Principal Variety Analysis

Reza Iraji
Department of Computer Science
Colorado State University
rezairaji@gmail.com

Hamidreza Chitsaz
Department of Computer Science
Colorado State University
chitsaz@chitsazlab.org

Abstract: We introduce a novel computational framework, Principal Variety Analysis (PVA), for primarily nonlinear data modeling. PVA accommodates algebraic sets as the target subspace through which limitations of other existing approaches is dealt with. PVA power is demonstrated in this paper for learning kinematics of objects, as an important application. PVA takes recorded coordinates of some pre-specified features on the objects as input and outputs a lowest dimensional variety on which the feature coordinates jointly lie. Unlike existing object modeling methods, which require entire trajectories of objects, PVA requires much less information and provides more flexible and generalizable models, namely an analytical algebraic kinematic model of the objects, even in unstructured, uncertain environments. Moreover, it is not restricted to predetermined model templates and is capable of extracting much more general types of models. Besides finding the kinematic model of objects, PVA can be a powerful tool to estimate their corresponding degrees of freedom. PVA computational success depends on exploiting sparsity, in particular algebraic dimension minimization through replacement of intractable ℓ_0 norm (rank) with tractable ℓ_1 norm (nuclear norm). Complete characterization of the assumptions under which ℓ_0 and ℓ_1 norm minimizations yield virtually the same outcome is introduced as an important open problem in this paper.

Keywords: Nonlinear Dimensionality Reduction, Nonlinear Data Modeling, Analytical Algebraic Modeling, Norm Minimization, Sparsity, Kinematics Learning

1 Introduction

With the advent of many advances in technology, such as low-cost high throughput sensors, human genome sequencing, physics and astronomy, social media, and etc., available data from real world phenomena has reached an unprecedented level. Captured high quality data are in fact growing at an astronomical rate. That necessitates advancement of algorithms and methods to extract models, patterns, and knowledge from data. For instance, methods for mostly linear dimensionality reduction [1], capturing topology of the space from which data points are sampled [2], and studying various aspects of data geometry including manifold learning have been proposed [3]. However in spite of that progress, there are still many important open problems and unexplored areas in data modeling, particularly for nonlinear spaces and systems. In this paper, we address that gap with applications in robotics as robots are one of the best examples of ubiquitous nonlinear systems.

Principal Component Analysis (PCA), in which data points are modeled over a linear or affine space, is the most well-known unsupervised data analysis method for dealing with high-dimensional data. However, over decades, its intrinsic deficiencies particularly in nonlinear regimes has motivated research towards alleviating the corresponding shortcomings. Geometrically, the developed methods can be classified into two rudimentary categories: *distance preservation* and *topology preservation* (see [4] for an inclusive review). Distance-preserving approaches, depending on the selected metric, are based on pairwise distances of the given data points. Technically, all of them strive to solve a maximization problem of finding top eigenvectors, which construct the embedding space. Even though metric multidimensional scaling (MDS) [5] was proposed as the pioneering distance-preserving method, it is still linear like PCA. The first well-known works, in which the transformation is nonlinear, are nonmetric MDS [6] and Sammon's nonlinear mapping (NAM) [7]. Thereafter, other

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

researchers attempted to amend those approaches by either stochastic techniques, such as curvilinear component analysis (CCA) [8], or even replacing Euclidean distance with other distance metrics, such as geodesic distance in [9]. Nevertheless, computing the geodesic distance is mathematically complex and, consequently, computationally intractable. Therefore, some other methods, like Isomap [10], and curvilinear distance analysis (CDA) [11], utilized graph distance metric instead to approximate it. Despite outperforming the Euclidean metric based approaches, these techniques are still incapable of capturing all possible manifolds. There are some other methods that cannot be classified into the aforementioned categories. Kernel PCA (KPCA) [12], semidefinite embedding (SDE), also known as maximum variance unfolding (MVU) [13], and generalized PCA [14] can be enumerated as some outstanding representatives of such approaches. Kernel PCA tries to overcome the nonlinearity of data using some predetermined kernel function which transforms the data into a higher-dimensional space. However, selecting the best-suited kernel function has always been problematic. To mitigate this issue MVU attempts to learn the best kernel from the given data. Vidal et al. [14] fit a union of linear subspaces into the data points that suffices to solve the motion segmentation problem in hand.

