nearest neighbours. Instead of choosing random samples Borderline-SMOTE specifically takes points near the boundaries between the classes which are more likely to be misclassified. In this chapter we investigate and compare both methods as preprocessing to balance the classes.

5.3 Methods

In this section, we introduce the state-of-the-art methodology used with focus on localized adaptive distance metrics in Learning Vector Quantization (LVQ) and corresponding interpretability. Moreover, we present a comparison between Random Forest (which is an ensemble of Decision Trees) and an ensemble of the Learning Vector Quantization models.

5.3.1 Learning Vector Quantization (LVQ)

We assume $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote the training set, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \dots, C\}$ represent i -th data point and its class label, respectively. An LVQ classifier models the distribution of classes via a set of labelled prototypes $\{(\boldsymbol{\omega}^j, c(\boldsymbol{\omega}^j))\}_{j=1}^m$, where $c(\boldsymbol{\omega}^j)$ is the label of the respective prototype. These prototypes tessellate the data space into smaller regions, called Voronoi cells, enclosing data points for which the

respective prototype is closer than any other. Classification follows a *nearest prototype scheme*, meaning any data point (including new ones) is assigned the class label of the nearest prototype. To find prototype positions that minimize the classification error E , *Generalized learning Vector Quantization* (GLVQ) (Sato and Yamada 1995) introduced the following cost function, aiming at large margin optimization for better generalization:

$$E = \sum_{i=1}^N \Phi(\mu_i) \quad \text{with} \quad \mu_i = \frac{d(\mathbf{x}_i, \boldsymbol{\omega}^J) - d(\mathbf{x}_i, \boldsymbol{\omega}^K)}{d(\mathbf{x}_i, \boldsymbol{\omega}^J) + d(\mathbf{x}_i, \boldsymbol{\omega}^K)}, \quad (5.1)$$

with Φ being a monotonically increasing function. We used the identity $\Phi(x) = x$ throughout this chapter. Furthermore, $d(\mathbf{x}_i, \boldsymbol{\omega}^J)$ denotes the distance of the data point $\mathbf{x}_i$ from the closest prototype with the same label $y_i = c(\boldsymbol{\omega}^J)$ and $d(\mathbf{x}_i, \boldsymbol{\omega}^K)$ the distance to the closest prototype with a different class label. The value of $\mu_i \in [-1, 1]$ shows the confidence of the classification. The cost function is non-convex and typically gradient techniques, such as stochastic gradient descent (Bunte 2011, Schneider et al. 2009a) are utilized to minimize the costs E .

From Eq. (5.1) it is clear that the distance measure d plays a major role for the performance of LVQ classifiers. While the Euclidean distance is a common choice all dimensions contribute equally in it, which has drawbacks in capturing underlying data semantics (Schneider et al. 2009a) in noisy high-dimensional and heterogeneous data spaces. As such it is not capable to reflect if features differ in importance for the classification task at hand. Therefore, (Hammer and Villmann 2002) proposed to incorporate a weighting factor for each feature dimension that is adapted during training:

$$d^\Lambda(\mathbf{x}_i, \boldsymbol{\omega}^j) = (\mathbf{x}_i - \boldsymbol{\omega}^j)^\top \Lambda (\mathbf{x}_i - \boldsymbol{\omega}^j), \quad (5.2)$$

where the weight matrix Λ , also referred to as *relevance matrix*, is a diagonal matrix with 0 in the off-diagonals and λ_i on the diagonal with $\sum_i \lambda_i = 1$. These relevance weights indicate the discriminative contribution of each feature dimensions, which could facilitate decreasing influence or pruning of redundant, noisy or ambiguous feature dimensions. This concept can be further extended to more complicated metric tensors with adaptive off-diagonal elements, namely Generalized Matrix LVQ (GMLVQ) (Schneider et al. 2009b, Schneider et al. 2009a), Limited Rank Matrix LVQ (LiRaM LVQ) (Bunte et al. 2012, Bunte 2011) and localized versions with prototype-wise or class-wise matrices called Localized GMLVQ (LGM-LVQ) (Schneider et al. 2009b). All algorithms are made publicly available in Matlab and can be found at https://github.com/kbunte/LVQ_toolbox.

The overlapping class regions as shown in the PCA projection Figure 5.1(a) intuitively suggest non-linear class boundaries and hence the localized adaptive metrics

are more suitable and therefore the focus for this paper. Local metric tensors allow to learn localized dissimilarities with respect to the specific class prototypes using a local transformation matrix Ω^j thus defining the non-linear decision boundaries. The localized distance metric is defined as:

$$d^{\Lambda^j}(\mathbf{x}_i, \boldsymbol{\omega}^j) = (\mathbf{x}_i - \boldsymbol{\omega}^j)^\top \Lambda^j (\mathbf{x}_i - \boldsymbol{\omega}^j) , \quad (5.3)$$

where $\Lambda^j = \Omega^j \Omega^j$ using the adaptive matrix $\Omega^j \in \mathbb{R}^{M \times D}$ with $M \leq D$ to guarantee that Λ^j is positive semi-definite. The cost function therefore reads as follows:

$$E_{\text{LGMLVQ}} = \sum_i^N \Phi(\mu_{\text{local}}^i) \quad \text{where} \quad \mu_{\text{local}}^i = \frac{d^{\Lambda^J} - d^{\Lambda^K}}{d^{\Lambda^J} + d^{\Lambda^K}} \quad (5.4)$$

where d^{Λ^J} and d^{Λ^K} are the distances of the point $\mathbf{x}_i$ from the closest correct and incorrect prototypes respectively. The update rules are described in detail in (Schneider et al. 2009a, Bunte 2011). Besides allowing non-linear decision boundaries and therefore learning of more complex classification problems, the localized matrices furthermore enable the investigation of localized or class-wise relevances, marked on each diagonal of Λ^j , identifying features that are important for the classification of each class respectively (Schneider et al. 2009b).

5.3.2 Ensemble methods

With increasing complexity, classifiers get more powerful showing impressive performance in practice. However, at the same time they often show overfitting effects in which the performance on training data is near perfect but it decreases facing new data samples not seen before. This decreased *generalization* performance is often tackled using ensemble methods, which combine several classifiers to assign a class label to a new data instance to overcome the limitations of a single model. In order to see the effect of ensemble learning, we explore an ensemble of LGMLVQ models and compare it to Random Forest (RF). Random forest (Breiman 2001) bags an ensemble of decision trees for classification that vote for the most popular class for a given input. The trees within, are constructed from the bootstrap examples that are sampled independently with replacement and random feature selection at the splits following a common distribution while growing to maximum depth with no pruning. The stochastic strategy is robust to outliers and improves the performance of random forest because of the law of large numbers from combination of rather low performing/weak decision tree classifiers (Breiman 2001). In addition to classification, feature relevance values that are insightful can be extracted from a random forest for comparison with the LVQ method. For a fair comparison with RF

the ensemble of LGMLVQ models is constructed from the same training bootstrap samples used in each decision tree in the random forest. This will result in as many LGMLVQ models as the number of decision trees for each cross validation fold. The results are then aggregated through majority voting (Ranawana and Palade 2006).