On the other hand, topology-preserving methods work with some kind of similarity measure rather than explicit use of pairwise distances. In contrast to distance-preserving methods, they are sturdy but arduous to implement. Kohonen's self-organizing map (SOM) [15] can be addressed as the first prominent method in this category. Other subsequent works, such as generative topographic mapping (GTM) [16] (in which SOM is reformulated within probabilistic framework), locally linear embedding (LLE) [17], and Isotop [18], are among various endeavors to remedy the SOM inadequacies. Regardless of which category those methods belong to, almost none of them succeeds when the intrinsic dimensionality of the given data exceeds four [4]. Moreover, because most of them struggle to embed the data into a new space, loss of information seems inevitable applying these approaches. In this paper, we address these issues by introducing a new methodology, called Principal Variety Analysis (PVA), as a general framework for analytical modeling of given data points over an *algebraic* space with arbitrary dimension. PVA eschews embedding the data into a new space thus causes no information loss. These extracted kinematic models of the surrounding objects can be utilized by motion planners, for instance, for either better roadmap approximation [19] or coping with dynamicity of the environment [20, 21].

Considering the problem as variety learning, Heldt et al. [22] introduce a numerically stable approximate vanishing ideal algorithm. Livni et al. [23] focus on compact representation of the vanishing ideals. Király and Tomioka [24], Király et al. [25] outline ideal-kernel-duality to extract discriminative and generative components from the data and address some other alternatives for rank minimization problem. Comparing to PVA, none of these works concern about the dimensionality optimization of the resulting variety of the vanishing ideals.

The dimensionality reduction problem can be cast as providing a low-dimensional representation of the given data as well. Technically, multivariate polynomial interpolation (MPI) [26] can be contemplated as a formidable formulation to do so. Interpolation theory for univariate functions has been well-developed over a long period of time but in the case of multivariate functions it is still immature. The conventional way to sort out the univariate version of the problem goes through solving a Vandermonde linear system [27] or LU factorization of it [28]. Basically, the most conspicuous strategy to tackle the MPI problem is to generalize existing tools for univariate case, such as Vandermonde linear system and LU factorization, to the multivariate one. However, in most of the works that embrace this tactic the number of points determine the dimension of the interpolating manifold [28]. On the other hand especially in the polynomial case, the problem gets more complicated because, considering the Haar-Mairhubert-Curtis theorem [29], multivariate polynomials are incapable of forming a Haar space. The Padua points [30] and the corresponding interpolation method is a technique to handle this problem. Nevertheless, it gets unstable while computing the basic Lagrange polynomials. To resolve that, Chebyshev polynomials of second kind are attained for interpolation [31]. Interestingly, the closest approach to PVA is the Kronecker's pioneering work [32]. There, it is assumed that the number of polynomials equals the number of variables, so a square nonlinear system of equations is solved symbolically to find the corresponding variety. First, comparing to PVA, the Kronecker's method cannot be practically applied to high-dimensional problems because it is computationally intractable due to symbolic computations. Second, PVA does not put any restriction on the number of polynomials.

The data driven approach matches the robotics spirit as robots essentially interact with the real world while constantly sensing and modeling it. In the past, operating articulated objects has been well-studied in many works [33] in which most of them are either model-free or have considerable prior knowledge about the model and its related parameters. However as robots are gradually introduced into more complex environments, kinematics of different kinds of objects which robots interact with and/or manipulate cannot be assumed to be necessarily known in advance. In general, a robot needs both dynamic and kinematic models of its surrounding objects to have a comprehensive understanding of its environment. The introduced framework in [34] struggles to extract a specific class of kinematic models, *planar* models, through graphical representation. In [35], models and architectures for controlling robot body and external controllable objects have been investigated. A recent work on dynamically modeling objects can be found in [36] in which it is applied to predict rigid body behavior from recorded trajectories of objects. The proposed approach employs kinematic optimization to find a feasible pose that satisfies all of the kinematic constraints. Probabilistic model learning is another way to model articulated objects kinematically in unstructured environments [37]. Nonetheless, it suffers from both maintaining a kinematic graph and performing an additional step to reduce the dimension. Furthermore, it is based on a limited set of pre-coded candidate models: rigid, prismatic, revolute, and Gaussian. Extracting kinematic model of objects' structure is considered to be a potential intriguing application of PVA. However, PVA is a general purpose method which is expected to have many other important applications.

2 Technical Background

We are given $D = \{p_1, p_2, \dots, p_k\} \in \mathbb{R}^n$ a set of data points that are assumed to lie on an unknown, low-dimensional, real algebraic variety W. We assume there are no outliers in D. The problem is to find V, an approximation of W the variety from which data points are captured, through computing the polynomials $f_1, f_2, \dots, f_m \in \mathbb{R}[X]$ that define

$$V = \mathcal{Z}(f) = \{x \in \mathbb{R}^n \mid f_i(x) = 0, \;\; i = 1, 2, \dots, m\}, \tag{1}$$

in which $\mathcal{Z}$ denotes the real zero set. Throughout this paper, we denote variables by $X = (X_1, X_2, \dots, X_n)$ and values by $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$.

3 Problem Formulation

It is clear that if we put no upper bound on the degrees of f_i's, then V simply degenerates into a zero-dimensional point set on every p_j. Hence, we assume that $\deg(f_i) \leq d$ for $i = 1, 2, \dots, m$, and $d \in \mathbb{N}$.

The points in D have to lie on V, i.e. $D \subseteq V$ if D is noiseless. For brevity and without loss of generality, we assume the data points are noiseless in the rest of this paper. Should it be necessary, we mention important changes in the presence of noise.

Problem 1 (Principal Variety Analysis) *Given input data set $D \subset \mathbb{R}^n$, the problem is formulated as the following optimization:*

$$\begin{aligned} \underset{V}{\textit{minimize}} \quad & \dim(V) \\ \textit{subject to} \quad & V = \{x \in \mathbb{R}^n \mid f_i(x) = 0, \;\; i = 1, 2, \dots, m\}, \\ & D \subseteq V, \\ & \deg(f_i) \leq d, \; 1 \leq i \leq m, \end{aligned} \tag{2}$$

in which dim represents the dimension of a variety and deg is the degree of a polynomial.

Above, we need a practically powerful definition of variety dimension that can also be conveniently and efficiently computed. Below, we present our definition of variety dimension for which we will give an efficient algorithm in the following sections.

3.1 Dimension of a Real Algebraic Variety

In this paper, we use a geometric notion of the dimension of an algebraic variety, which serves our quest the best. Mathematically, the real dimension of a real algebraic set is the dimension of its Zariski closure. We use a slightly different definition here that can be computed more easily.

Definition 1 (Dimension of Variety) *The dimension of a real algebraic variety V, which is the zero set of $f = (f_1, f_2, \dots, f_m)$ real polynomials, is*

$$\dim(V) = \max_{p \in V} \left(n - \mathit{rank} \left. \frac{\partial f}{\partial x} \right|_p \right) = \quad n - \min_{p \in V} \ \mathit{rank} \left. \frac{\partial f}{\partial x} \right|_p . \tag{3}$$

Above, $\frac{\partial f}{\partial x}$ is the Jacobian of f. Therefore, rank $\left.\frac{\partial f}{\partial x}\right|_p$ is the dimension of the orthogonal (co-tangent) space at point p and $n -$ rank $\left.\frac{\partial f}{\partial x}\right|_p$ is the dimension of the tangent space at point p. Maximum tangent space dimension over all points of the variety gives the dimension of variety.

With this definition, Problem 1 is rewritten as

$$\begin{array}{ll} \underset{V}{\text{maximize}} & \min_{p \in V} \ \text{rank} \left. \frac{\partial f}{\partial x} \right|_p \\ \text{subject to} & V = \{x \in \mathbb{R}^n \mid f_i(x) = 0, \ \ i = 1, 2, \dots, m\}, \\ & D \subseteq V, \\ & \deg(f_i) \le d, \ 1 \le i \le m. \end{array} \tag{4}$$

However, solving the optimization (4) is not easy since rank minimization is essentially an ℓ_0 norm minimization. The vast majority of rank (ℓ_0 norm) minimization problems are NP-Hard, even in affine subspaces, and for most practical problems there is no efficient algorithm that yields an exact solution. This is predominantly because of the non-convex nature of the rank optimization problem [38]. Hence, we seek a proxy for the rank function that is guaranteed to yield the same optimal solution under reasonable sparsity assumptions and whose minimization is computationally tractable.