5.3.3 Interpretability

For many applications, it is crucial for machine learning models to be interpretable, such that the domain expert is able to examine the significance of the resulting trained model and its suitability for classification tasks. Intrinsic model interpretability can be understood as how understandable the internals of a model and its output are to users (Gilpin et al. 2018). It is furthermore suggestively explained by Backhaus and Seiffert (Backhaus and Seiffert 2014) through a three-fold criteria of the model's 1) feature selection capability, 2) ability to define class representatives, such as prototypes and 3) encoding of classification boundary information. Interpretability for the random forest is achieved through random permutation of a feature's observations (Breiman 2001, Strobl et al. 2007) for the out of bag samples and estimating the corresponding decision tree's accuracy with the permuted features. Here, more discriminative features are easily identified, since they have significant effect on the classification error. The out of bag predictor importance uncovers the individual impact of the features and the information could similarly be used for feature selection and understanding the random forest's classification. On the other hand, the adaptive LVQ methods satisfy all three criteria by: 1) Feature selection by means of the diagonal of the metric tensors, Λ from GMLVQ and the local Λ^j in LGMLVQ, that represents individual feature contribution that could be used as feature pruning criteria. 2) Prototype feature values used to classify novel observations are learned during the model training, which subsequently become typical representatives of their corresponding classes. 3) The decision boundaries for classification can be extracted and visualised, for example by linear projection of the data samples and the resulting class prototypes using Ω from GMLVQ. Nonlinear visualizations using the localized variants LiRaMLVQ and LGMLVQ are also possible and the latter is outlined in the following.

5.3.4 Nonlinear visualization with charting

Visualization can be useful to get a holistic view of the data and identify difficult instances. From the definition of $\Lambda^j = \Omega^{j\top} \Omega^j$ in Eq. (5.3), we see that the distance metric first transforms data points using the following local linear projections:

$$P_j : \mathbf{x} \rightarrow \Omega^{j\top} (\mathbf{x} - \boldsymbol{\omega}^j) .$$

For specific cases $M \in \{2, 3\}$, the projected data points can be visualized, which can be used for discriminant visualization of the data based on the space the classification takes place in. However, since the localized metric provides several projections for each sample, it is challenging to study the outputs directly. In order to tackle this problem, (Bunte et al. 2009) proposed a post-processing step which combines local projections using charting (Brand 2003) to form a global nonlinear embedding of the data:

$$\mathbf{x} \rightarrow \sum_j p_j(\mathbf{x}) B_j(P_j(\mathbf{x})) .$$

Here, $B_j(\cdot) : \mathbb{R}^M \rightarrow \mathbb{R}^M$ is an affine transformation and $p_j(\mathbf{x})$ is the responsibility of local transformation ω^j for the data sample $\mathbf{x}$ with $\sum_j p_j(\mathbf{x}) = 1$. More details about how to compute prototypes' responsibilities and affine transformations can be found in (Bunte et al. 2009)¹. Using this nonlinear embedding we can easily project data to 2 (or 3) dimensions for further investigation of the overlapping regions and difficult samples.

5.4 Experiments and Discussion

This section shows the results and discussion of the experiments conducted with the localized adaptive distance metric LVQ method (LGMLVQ) coupled with presence or absence of resampling as a pre-processing step. The performance and feature importance is compared with Random Forest (RF). The corresponding feature relevances are examined in comparison with conventional astronomical expectations.

5.4.1 Experimental setup and Evaluation Measures

The experiments are set up within a 10-fold cross validation where the data observations are divided up into a 90/10 random but stratified training and test splits with each individual class preserving its sample frequency. For the distance based classifiers, such as LGMLVQ, each training set is normalised via Z-score transformation, e.g. zero mean and unit standard deviation, with the same parameters used to transform the respective test set. Decision trees and RF are not distance based and build instead rules on the features directly and therefore do not require transformative pre-processing in general. However, the RF is very sensitive to class imbalance, which should be handled before training. Therefore, resampling of the training data can be introduced to reduce or eliminate the imbalance of the classes. In this contribution we compare two strategies, namely the Synthetic Minority Oversampling

¹The code is available at https://github.com/kbunte/LVQ_toolbox

Technique (SMOTE) and Borderline-SMOTE, creating new feature vectors using the training samples of each minority class until their amount matches the size of the majority class. The created synthetic minority samples lead to balanced classes to be used for training of the classification models.

The different models have different hyper-parameters. In the experiments we train the RF with 100 decision trees, sampling with replacement of 0.75 fractions of the training set and using the bagging aggregation method (Breiman 2001). The LVQ models provide several hyper-parameters to control the methods complexity, such as the number of prototypes, number of metric tensors and their rank determining the projection dimension saving memory and enabling visualization. Due to the non-linearity of the problem we use the localized most powerful version of the LVQ family, namely LGMLVQ (Schneider et al. 2010). with one prototype per class and regularization of 10^{-5} . In order to choose the lower dimensional projection dimension we train the method using full metric tensors constructed using $\Omega^j \in \mathbb{R}^{M \times D}$ with $M = D$. Subsequently, an Eigenvalue decomposition of the trained $\Lambda^j = U^j \Sigma^j U^{j(-1)}$ with diagonal matrices $\Sigma_{ii}^j = \sigma_i^j$ providing information about the intrinsic dimensionality of the classification problem by counting the Eigenvalues $M = \max_j \sum_{i=1}^D [\sigma_i^j > \epsilon]$ significantly larger than 0. The model is then retrained with the rank M reduced to maximal number obtained from the matrices. The RF is an ensemble of decision trees and therefore we also build an ensemble of the trained LGMLVQ models from the same bootstrap samples used the 100 decision trees for direct comparison and the results are aggregated through majority voting. Resulting models were averaged across cross-validation runs and evaluated with focus on the class of interest.

For evaluation output statistics are generated after prediction with the models, i.e. training and test errors, and their standard deviations. Since this is a multi-class classification problem and the major interest is in Class 2 UCD/GC objects, we extract the evaluation measures for each class and build the macro averaged accuracies to evaluate the performance across the different classes and eliminate bias of majority populated class. We report also binary class measures, such as precision, sensitivity and specificity, for the class of interest versus all others classes combined to represent the algorithms performance in classifying the novel test data. The confusion matrix as illustrated in Table 5.1, can be used to evaluate the classification performance providing detailed information about the accuracy for each class and the nature of misclassifications. From it one can extract the binary measures, namely false positives (FP), true positives (TP), false negatives (FN) and true negatives (TN) as shown below Table 5.1 in Eq. (5.5). Additionally, the false positive rate and true positive rate of the test set are plotted on a Receiver Operating Characteristic (ROC) curve (Fawcett 2006). This curve shows the model's discriminative ability and the

Table 5.1: Three-class confusion matrix and formulae to obtain the False Negatives, True Positives, False Positives and True Negatives (FN_b , TP_b , FP_b and TN_b) to calculate Precision, Specificity and Sensitivity with respect to class b .

Predicted Actual	Class 1	Class 2	Class 3
Class 1	C_{11}	C_{12}	C_{13}
Class 2	C_{21}	C_{22}	C_{23}
Class 3	C_{31}	C_{32}	C_{33}

$$\begin{aligned}
 FN_b &= \sum_{f \neq b}^3 C_{bf} & FP_b &= \sum_{f \neq b}^3 C_{fb} \\
 TN_b &= \sum_{f \neq b}^3 \sum_{q \neq b}^3 C_{fqx} & TP_b &= C_{bb}
 \end{aligned} \tag{5.5}$$

Table 5.2: Macro average performance (standard deviation) for techniques LGMLVQ (T=L) and Random Forest (T=RF) (abbreviated by $\{\emptyset|B|R\}T_M^{\{\emptyset|E\}}$).

	Prec.	Spec.	Sens.	Train accur	Test accur
L_2	0.969 (0.012)	0.986 (0.004)	0.930 (0.021)	0.985 (0.001)	0.984 (0.004)
${}_RL_2$	0.935 (0.020)	0.987 (0.005)	0.963 (0.020)	0.983 (0.001)	0.982 (0.005)
${}_BL_2$	0.889 (0.026)	0.979 (0.007)	0.950 (0.027)	0.971 (0.005)	0.971 (0.006)
${}_RL_2^E$	0.937 (0.019)	0.987 (0.005)	0.962 (0.020)	0.983 (0.001)	0.983 (0.005)
${}_R{}^{RF}E$	0.950 (0.018)	0.989 (0.005)	0.964 (0.018)	0.999 (0.000)	0.986 (0.005)

Area Under the Curve (AUC) summarizes its overall performance for the classification of the class of interest.