4 Sparsity Comes to Rescue

4.1 Nuclear Norm Minimization

Fazel et al. [39] proposed a popular heuristic algorithm that replaces the rank function with the nuclear norm in the matrix completion framework [40]. Nuclear norm of a matrix, in this case the Jacobian, is the sum of the matrix singular values. More precisely, let for the Jacobian matrix $\frac{\partial f}{\partial x} \in \mathbb{R}^{m \times n}$ of f, $\sigma_i(\cdot)$ be the ith largest singular value. The nuclear norm of the matrix $\frac{\partial f}{\partial x}$ is defined as

$$\left\| \frac{\partial f}{\partial x} \right\|_* = \sum_{i=1}^{n} \sigma_i(\frac{\partial f}{\partial x}). \tag{5}$$

Recht et al. [40] provided a necessary and sufficient condition that determines when this heuristic successfully finds the minimum rank solution of a linear constraint set [38].

4.2 Dimension of a Sparse Real Algebraic Variety

Recht et al. [40] showed that when the linear map defining the constraint set in rank minimization problem is *restricted isometric*, the nuclear norm is the best convex lower approximation of the rank function.

Although there are currently no proofs for our case of nonlinear constraints, our experimental results in this paper demonstrate that substitution of rank in (4) with nuclear norm is still legitimate, under some conditions, for nonlinear constraint sets. Replacing rank with the nuclear norm, Problem 1 is then rewritten as

$$\begin{array}{ll} \underset{V}{\text{maximize}} & \text{rank} \left. \frac{\partial f}{\partial x} \right|_{p^*} \\ \text{subject to} & V = \{x \in \mathbb{R}^n \mid f_i(x) = 0, \ \ i = 1, 2, \dots, m\}, \\ & D \subseteq V, \\ & \deg(f_i) \le d, \ 1 \le i \le m, \end{array} \tag{6}$$

in which

$$p^* = \underset{p \in V}{\operatorname{argmin}} \quad \left\| \left. \frac{\partial f}{\partial x} \right|_p \right\|_* . \tag{7}$$

5 Algorithm

5.1 Inner Polynomial Optimization

The inner nuclear norm minimization in (7) can be written as a polynomial optimization on an algebraic (semialgebraic in the case of noisy data) set. Recall that the nuclear norm of a matrix $\mathbf{J}$ is

$$\|\mathbf{J}\|_* = \operatorname{trace}(\sqrt{\mathbf{J}^T \mathbf{J}}), \tag{8}$$

in which $\sqrt{\mathbf{B}} = \mathbf{C}$ if and only if $\mathbf{C}^2 = \mathbf{B}$. Hence for a given $f = (f_1, f_2, \ldots, f_m)$ such that $D \subseteq \mathcal{Z}(f) = V$, the nuclear norm minimization problem in (7) becomes finding the point for which the following optimization problem attains its solution:

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \operatorname{trace}(\mathbf{C}) \\ \text{subject to} \quad & \mathbf{C}^2 = \left. \frac{\partial f}{\partial x} \right|_q^T \left. \frac{\partial f}{\partial x} \right|_q , \\ & f(q) = 0, \\ & q \in \mathbb{R}^n. \end{aligned} \tag{9}$$

Note that this is the crucial phase of the algorithm for detecting possible singularities that would affect the dimensionality of the resulting variety.

5.2 Outer Optimization

Since the optimization problem formulated in (6) does not currently have a known closed form solution, we exploited a breadth-first technique to solve the outer optimization. To do so, all possible candidates can be formulated as null space of a homogeneous system of linear equations below.

For $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{N}^n$, we define

$$\Lambda(d) := \{\alpha \in \mathbb{N}^n | \, |\alpha| \leq d\}, \tag{10}$$

in which $|\alpha| := \alpha_1 + \alpha_2 + \ldots + \alpha_n$. Hence, a degree-constrained polynomial, f_i, can be written as

$$f_i = \sum_{\alpha \in \Lambda(d)} c_{\alpha,i} X^\alpha, \tag{11}$$

where $c_{\alpha,i}$'s are coefficients of corresponding monomials in f_i. Given the set D, by substituting the D data points in (11) the homogeneous system of linear equations is constructed as

$$\mathbf{A}c = 0, \tag{12}$$

in which matrix $\mathbf{A}$ contains the numbers obtained by evaluating the D data points in $(X^\alpha)_{\alpha \in \Lambda(d)}$ and $\mathbf{c}$ is the corresponding coefficients variable vector in (11). It is required to keep the number of degree-constrained polynomials less than or equal to the number of elements in $\Lambda(d)$ that have nonzero corresponding coefficient (which is usually the case in practice). Thus, the system of linear equations (12) is guaranteed to be never over-determined. The general solution for (12) is written as