5.4.2 Discussion

In this section we summarize the results of the methods on the astronomical classification problem. The hyperparameter settings for the techniques LGMLVQ (T=L) and Random Forest (T=RF) is abbreviated by $\{\emptyset|B|R\}T_M^{\{\emptyset|E\}}$. Here the preprocessing is marked by letters R and B for resampling with SMOTE or Borderline-SMOTE, the subscript M denotes the rank for the local metric tensors and the superscript E denotes the results of an ensemble of 1000 models. The performance of the LGMLVQ models is excellent already with an intrinsic dimensionality of $M = 2$ even without resampling, evident from the sensitivity and precision for the minority class, as shown in Table 5.2. Resampling improves the correct classification by 14.79% for the minority class as demonstrated by an increase in the sensitivity. However, precision reduces indicating that there are more false positives brought about by

Figure 5.2: Panel (a) shows the test confusion matrix for the RL_2 model and panel (b) shows the corresponding ROC curve of minority class 2 vs. all the other classes and the corresponding AUC value of 0.99387.

Figure 5.3: Wrong (a) and correct (b) classification of two minority class observations. Each shows the distances of all samples projected using the local Ω^j of each prototype, highlighting the samples with corresponding label j in colour.

SMOTE resampling, which is an acceptable trade-off. Figure 5.2 panel a) and b) show the test performance of RL_2 with only 3 false negatives for the minority class 2 of UCDs and GCs. Contrary to our expectations Borderline-SMOTE resampling does not perform better. This could be caused by the overlap of the classes which increases the difficulty to define a clear boundary and hence boundary resampling becoming ineffective. In summary the Random Forest and LVQ models show comparable performance. However, especially the latter is less complex and provides further insights into the results of the classification which will be discussed in the following.

As mentioned before the LVQ models are intrinsically interpretable and transparent in many regards. We can for example interpret the certainty of the classi-

Figure 5.4: a) Manifold charting projection of the data by LGMLVQ ($\mathbb{R}L_2$) and b) conventional color-color diagram used in photometric selection by Astronomy.

fication by investigating the distance of each sample to all prototypes. To demonstrate this we project all samples and all prototypes with the local transformations Ω^j and compute the distance to each prototype in the transformed space. Figure 5.3 visualizes these distances highlighting the samples with the same respective class label j in colour and of different classes in grey. We furthermore highlight in black the distances of an observations consistently misclassified in repeated training runs (panel a) and a correctly classified sample (panel b). The wrongly classified minority sample in (a) is within the boundary region where the classes 2 and 3 overlap as indicated by very similar distances to the prototypes of these classes. Panel (b), on the other hand, shows a typical example of a very certain correct classification where the sample is much closer to prototype 2 compared to the others. This transparently informs the user how sure the LGMLVQ classifier is with the assignment of a class label for each observation, which may also indicate samples with potentially controversial current label identification given the data at hand, marking them as candidates for further investigation.

Moreover, the local linear projections $\Omega^j \in \mathbb{R}^{M \times D}$ can be used for nonlinear visualization for $M \in \{2, 3\}$ using manifold charting as outlined in section 5.3.4. Hence we report very good performance for LGMLVQ using the rank $M = 2$, we show the resulting projected data of $\mathbb{R}L_2$ to complement the data analysis in Figure 5.4 panel a. The more distinctive separation provided by the LGMLVQ model, especially for the minority class, explains the efficiency and nuance of the method's classification performance as compared to the traditional astronomical color-color diagram as shown in panel b. This visualization shows the difficult regions and can be used to identify difficult observations, such as the circled class 2 sample in panel a, in the now reduced overlapping areas.

As mentioned before the RF and LGMLVQ models allow to extract the impor-

tance of the measurements for the classification problem. However, the Random Forest method is expensive in terms of memory costs and shows a clear tendency to overfit as seen in the test error being higher than the training error. Panel (d) in Fig-

Figure 5.5: Panel (a-c): class-wise feature relevance profiles of LGMLVQ_RL₂ marking the top 6 in green and their corresponding percentage of contribution to the respective class and (d) the Random Forest (RF) feature relevance profile.

ure 5.5 shows the dominance of the angular size features $FWHM*u$, $FWHM*g$, $FWHM*r$, $FWHM*i$, $FWHM*j$ and $FWHM*ks$ in importance for the Random Forest classifier. In contrast to RF the LGMLVQ classifier extracts the feature relevances for each prototype and hence we can discuss also the relevance of the measurements for the classification of each class of objects in our astronomical data. Figure 5.5(a-c) shows the class-wise normalized feature importance profiles sorted by value of contribution with varying top relevant features further explaining the

Figure 5.6: Pairwise plots of the R_{L_2} top 6 features for the UCDs/GCs class 2 (red markers) with focus on the area covered by that class.

Figure 5.7: Corresponding violin plots with prototype positions relative to the feature value distribution for minority class 2.

non-linearity and motivation for choice of local metric tensors. The top relevant features for classifying the minority class of UCDs and GCs from this experiment

dominantly consist of the angular size parameters, namely $FWHM^*g$, $FWHM^*i$, $FWHM^*u$ and $FWHM^*r$ and their pairwise correlation plots are visualized in Figure 5.6. Similarly 5.7 provides additional information in form of a violin plot for the 6 features most important for the classification of class 2, showing the distributions of the measurements of each class together with the value the class prototype exhibits after training. These angular size features are important for separating the majority of objects in class 1, whose objects are larger, from class 2 and 3 that have small sizes. The astronomical expectation is that the angular size cannot discriminate class 2 and 3 as illustrated by the model. However, in contrary to astronomical expectation, the angular sizes are found to be important to distinguish class 2 and 3 by both the LGMLVQ and RF, as shown in Figure 5.5. The disparity can be attributed to the measurement biases: The minority class 2 objects are faint (and lower in signal-to-noise ratio) and hence angular size measurements happen to be larger than the actual size which could introduce a bias to the data. Therefore, their angular size proxies, i.e. $FWHM^*g$, $FWHM^*i$, $FWHM^*u$ and $FWHM^*r$, could possess more discriminating information than colour indices.

Based on (Munoz et al. 2013), in the combined optical/near-infrared observations, the color indices of $u - i$ and $i - Ks$ are expected to be the most important features. In a simple view, the $u - i$ color is more sensitive to the age of an object while $i - Ks$ represents the metallicity (the amount of metals heavier than Helium) of the object. Other color indices, such as $g - i$, $g - r$, $r - J$ etc. could also partially carry these information but degenerate. In contrary, the observations in the u and Ks are harder to be carried out and often have a lower signal-to-noise ratio compared to the other filters. This makes the expected feature importance of $u - i$ and $i - Ks$ relatively uncertain. Values of $g - r$ and $g - i$ on the other hand are notably accurate.

To examine further the difficult misclassified UCD/GCs example circled in Figure 5.4 a), we visualize in Figure 5.8 the feature values (bottom panel) in comparison with the closest correct prototype (middle) and the corresponding local relevance profile (top panel). The 6 most important measurements for the classification are marked in green as before in each panel. This difficult example is quite different from the correct classifying prototype in the four most important feature dimensions $FWHM^*g$, $FWHM^*j$, $FWHM^*r$ and $g - r$, where the values are much smaller, which explains the misclassification. Such examples require more accurate measurements sizes (using deeper observations) to investigate whether the object indeed belongs to the expected class.

Figure 5.8: Difficult UCD/GCs example (bottom) in contrast with the the closest correct prototype (middle) and corresponding relevance profile (top). The 6 most relevant features for UCD/GCs LGMLVQ classification are marked green.

5.5 Conclusion

In this paper, we explore and compare two interpretable machine learning techniques, namely Localized General Matrix LVQ (LGMLVQ) and Random Forest (RF), for the analysis and classification of foreground stars and background galaxies ver-

sus UCDs and GCs. Due to the class of interest being highly underrepresented compared the former objects we also investigate the influence of Synthetic Minority Oversampling TEchnique (SMOTE) and its borderline extension on the classification performance. Localized distances allowing non-linear decision boundaries within the data improves the classification in LGMLVQ, even in this situation where the classes largely overlap. LGMLVQ is also highly interpretable in the form of prototype class representatives and feature relevances which attach values to the contribution of a feature to classification. Additionally, the experiments uncover classification patterns in terms of feature relevances, which serve as discriminative markers for the classification.

The $u - i$ and $i - Ks$ colors are expected to be the most relevant colors for classification since they carry physical information on ages and metallicities of astronomical objects. However, higher signal-to-noise ratio colors such as $g - i$ and $r - J$ in LGMLVQ and $u - g$ and $g - r$ in Random Forest are found to be more important for the data-driven classification. The importance of other colors compared to $u - i$ and $i - Ks$, that have almost 0 relevance contribution, implies that this disparity may be attributed to astronomically expected features having uncertain measurements, but also the correlation across the features, meaning that they partially contain the same critical information. Furthermore, angular size features $FWHM * g$, $FWHM * i$, $FWHM * u$ are identified by both methods independently as the most important features for the classification. We discuss that this outcome is due to a measurements biases of the faint sources of class 2 (UCDs/GCs).

In this work we showed that existing machine learning techniques can be used to identify UCDs/GCs in big astronomical data. These methods can handle the imbalance in the data and classify sources with a good performance. Subsequent analysis of the transparent techniques allows further insight and can provide valuable information for the astronomical experts to inform about possible biases in the data set. A future deeper data set with more accurate size and color measurements will most likely increase the performance of the automated classification even further.

Chapter 6

Summary

Gravitational forces cause the formation and evolution of a variety of cosmological structures. The detailed investigation and study of these structures is a crucial step towards our understanding of the universe. This thesis provides several solutions for the detection and classification of such structures. In the first part of the thesis, we focus on astronomical simulations, and we propose two algorithms to extract stellar structures, such as globular clusters. In the second part, we consider observational data from surveys such as GAIA, and the goal is to overcome some of the challenges in observational data such as noisy features and imbalanced classes.

In chapter 2, we tackle the problem of extracting low-dimensional structures, called manifolds, buried inside potentially large data-bases. Inspired by the ants' strategy in finding food, we propose the Manifold Alignment Aware Ants (M3A) algorithm. It releases a group of agents on the data space where they discover manifolds and stay close to them, based on a novel formulation of transition probabilities. During this process, they deposit pheromone on the visited points, which serves as positive feedback that encourages subsequent jumps to follow the same direction. Our analysis demonstrates the utility of resulting pheromone values to sort data points according to their closeness to the manifolds. In the experiments, we also show that pheromone can help to build a better probabilistic model in terms of descriptivity and sparsity. Moreover, since the underlying formulation uses local PCA to take into account the geometrical shape of manifolds, it decodes a form of geodesic similarity, and we demonstrate that it can improve the performance of t -distributed stochastic neighbor embedding and spectral clustering.

In chapter 3, we focus on the discovered manifolds. Since in cosmic web simulations millions of particles are distributed around very noisy structures, cleaning manifolds can substantially facilitate the subsequent analysis. However, due to significant deviations in densities and dimensionalities of manifolds, it is necessary to have a method that overcomes these challenges. A biologically inspired algorithm, called Evolutionary Manifold Alignment Aware Agents (EM3A), is proposed to denoise the existing manifolds. Moreover, principles from Evolutionary Game Theory are used to tune the algorithm with appropriate values of hyper-parameter automatically. The EM3A algorithm is demonstrated on a large scale cosmic web

simulation, where it successfully not only denoises one-dimensional filaments, but simultaneously preserves three-dimensional clusters, so called nodes, unchanged.

In part II of this thesis, we focus on astronomical observations where datasets are collected by ground-based or space telescopes. In chapter 4, we consider the detection of astronomical structures in our neighborhood, i.e. in the halo of the Milky Way. We use the GAIA DR1 survey, which contains 2D positional data and 1D G-band magnitude for point sources. In the absence of phase-space and color information, detection of stellar structures is difficult. In order to overcome this difficulty, we use 3D kernel density estimation (KDE) to approximate the distribution of stars in spatial and magnitude space. We compare two conceptually different techniques: a) nearest neighbor retrieval, b) anomaly detection. While the latter strategy does not use any examples of the known structures, the former method take advantage of the known Globular Clusters to extract new candidates. By tuning appropriate values for their hyper-parameters, both methods can retrieve the known structure consistently, as well as several candidates for further investigation.

Finally, chapter 5 is about detection of dense structures in the Fornax galaxy cluster, with the goal to separate Ultra-compact dwarfs (UCDs) and Globular Clusters (GCs) from foreground stars and background galaxies using only multi-wavelength imaging data. Here, we have a noisy data set with highly imbalanced classes. In order to tackle the imbalanced data problem, Synthetic Minority Over-sampling (SMOTE) is applied to increase the number of samples in the minority class, containing UCDs and GCs. To suppress the effect of noise, we apply Localized-GMLVQ, which uses an adaptive metric to find the most relevant features to separate each of the classes against each other. The experiment shows its success in separating UCDs and GCs from the others. Moreover, Localized-GMLVQ provides class-wise representative samples which presents further insight into the classes.

Chapter 7

Future works

Modern telescopes and computers allow collecting enormous amount of astronomical data with ever more details. In this thesis, we touch on several challenges we may encounter, from the lack of exemplars of interesting objects and imbalanced classes in observations to the existence of complex structures with different densities and dimensionalities in simulations. The present work proposes two new algorithms and extends the applications of several machine learning techniques to deal with the mentioned challenges. Our results suggest the following directions for future works.

- In chapter 2, we propose the algorithm M3A to extract astronomical structures buried on large databases. For demonstration, it is used on a small part of the sky in the GAIA survey to recover Globular Clusters. Future experiments contain the application of the algorithm on other parts of the sky for extraction of different types of structures, from circular structures such as Globular Clusters to 1D structures such as streams. In this thesis, we only consider positional information. In the future, we would like to use more features, such as metallicity and velocities, which are available in simulations, and become available by deep imaging techniques and improved telescopes in Observations.
- Chapter 3 explains the algorithm EM3A⁽⁺⁾ for denoising low-dimensional manifolds. In this chapter, we represent its applications on two astronomical simulations, from a cosmic-web simulation to a simulated jelly-fish galaxy. These experiments show its superiority over previous techniques. In future works, it will be applied to more astronomical simulations. For instance, with investigation of the structures in the evolution of the simulation, it can be used as a preprocessing step on different snapshots in order to have a better initialization step for creating probabilistic models such as Gaussian Mixture Model, Fast Parzen Window (FPW), and Generative Topographic Map (GTM).
- As suggested before, EM3A⁺ may have many other potential applications. Moreover, this algorithm already has been confirmed to be fruitful in combination with other techniques like FPW and GTM. Thus, we are going to also

present it as a part of a package with some other compatible techniques for a full analysis pipeline. This package, containing python code and documentation, will be released for the public to be used with a wider group of astronomers or scientists from other fields to explore more applications.