$$c = \lambda_1 n_1 + \lambda_2 n_2 + \ldots + \lambda_k n_k, \tag{13}$$

where $n_1, n_2, \ldots, n_k$ is a basis for the kernel of $\mathbf{A}$ and λ_i's are picked randomly. The resulting coefficient vector, c, is utilized in (11) to produce a particular polynomial out of the possible candidates. PVA commences exploring for a lowest dimensional variety by selecting $m = n - 1$. If it fails to find a solution it will seek for a higher dimensional variety incrementally (See Algorithm 1 in Appendix for more details).

(a) (b) (c) (d)

Figure 1: The PVA constituent polynomials for the location of (a) the microwave door corner vs a fixed feature on the body (LED display) and (b) the drawer corner vs a fixed feature on the counter. The PVA variety of (c) the microwave and (d) the drawer.

6 Experimental Results

We developed a prototype implementation of PVA in MATLAB as a proof of concept. Faster implementations are conceivable using more advanced optimization libraries written in C. PVA was run on a machine with 64 AMD Opteron 6386 SE 2.8 GHz cores and 512 GB RAM. The resulting optimization problems were solved using YALMIP toolbox [41]. Experiments were performed in two groups.

First, we selected some well-known complex test cases and fed them into the inner (polynomial) optimization step of the PVA to examine its performance. We scrutinized the inner nuclear norm minimization on three major scenarios: (i) curves embedded in $\mathbb{R}^3$, (ii) surfaces in $\mathbb{R}^3$, and (iii) spheres in n-dimensional spaces. The experiments show that PVA correctly computes the dimension of input varieties within a few seconds (see Table 2 in Appendix for more details).

Second, we performed experiments on three different objects to illustrate performance of the entire PVA including the outer optimization:

1. **Microwave Door:** We put a marker on the corner of a home microwave door and recorded ten 3D coordinates of the marker against a fixed feature on the microwave body, the LED display. The data points in D were $(p_{door}, p_{led}) \in \mathbb{R}^6$ in this case. Fig. 1 (a) and (c) illustrate the PVA variety with the constituent polynomials.

2. **Drawer:** We put a marker on the corner of a drawer and recorded ten 3D coordinates of the marker against a fixed feature on the counter. The data points in D were $(p_{drawer}, p_{counter}) \in \mathbb{R}^6$ in this case. Fig. 1 (b) and (d) illustrate the PVA variety with the constituent polynomials.

3. **Puma 560:** We simulated trajectory of a 6-DOF Puma 560 robot using Robotics Toolbox [42]. In each experiment, ten configurations were sampled from the trajectory and the 3D coordinates of the joints were used as data points. The data points in D were $(p_1, p_2, p_3, p_4, p_5, p_6) \in \mathbb{R}^{18}$ where p_i is the 3D coordinates of the i^{th} joint. First, two completely random experiments were executed. In one of them, eight coordinates of the 18 coordinates in D were constant and in the other, all of the 18 coordinates were varying. Therefore, we performed PVA both on all 18 coordinates and the 10 varying coordinates respectively in $\mathbb{R}^{18}$ and $\mathbb{R}^{10}$. Figures 2 and 3 (a) illustrate the ten sampled configurations on the robot trajectory and the end effector location computed from the one dimensional PVA variety on which the features (joint coordinates) lie, respectively. Second, we conducted ten random independent experiments and collected ten samples of each experiment. Figure 3 (b) demonstrates the two dimensional PVA variety on which the end effector in all ten random experiments lies.

In all experiments, the resulted variety and the corresponding polynomials exactly pass through the given data points. Table 1 shows the running time, dimension of output variety, and the number of polynomials for each of the experiments above. Performance of the inner optimization problem solver was verified over hundreds of ersatz and well-known varieties. The algorithm was impeccable in determining the dimension of all the test cases in palatable amount of time.

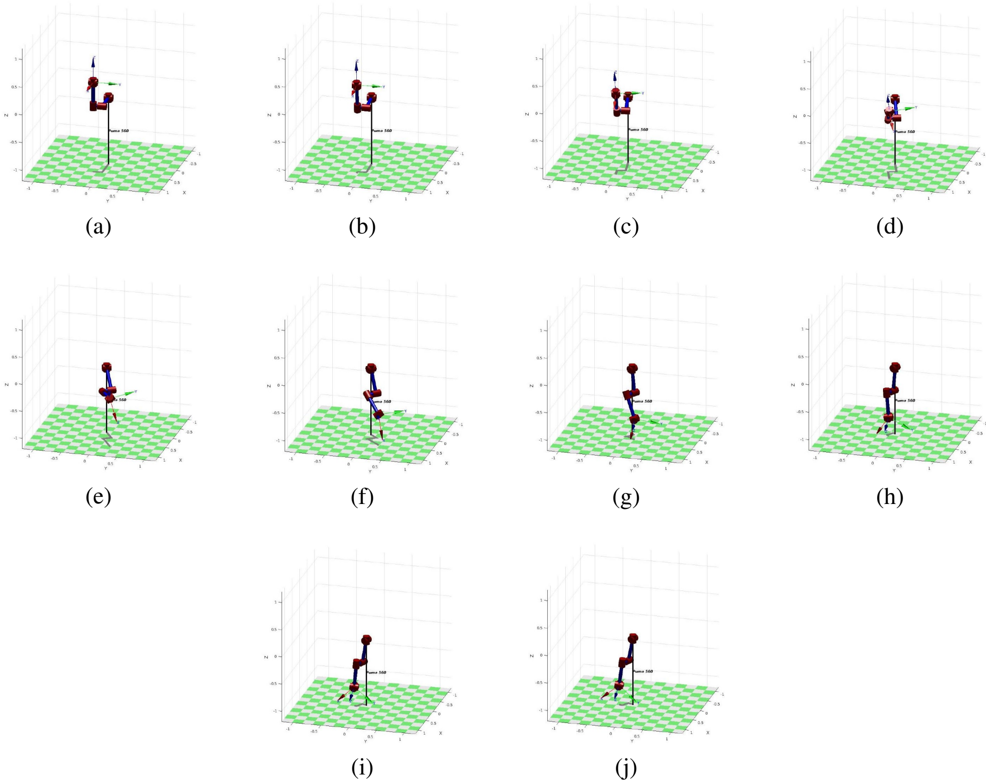

Figure 2: Ten sampled configurations on a trajectory of a Puma 560 robot in a single experiment.

Figure 3: End effector location computed from the PVA of joints coordinates in 18D. (a) Resulting 1D variety after 1 random experiment. (b) Resulting 2D variety after 10 random experiments.

7 Conclusion and Future Work

Principal Variety Analysis (PVA) was introduced in this paper as a general framework for primarily nonlinear data modeling. Its performance was demonstrated for modeling objects kinematics. Unlike the other object modeling approaches, which are based on recorded trajectories of objects, PVA just exploits the recorded coordinates of some pre-specified features on the objects. Therefore, it requires much less information to provide an analytical kinematic model of the objects, even in unstructured, uncertain environments. Furthermore, in contrast to local interpolation approaches, PVA can potentially reduce the effect of noise and non-smooth data. Our framework accommodates two methods to handle noise:

Table 1: Running Times, Dimensions, and Number of Polynomials for the PVA Experiments (the maximum polynomial degrees were set to four in all experiments).

Experiment	No. of Polys	Output Dim	Running Time (s)
Microwave	2	1	0.65
Drawer	2	1	1.15
Puma 560 (10D)	9	1	169.51
Puma 560 (18D)	17	1	18861.17
Puma 560 (18D)	16	2	13401.72
Puma 560 (18D, 10 random experiments)	16	2	14358.87
Puma 560 (18D)	15	3	13342.30
Puma 560 (18D)	14	4	13512.64
Puma 560 (18D)	13	5	13891.89
Puma 560 (18D)	12	6	13231.88

- **Implicit:** Replace $D \subseteq V$ with adding a Lagrange multiplier for the constraints,

$$\lambda \sum_{i=1}^{m} \sum_{j=1}^{k} f_i^2(p_j),$$

to the objective function, i.e. minimizing distance from the fitted variety.

- **Explicit:** Allow the data points to be off of V by a small distance, i.e. $f_i^2(p_j) \leq \epsilon$ for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, k$, and a small $\epsilon \geq 0$.

Both methods above would work well with the PVA framework and left as an interesting research for future work.