- In astronomy, it is important to have transparent models that enable us to understand the decision made by the models, since it provides a deeper insight into the data. In chapter 5, we use two Explainable AI techniques not only to classify astronomical objects but also to learn about their decision processes. As demonstrated Explainable AI concepts and methods help to see how much data supports our expectations, which is demanding for astronomical applications since different datasets may have different levels of quality.

Bibliography

- Angora, G., Brescia, M., Cavuoti, S., Paolillo, M., Longo, G., Cantiello, M., Capaccioli, M., D'Abrusco, R., D'Ago, G., Hilker, M., Iodice, E., Mieske, S., Napolitano, N., Peletier, R., Pota, V., Puzia, T., Riccio, G. and Spavone, M.: 2019, Astrominformatics-based search for globular clusters in the Fornax Deep Survey, *MNRAS* **490**(3), 4080–4106.
- Arya, S., Mount, D., Netanyahu, N., Silverman, R. and Wu, A.: 1998, An optimal algorithm for approximate nearest neighbor searching fixed dimensions, *Journal of the ACM (JACM)* **45**(6), 891–923.
- Backhaus, A. and Seiffert, U.: 2014, Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size, *Neurocomputing* **131**, 15–22.
- Ball, N. M., Loveday, J., Fukugita, M., Nakamura, O., Okamura, S., Brinkmann, J. and Brunner, R. J.: 2004, Galaxy types in the sloan digital sky survey using supervised artificial neural networks, *Monthly Notices of the Royal Astronomical Society* **348**(3), 1038–1046.
- Barchi, P., de Carvalho, R., Rosa, R., Sautter, R., Soares-Santos, M., Marques, B., Clua, E., Gonçalves, T., de Sá-Freitas, C. and Moura, T.: 2020, Machine and deep learning applied to galaxy morphology-a comparative study, *Astronomy and Computing* **30**, 100334.
- Barnes, J. E.: 1989, Evolution of compact groups and the formation of elliptical galaxies, *Nature* **338**(6211), 123–126.
- Beasley, M. A.: 2020, *Globular Cluster Systems and Galaxy Formation*, pp. 245–277.
- Behroozi, P., Wechsler, R. and Wu, H.-Y.: 2012, The rockstar phase-space temporal halo finder and the velocity offsets of cluster cores, *The Astrophysical Journ.* **762**(2), 109.

- Beis, J. and Lowe, D.: 1997, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, IEEE, pp. 1000–1006.
- Belkin, M. and Niyogi, P.: 2003, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural computation* **15**(6), 1373–1396.
- Bentley, J. L.: 1975, Multidimensional binary search trees used for associative searching, *Communications of the ACM* **18**(9), 509–517.
- Berry, T. and Sauer, T.: 2016, Local kernels and the geometric structure of data, *Applied and Computational Harmonic Analysis* **40**(3), 439–469.
- Bishop, C. M.: 2006a, *Pattern recognition and machine learning*, springer.
- Bishop, C. M.: 2006b, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg.
- Blum, C., Roli, A. and Dorigo, M.: 2001, Hc-aco: The hyper-cube framework for ant colony optimization, *Proceedings of MIC*, Vol. 2, pp. 399–403.
- Brand, M.: 2003, Charting a manifold, *Advances in neural information processing systems*, pp. 985–992.
- Breiman, L.: 2001, Random forests, *Machine learning* **45**(1), 5–32.
- Brown, A., Vallenari, A., Prusti, T., De Bruijne, J., Mignard, F., Drimmel, R., Babusi-
aux, C., Bailer-Jones, C., Bastian, U. et al.: 2016, Gaia data release 1-summary of
the astrometric, photometric, and survey properties, *Astronomy & Astrophysics*
595, A2.
- Bunte, K.: 2011, *Adaptive dissimilarity measures, dimension reduction and visualization*,
PhD thesis.
- Bunte, K., Hammer, B., Schneider, P. and Biehl, M.: 2009, Nonlinear discriminative
data visualization., *ESANN*, pp. 65–70.
- Bunte, K., Schneider, P., Hammer, B., Schleif, F.-M., Villmann, T. and Biehl, M.: 2012,
Limited rank matrix learning, discriminative dimension reduction and visual-
ization, *Neural Networks* **26**, 159 – 173.
URL: <http://www.sciencedirect.com/science/article/pii/S0893608011002632>
- Cantiello, M., Grado, A., Rejkuba, M., Arnaboldi, M., Capaccioli, M., Greggio, L.,
Iodice, E. and Limatola, L.: 2018, A VST and VISTA study of globular clusters
in NGC 253, *A&A* **611**, A21.

- Cantiello, M., Venhola, A., Grado, A., Paolillo, M., D'Abrusco, R., Raimondo, G., Quintini, M., Hilker, M., Mieske, S., Tortora, C., Spavone, M., Capaccioli, M., Iodice, E., Peletier, R., Falcon Barroso, J., Limatola, L., Napolitano, N., Schipani, P., van de Ven, G., Gentile, F. and Covone, G.: 2020, The Fornax Deep Survey with VST. X. The catalog of sources in the FDS area, with an example study for globular clusters and background galaxies, *arXiv e-prints* p. arXiv:2005.12085.
- Carrasco, D., Barrientos, L., Pichara, K., Anguita, T., Murphy, D. N., Gilbank, D. G., Gladders, M. D., Yee, H. K., Hsieh, B. C. and López, S.: 2015, Photometric classification of quasars from rcs-2 using random forest, *Astronomy & Astrophysics* **584**, A44.
- Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P.: 2002, Smote: Synthetic minority over-sampling technique, *J. Artif. Int. Res.* **16**(1), 321–357.
- Chu, C.-Y., Henderson, D. J. and Parmeter, C. F.: 2017, On discrete epanechnikov kernel functions, *Computational Statistics & Data Analysis* **116**, 79–105.
- Chu, S.-C., Roddick, J. F., Su, C.-J. and Pan, J.-S.: 2004, Constrained ant colony optimization for data clustering, *Pacific Rim International Conference on Artificial Intelligence*, Springer, pp. 534–543.
- Cinlar, E.: 2013, *Introduction to stochastic processes*, Courier Corporation.
- Coifman, R. R. and Lafon, S.: 2006, Diffusion maps, *Applied and computational harmonic analysis* **21**(1), 5–30.
- Collaboration, G., Brown, A., Vallenari, A., Prusti, T., de Bruijne, J., Babusiaux, C., Bailer-Jones, C., Biermann, M., Evans, D., Eyer, L. et al.: 2018, Gaia data release 2. summary of the contents and survey properties, *Astronomy and Astrophysics* **616**.
- Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chrétien, L.: 1991, The dynamics of collective sorting robot-like ants and ant-like robots, *Proc. of the first international conference on simulation of adaptive behavior on From animals to animats*, pp. 356–363.
- Dixit, P. D.: 2019, Introducing user-prescribed constraints in markov chains for non-linear dimensionality reduction, *Neural computation* **31**(5), 980–997.
- Donoho, D. L. and Grimes, C.: 2003, Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, *Proceedings of the National Academy of Sciences* **100**(10), 5591–5596.

- Dorigo, M.: 1992, Optimization, learning and natural algorithms, *Ph. D. Thesis, Politecnico di Milano*.
- Dorigo, M., Maniezzo, V. and Colorni, A.: 1991, Positive feedback as a search strategy.
- Dorigo, M., Maniezzo, V. and Colorni, A.: 1996, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **26**(1), 29–41.
- Fawcett, T.: 2006, An introduction to roc analysis, *Pattern recognition letters* **27**(8), 861–874.
- Fuentes, S. S., De Ridder, J. and Debosscher, J.: 2017, Stellar halo hierarchical density structure identification using (f) optics, *Astronomy & Astrophysics* **599**, A143.
- Gao, D., Zhang, Y.-X. and Zhao, Y.-H.: 2009, Random forest algorithm for classification of multiwavelength data, *Research in Astronomy and Astrophysics* **9**(2), 220.
- Gecer, B., Azzopardi, G. and Petkov, N.: 2017, Color-blob-based cosfire filters for object recognition, *Image and Vision Computing* **57**, 165–174.
- Gil, C., Baños, R., Ortega, J., Márquez, A. L., Fernández, A. and Montoya, M.: 2011, Ant colony optimization for water distribution network design: a comparative study, *International Work-Conference on Artificial Neural Networks*, Springer, pp. 300–307.
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. and Kagal, L.: 2018, Explaining explanations: An approach to evaluating interpretability of machine learning, *CoRR*.
URL: <http://arxiv.org/abs/1806.00069>
- Gisbrecht, A. and Schleif, F.: 2015, Metric and non-metric proximity transformations at linear costs, *Neurocomputing* **167**, 643–657.
- Goldberger, J. and Roweis, S. T.: 2005, Hierarchical clustering of a mixture model, *Advances in Neural Information Processing Systems*, Citeseer, pp. 505–512.
- Golub, G. H. and Van Loan, C. F.: 2012, *Matrix computations*, Vol. 3, JHU press.
- Gong, D., Sha, F. and Medioni, G.: 2010, Locally linear denoising on image manifolds, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, pp. 265–272.
- Gorban, A. N. and Zinovyev, A.: 2010, Principal manifolds and graphs in practice: from molecular biology to dynamical systems, *International journal of neural systems* **20**(03), 219–232.

- Hammer, B. and Villmann, T.: 2002, Generalized relevance learning vector quantization, *Neural Networks* **15**(8-9), 1059–1068.
- Han, H., Wang, W.-Y. and Mao, B.-H.: 2005, Borderline-smote: A new over-sampling method in imbalanced data sets learning, in D.-S. Huang, X.-P. Zhang and G.-B. Huang (eds), *Advances in Intelligent Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 878–887.
- Harris, W.: 1996, A catalog of parameters for globular clusters in the milky way, *The Astronomical Journal* **112**, 1487.
- Hein, M. and Maier, M.: 2006, Manifold denoising, *NIPS*, Vol. 19, pp. 561–568.
- Hofbauer, J., Sigmund, K. et al.: 1998, *Evolutionary games and population dynamics*, Cambridge university press.
- Janz, J.: 2012, *The puzzling nature of dwarf-sized gas poor disk galaxies*, PhD thesis.
- Jones, E. and Singal, J.: 2017, Analysis of a custom support vector machine for photometric redshift estimation and the inclusion of galaxy shape information, *Astronomy & Astrophysics* **600**, A113.
URL: <http://dx.doi.org/10.1051/0004-6361/201629558>
- Jordán, A., Peng, E. W., Blakeslee, J. P., Côté, P., Eyheramendy, S., Ferrarese, L., Mei, S., Tonry, J. L. and West, M. J.: 2009, The ACS Virgo Cluster Survey XVI. Selection Procedure and Catalogs of Globular Cluster Candidates, *The Astrophysical Journal Supplement Series* **180**(1), 54–66.
- Kaslovsky, D. N. and Meyer, F. G.: 2014, Non-asymptotic analysis of tangent space perturbation, *Information and Inference: a Journal of the IMA* **3**(2), 134–187.
- Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, K. S. and Igel, C.: 2017, Big universe, big data: machine learning and image analysis for astronomy, *IEEE Intelligent Systems* **32**(2), 16–22.
- Kulkarni, V. G.: 1999, *Modeling, Analysis, Design, and Control of Stochastic Systems*, Springer.
- Lerman, G., McCoy, M. B., Tropp, J. A. and Zhang, T.: 2015, Robust computation of linear models by convex relaxation, *Foundations of Computational Mathematics* **15**(2), 363–410.
- Leskovec, J., Rajaraman, A. and Ullman, J.: 2014, *Mining of massive datasets*, Ch. 3, Cambridge university press.

- Li, L., Zhang, Y. and Zhao, Y.: 2008, k-nearest neighbors for automated classification of celestial objects, *Science in China Series G: Physics, Mechanics and Astronomy* **51**(7), 916–922.
- Little, A. V., Maggioni, M. and Murphy, J. M.: 2020, Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms, *Journal of machine learning research* **21**.
- Little, A. V., Maggioni, M. and Rosasco, L.: 2011, Multiscale geometric methods for estimating intrinsic dimension, *Proc. SampTA* **4**(2).
- Little, A. V., Maggioni, M. and Rosasco, L.: 2017, Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature, *Applied and Computational Harmonic Analysis* **43**(3), 504–567.
- Lumer, E. D. and Faieta, B.: 1994, Diversity and adaptation in populations of clustering ants, *Proc. of the 3rd international conference on Simulation of adaptive behavior: from animals to animats 3*, MIT Press, pp. 501–508.
- Maniezzo, V.: 1999, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, *INFORMS journal on computing* **11**(4), 358–369.
- McMahon, R. G., Banerji, M., Gonzalez, E., Koposov, S. E., Bejar, V. J., Lodieu, N., Rebolo, R. and VHS Collaboration: 2013, First Scientific Results from the VISTA Hemisphere Survey (VHS), *The Messenger* **154**, 35–37.
- Mohammadi, M., Peletier, R. F., Schleif, F. M., Petkov, N. and Bunte, K.: 2018, Globular cluster detection in the gaia survey, *ESANN*.
- Mohammadi, M., Petkov, N., Bunte, K., Peletier, R. F. and Schleif, F.-M.: 2019, Globular cluster detection in the gaia survey, *Neurocomputing* **342**, 164–171.
- Mordohai, P. and Medioni, G. G.: 2005, Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting., *IJCAI*, pp. 798–803.
- Muñoz, R. P., Puzia, T. H., Lançon, A., Peng, E. W., Côté, P., Ferrarese, L., Blakeslee, J. P., Mei, S., Cuillandre, J.-C., Hudelot, P., Courteau, S., Duc, P.-A., Balogh, M. L., Boselli, A., Bournaud, F., Carlberg, R. G., Chapman, S. C., Durrell, P., Eigenthaler, P., Emsellem, E., Gavazzi, G., Gwyn, S., Huertas-Company, M., Ilbert, O., Jordán, A., Läsker, R., Licitra, R., Liu, C., MacArthur, L., McConnachie, A., McCracken, H. J., Mellier, Y., Peng, C. Y., Raichoor, A., Taylor, M. A., Tonry, J. L., Tully, R. B. and Zhang, H.: 2014, The Next Generation Virgo