Besides, PVA is not restricted to just some predetermined model templates of objects and is capable of extracting any kind of model the object may possess. In addition to finding kinematic model of objects, PVA can be considered as a powerful tool to estimate their corresponding Degrees of Freedom (DOFs). Its computational success depends on dimension minimization through replacement of intractable ℓ_0 norm (rank) with tractable ℓ_1 norm (nuclear norm). Previous results prove validity of that replacement for linear spaces. Our simulation results demonstrate that substitution still holds for nonlinear spaces. A formal proof is beyond the scope of this paper and remains as future work.

References

[1] S. Dasgupta and A. Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, Jan. 2003. ISSN 1042-9832. doi:10.1002/rsa.10073.

[2] G. Carlsson. Topology and Data. *Bull. Amer. Math. Soc.*, pages 255–308, 2009.

[3] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec 2000.

[4] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 0387393501, 9780387393506.

[5] G. Young and A. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.

[6] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[7] J. Sammon. A Nonlinear Mapping for Data Structure Analysis. *Computers, IEEE Transactions on*, C-18(5):401–409, May 1969. ISSN 0018-9340. doi:10.1109/T-C.1969.222678.

[8] P. Demartines and J. Herault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on*, 8(1):148–154, Jan 1997. ISSN 1045-9227. doi:10.1109/72.554199.

[9] P. Estevez and A. Chong. Geodesic Nonlinear Mapping Using the Neural Gas Network. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 3287–3294, 2006.

[10] J. B. Tenenbaum. Mapping a Manifold of Perceptual Observations. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 682–688. MIT Press, 1998.

[11] J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A robust non-linear projection method. In *ESANN*, pages 13–20, 2000.

[12] B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998. ISSN 0899-7667. doi:10.1162/089976698300017467.

[13] K. Weinberger and L. K. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-04)*, volume 2, pages 988–995, Washington D.C., 2004.

[14] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1945–1959, Dec. 2005. ISSN 0162-8828. doi:10.1109/TPAMI.2005.244.

[15] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982. ISSN 1432-0770. doi:10.1007/BF00337288.

[16] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: a principled alternative to the self-organizing map. In *International Conference on Artificial Neural Networks, ICANN'96*, volume 9, page 165–170. Springer, January 1997.

[17] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[18] J. A. Lee, C. Archambeau, and M. Verleysen. Locally linear embedding versus isotop. In *Proceedings of ESANN'2003, 11th European Symposium on Artificial Neural Networks*, pages 527–534, 2003.

[19] R. Iraji and H. Chitsaz. NUROA: A numerical roadmap algorithm. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5359–5366, Dec 2014. doi:10.1109/CDC.2014.7040227.

[20] R. Iraji and M. Manzuri-Shalmani. A new fuzzy-based spatial model for robot navigation among dynamic obstacles. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 1323–1328, May 2007. doi:10.1109/ICCA.2007.4376575.

[21] R. Iraji and M. T. Manzuri-Shalmani. AMF: A novel reactive approach for motion planning of mobile robots in unknown dynamic environments. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1698–1703, Dec 2009. doi:10.1109/ROBIO.2009.5420410.

[22] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation*, 44(11):1566 – 1591, 2009. ISSN 0747-7171. doi:http://dx.doi.org/10.1016/j.jsc.2008.11.010.

[23] R. Livni, D. Lehavi, S. Schein, H. Nachlieli, S. Shalev-Shwartz, and A. Globerson. Vanishing Component Analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

[24] F. Király and R. Tomioka. A Combinatorial Algebraic Approach for the Identifiability of Low-Rank Matrix Completion. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 967–974, New York, NY, USA, July 2012. Omnipress. ISBN 978-1-4503-1285-1.

[25] F. J. Király, L. Theran, and R. Tomioka. The Algebraic Combinatorial Approach for Low-rank Matrix Completion. *J. Mach. Learn. Res.*, 16(1):1391–1436, Jan. 2015. ISSN 1532-4435.

[26] M. Gasca and T. Sauer. On the history of multivariate polynomial interpolation. *Journal of Computational and Applied Mathematics*, 122(1–2):23 – 35, 2000. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/S0377-0427(00)00353-8. Numerical Analysis in the 20th Century Vol. II: Interpolation and Extrapolation.

[27] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2002. ISBN 0898715210.

[28] P. J. Olver. On Multivariate Interpolation. *Studies in Applied Mathematics*, 116:201–240, 2006. doi:10.1111/j.1467-9590.2006.00335.x.

[29] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004. ISBN 9780521843355.