- Cluster Survey-Infrared (NGVS-IR). I. A New Near-Ultraviolet, Optical, and Near-Infrared Globular Cluster Selection Tool, *The Astrophysical Journal Supplement Series* **210**(1), 4.
- Munoz, R. P., Puzia, T. H., Lançon, A., Peng, E. W., Cote, P., Ferrarese, L., Blakeslee, J. P., Mei, S., Cuillandre, J.-C., Hudelot, P. et al.: 2013, The next generation virgo cluster survey-infrared (ngvs-ir). i. a new near-ultraviolet, optical, and near-infrared globular cluster selection tool, *The Astrophysical Journal Supplement Series* **210**(1), 4.
- Pekalska, E. and Duin, R.: 2005, *The dissimilarity representation for pattern recognition: foundations and applications*, Ch. 2, World Scientific.
- Pota, V., Napolitano, N. R., Hilker, M., Spavone, M., Schulz, C., Cantiello, M., Tortora, C., Iodice, E., Paolillo, M., D'Abrusco, R., Capaccioli, M., Puzia, T., Peletier, R. F., Romanowsky, A. J., van de Ven, G., Spiniello, C., Norris, M., Lisker, T., Munoz, R., Schipani, P., Eigenthaler, P., Taylor, M. A., Sánchez-Janssen, R. and Ordenes-Briceño, Y.: 2018, The Fornax Cluster VLT Spectroscopic Survey - I. VIMOS spectroscopy of compact stellar systems in the Fornax core region, *MNRAS* **481**(2), 1744–1756.
- Prole, D. J., Hilker, M., van der Burg, R. F. J., Cantiello, M., Venhola, A., Iodice, E., van de Ven, G., Wittmann, C., Peletier, R. F., Mieske, S., Capaccioli, M., Napolitano, N. R., Paolillo, M., Spavone, M. and Valentijn, E.: 2019, Halo mass estimates from the globular cluster populations of 175 low surface brightness galaxies in the Fornax cluster, *MNRAS* **484**(4), 4865–4880.
- Ranawana, R. and Palade, V.: 2006, Multi-classifier systems: Review and a roadmap for developers, *Int. J. Hybrid Intell. Syst.* **3**(1), 35–61.
- Riello, M., Cassisi, S., Piotto, G., Recio-Blanco, A., De Angeli, F., Salaris, M., Pietrinferni, A., Bono, G. and Zoccali, M.: 2003, The red giant branch luminosity function bump, *Astronomy & Astrophysics* **410**(2), 553–563.
- Rizzoli, A. E., Oliverio, F., Montemanni, R. and Gambardella, L. M.: 2004, Ant colony optimisation for vehicle routing problems: from theory to applications, *Galleria Rassegna Bimestrale Di Cultura* **9**(1), 1–50.
- Rockosi, C., Odenkirchen, M., Grebel, E., Dehnen, W., Cudworth, K., Gunn, J., York, D., Brinkmann, J., Hennessy, G. and Ivezić, Z.: 2002, A matched-filter analysis of the tidal tails of the globular cluster palomar 5, *The Astronomical Journ.* **124**(1), 349.

- Roweis, S. T. and Saul, L. K.: 2000, Nonlinear dimensionality reduction by locally linear embedding, *science* **290**(5500), 2323–2326.
- Runkler, T. A.: 2005, Ant colony optimization of clustering models, *International Journal of Intelligent Systems* **20**(12), 1233–1251.
- Sakoe, H. and Chiba, S.: 1978, Dynamic programming algorithm optimization for spoken word recognition, *IEEE transactions on acoustics, speech, and signal processing* **26**(1), 43–49.
- Sato, A. and Yamada, K.: 1995, Generalized learning vector quantization, *Proceedings of the 8th International Conference on Neural Information Processing Systems, NIPS'95*, MIT Press, Cambridge, MA, USA, p. 423–429.
- Schleif, F.-M. and Tino, P.: 2015, Indefinite proximity learning: A review, *Neural Computation* **27**(10), 2039–2096.
- Schneider, P., Biehl, M. and Hammer, B.: 2009a, Adaptive relevance matrices in learning vector quantization, *Neural Comput.* **21**(12), 3532–3561.
URL: <https://doi.org/10.1162/neco.2009.11-08-908>
- Schneider, P., Biehl, M. and Hammer, B.: 2009b, Distance learning in discriminative vector quantization, *Neural Computation* **21**(10), 2942–2969.
URL: <https://doi.org/10.1162/neco.2009.10-08-892>
- Schneider, P., Bunte, K., Stiekema, H., Hammer, B., Villmann, T. and Biehl, M.: 2010, Regularization in matrix relevance learning, *IEEE Transactions on Neural Networks* **21**(5), 831–840.
- Shawe-Taylor, J. and Cristianini, N.: 2004, *Kernel Methods for Pattern Analysis and Discovery*, Cambridge University Press.
- Shelokar, P., Jayaraman, V. K. and Kulkarni, B. D.: 2004, An ant colony approach for clustering, *Analytica Chimica Acta* **509**(2), 187–195.
- Simonoff, J. S.: 2012, *Smoothing methods in statistics*, Springer Science & Business Media.
- Skiena, S. S.: 1998, *The Algorithm Design Manual*, Springer-Verlag, Berlin, Heidelberg.
- Strobl, C., Boulesteix, A.-L., Zeileis, A. and Hothorn, T.: 2007, Bias in random forest variable importance measures: Illustrations, sources and a solution, *BMC bioinformatics* **8**(1), 25.
- Stützle, T. and Hoos, H. H.: 2000, Max–min ant system, *Future generation computer systems* **16**(8), 889–914.

- Taghribi, A., Bunte, K., Smith, R., Shin, J., Mastropietro, M., Peletier, R. F. and Tino, P.: 2020, Laat: Locally aligned ant technique for detecting manifolds of varying density, *arXiv preprint arXiv:2009.08326*.
- Taubin, G.: 1995, A signal processing approach to fair surface design, *Proc. of the 22nd conf. on Computer graphics and interactive techniques*, pp. 351–358.
- Tax, D. M. and Duin, R. P.: 1999, Support vector domain description, *Pattern recognition letters* **20**(11-13), 1191–1199.
- Theraulaz, G., Gautrais, J., Camazine, S. and Deneubourg, J.-L.: 2003, The formation of spatial patterns in social insects: from simple behaviours to complex structures, *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **361**(1807), 1263–1282.
- Tian, H.: 2017, *Hide and Seek in the Halo of the MW*, PhD thesis, Univ. of Groningen.
- Tsai, C.-F., Tsai, C.-W., Wu, H.-C. and Yang, T.: 2004, Acodf: a novel data clustering approach for data mining in large databases, *Journal of Systems and Software* **73**(1), 133–145.
- Tsang, I. W., Kwok, J. T. and Zurada, J. M.: 2006, Generalized core vector machines, *IEEE Trans. Neural Networks* **17**(5), 1126–1140.
- Tu, L. W.: 2011, *An introduction to manifolds*, Springer.
- Van der Maaten, L. and Hinton, G.: 2008, Visualizing data using t-sne, *Journal of machine learning research* **9**(11).
- Vincent, P. and Bengio, Y.: 2003, Manifold parzen windows, *Advances in neural information processing systems* pp. 849–856.
- Voggel, K. T., Seth, A. C., Sand, D. J., Hughes, A., Strader, J., Crnojevic, D. and Caldwell, N.: 2020, A Gaia-based Catalog of Candidate Stripped Nuclei and Luminous Globular Clusters in the Halo of Centaurus A, *The Astrophysical Journal Supplement Series* **899**(2), 140.
- Von Luxburg, U.: 2007, A tutorial on spectral clustering, *Statistics and computing* **17**(4), 395–416.
- Wang, W. and Carreira-Perpinán, M. A.: 2010, Manifold blurring mean shift algorithms for manifold denoising, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1759–1766.
- Wang, X., Tino, P. and Fardal, M. A.: 2008, Multiple manifolds learning framework based on hierarchical mixture density model, *ECML PKDD*, pp. 566–581.

- Wang, X., Tino, P., Fardal, M. A., Raychaudhury, S. and Babul, A.: 2009, Fast parzen window density estimator, *2009 International Joint Conference on Neural Networks*, IEEE, pp. 3267–3274.
- Xiang, Y. and Chen, Y.-C.: 2016, Statistical inference using mean shift denoising, *arXiv preprint arXiv:1610.03927* .
- Xiao, H., Cao, H., Fan, J., Costantin, D., Luo, G. and Pei, Z.: 2020, Efficient fermi source identification with machine learning methods, *Astronomy and Computing* p. 100387.
- Zhang, K. and Kwok, J. T.: 2010, Simplifying mixture models through function approximation, *IEEE Transactions on Neural Networks* **21**(4), 644–658.
- Zhang, Z. and Zha, H.: 2003, Nonlinear dimension reduction via local tangent space alignment, *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, pp. 477–481.

Samenvatting

Gravitatiekrachten veroorzaken de formatie en evolutie van een variëteit aan kosmologische structuren. Gedetailleerd onderzoek en bestudering van deze structuren is een cruciale stap in ons begrip van het universum. Dit proefschrift geeft verschillende oplossingen voor de detectie en classificatie van zulke structuren. In het eerste gedeelte van het proefschrift richten we ons op sterrenkundige simulaties, en we stellen twee algoritmen voor om stellaire structuren te extraheren, zoals bolvormige sterrenhopen. In het tweede gedeelte behandelen we waarnemingsgegevens van onderzoeken zoals GAIA, en het doel is om sommige uitdagingen van waarnemingsgegevens, zoals ruis en ongebalanceerde klassen, aan te pakken.

In hoofdstuk 2, pakken we het probleem op van het extraheren van laag-dimensionale structuren, manifolds/variëteiten genoemd, die verscholen liggen in potentieel grote databanken. Geïnspireerd door de strategie van mieren voor het vinden van voedsel, stellen we het *Manifold Alignment Aware Ants* (M3A) algoritme voor. Een groep van agenten wordt uitgezet in de dataruimte waar ze manifolds ontdekken en daar dichtbij blijven, aan de hand van een nieuwe formulatie van overgangswaarschijnlijkheden. Gedurende dit proces geven ze feromonen af bij de bezochte punten, die dienen als een positieve terugkoppeling, die daaropvolgende sprongen aanmoedigt om dezelfde richting te volgen. Onze analyse laat de bruikbaarheid zien van de resulterende feromoonwaarden voor het sorteren van de datapunten aan de hand van de nabijheid tot de manifold. In de experimenten laten we ook zien dat feromoon kan helpen om een beter probabilistisch model te bouwen wat betreft de beschrijving en de sparsity. Omdat de onderliggende formulering lokale PCA gebruikt om de geometrische vorm van de manifold erbij te betrekken, decodeert het bovendien een vorm van geodetische similariteit, en we tonen aan dat het de prestaties van t-distributed stochastic neighbor embedding en spectral clustering kan verbeteren.

In hoofdstuk 3, richten we ons op de ontdekte manifolds. Aangezien er in kosmische web simulaties miljoenen deeltjes verspreid zijn rondom structuren met veel ruis, kan het opschonen van manifolds de daaropvolgende analyse substantieel vergemakkelijken. Echter, wegens significante afwijkingen in de dichtheden en di-

mensies van de manifolds, is het nodig om een methode te hebben die deze uitdagingen kan verhelpen. Een door de biologie geïnspireerd algoritme, genaamd *Evolutionary Manifold Alignment Aware Agents* (EM3A), wordt voorgesteld om de bestaande manifolds van ruis te ontdoen. Daarnaast worden principes uit de Evolutionaire Speltheorie gebruikt om automatisch de hyper-parameters van het algoritme op geschikte waarden te zetten. Het EM3A algoritme wordt getoond op een grootschalige kosmische web simulatie, waar het niet alleen succesvol één-dimensionale filamenten van ruis ontdoet, maar tegelijkertijd ook drie-dimensionale clusters, knopen genaamd, onveranderd laat.

In gedeelte II van het proefschrift richten we ons op astronomische waarnemingen waar datasets verzameld worden door telescopen op de grond of in de ruimte. In hoofdstuk 4 nemen we de detectie van astronomische structuren bij ons in de buurt in beschouwing, in de halo van het Melkwegstelsel. We gebruiken het GAIA DR1 onderzoek dat 2D positionele data bevat en 1D G-band magnitude voor puntbronnen. Bij het ontbreken van faseruimte- en kleurinformatie is de detectie van stellaire structuren moeilijk. Om deze moeilijkheid te verhelpen gebruiken we 3D *kernel density estimation* (KDE) om de verdeling te benaderen van sterren in de ruimte en in magnitude. We vergelijken twee conceptueel verschillende technieken: a) nearest neighbor retrieval, b) anomaly detection. Hoewel de laatstgenoemde strategie niet gebruik maakt van voorbeelden van bekende structuren, doet de eerstgenoemde methode zijn voordeel met de bekende bolvormige sterrenhopen om nieuwe kandidaten te extraheren. Door het afstellen op geschikte waarden van de hyper-parameters, kunnen beide methoden de bekende structuur consistent terugvinden, alsook verscheidene kandidaten voor nader onderzoek.

Tenslotte gaat hoofdstuk 5 over de detectie van dichte structuren in de Fornaxcluster, met als doel om de Ultra-compact dwarfs (UCDs) en Globular Clusters (GCs) te scheiden van sterren op de voorgrond en sterrenstelsels op de achtergrond gebruikmakend van alleen multi-wavelength imaging data. Hier hebben we een dataset met ruis en een hoge disbalans in de klassen. Om het probleem van disbalans in de data aan te pakken wordt *Synthetic Minority Over-sampling* (SMOTE) toegepast om het aantal samples van de klassen die in de minderheid zijn te verhogen, die UCDs en GCs bevatten. Om het effect van ruis te onderdrukken passen we *Localized-GMLVQ* toe, welke een adaptieve metriek gebruikt om de meest relevante variabelen te vinden die de verschillende klassen van elkaar scheidt. Het experiment toont zijn succes in het onderscheid maken tussen UCDs en GCs. Daarnaast geeft Localized-GMLVQ voor elke klasse representatieve samples die meer inzicht verschaffen in die klasse.

Research Activities

Journal Papers

- **M. Mohammadi**, J. Mutatiina, T. Saifollahi, K. Bunte, *Detection of extragalactic Ultra-Compact Dwarfs and Globular Clusters using Explainable AI techniques*, submitted.
- **M. Mohammadi**, P.Tino, K. Bunte, *Manifold Alignment Aware Ants: a Markovian process for manifold extraction*, Neural Computation, accepted.
- **M. Mohammadi**, N. Petkov, K. Bunte, R.F. Peletier, and F-M. Schleif, *Globular cluster detection in the gaia survey*, Neurocomputing, 342 (2019): 164-171.

Conference Proceedings

- **M. Mohammadi**, K. Bunte, *Multi-agent based manifold denoising*, 21st International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2020).
- **M. Mohammadi**, R.F. Peletier, F.M. Schleif, N. Petkov, and K. Bunte, *Globular cluster detection in the gaia survey*, 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2018).
- **M. Mohammadi**, N. Petkov, R.F. Peletier, P. Bibiloni, and K. Bunte, *Detection of Globular Clusters in the halo of Milky Way*, In Frontiers in Artificial Intelligence and Applications, 2018.

Attended Workshops and Training Schools

- SUNDIAL Data mining astronomical archives and the Virtual Observatory, 2019, Groningen, The Netherlands.
- 9th to 11th Mittweidaer Workshop for Computational Intelligence (MiWoCi), 2017-2019, Mittweida, Germany.
- International Workshop on Intelligent Systems and Computational Intelligence (WISCI) 2018, Groningen, the Netherlands.
- SUNDIAL Training School on Professional Software Writing and Project Management, 2018, Napels, Italy.
- SUNDIAL Astronomical Observing Training School, 2018, La Palma in the Canary Islands.
- Autumn school on model order reduction, 2017, Groningen, The Netherlands.