[30] M. Caliari, S. Marchi, A. Sommariva, and M. Vianello. Padua2DM: Fast Interpolation and Cubature at the Padua Points in Matlab/Octave. *Numerical Algorithms*, 56(1):45–60, Jan. 2011. ISSN 1017-1398. doi:10.1007/s11075-010-9373-1.

[31] L. Bos, M. Caliari, S. De Marchi, and M. Vianello. A numerical study of the Xu interpolation formula in two variables. *Computing*, 76(3–4):311–324, 2006.

[32] L. Kronecker. Über einige interpolationsformeln für ganze funktionen mehrerer variabeln. *Lecture at the academy of sciences*, Dec. 1865. H. Hensel (Ed.), L. Kroneckers Werke, Vol. I, Teubner, Stuttgart, 1895, pp. 133—141. (reprinted by Chelsea, New York, 1968).

[33] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger. Autonomous door opening and plugging in with a personal robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 729–736, May 2010. doi:10.1109/ROBOT.2010.5509556.

[34] D. Katz and O. Brock. Extracting Planar Kinematic Models Using Interactive Perception. In *In Unifying Perspectives In Computational and Robot Vision*, volume 8 of *Lecture Notes in Electrical Engineering*, pages 11–23. Springer Verlag, May 2008. ISBN 978-0-387-75521-2.

[35] D. Nguyen-Tuong and J. Peters. Model Learning in Robotics: a Survey. *Cognitive Processing*, Oct. 2011.

[36] D. Belter, M. Kopicki, S. Zurek, and J. Wyatt. Kinematically optimised predictions of object motion. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4422–4427, Sept 2014. doi:10.1109/IROS.2014.6943188.

[37] J. Sturm. *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. Springer Publishing Company, Incorporated, 2013. ISBN 3642371590, 9783642371592.

[38] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[39] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference*, volume 6, pages 4734–4739. IEEE, 2001.

[40] B. Recht, W. Xu, and B. Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. In *IEEE Conf. Decision & Control*, pages 3065–3070, Dec 2008. doi:10.1109/CDC.2008.4739332.

[41] J. Lofberg. YALMIP: a toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289, Sept 2004. doi:10.1109/CACSD.2004.1393890.

[42] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2011. ISBN 9783642201431.

8 Appendix

Algorithm 1 PVA algorithm

procedure PVA(**dataPoints**, spaceDim, maxDegree, range)
  $pointsNo \leftarrow size(\textbf{dataPoints}, 2)$
  $n \leftarrow spaceDim + spaceDim * spaceDim$
  $\vec{x} \leftarrow sdpvar(1, n)$
  $\overrightarrow{monomials} \leftarrow monolist(\vec{x}(1 : spaceDim), maxDegree)$

  $\mathbf{B} \leftarrow \emptyset$
  for $i \leftarrow 1, pointsNo$ **do**
    $assign(\vec{x}(1 : spaceDim), \textbf{dataPoints}(i))$
    $\mathbf{B} \leftarrow \mathbf{B} \cup$ new row $value(\overrightarrow{monomials})^T$
  end for
  basis $\leftarrow nullSpace(\mathbf{B})$
  $coeffNo \leftarrow size(\textbf{basis}, 1)$
  $basisNo \leftarrow size(\textbf{basis}, 2)$

  $polyNo \leftarrow spaceDim$
  $solved \leftarrow False$
  repeat
    $polyNo \leftarrow polyNo - 1$
    for $i \leftarrow 1, polyNo$ **do**
      $\vec{\lambda} \leftarrow rand(1, range, basisNo)$

      $\overrightarrow{coeffs} \leftarrow 0$
      for $j \leftarrow 1, basisNo$ **do**
        $\overrightarrow{coeffs} \leftarrow \overrightarrow{coeffs} \cup \vec{\lambda}(j) * \textbf{basis}(:, j)$
      end for
      $f(i) \leftarrow \overrightarrow{coeffs}^T * \overrightarrow{monomials}$
    end for
    $\mathbf{C} \leftarrow reshape(\vec{x}(spaceDim + 1 : n), [spaceDim, spaceDim])^T$
    $obj \leftarrow trace(\mathbf{C})$
    $F \leftarrow \emptyset$
    for $i \leftarrow 1, polyNo$ **do**
      $F \leftarrow F \cup (f(i) \geq 0)$
      $F \leftarrow F \cup (-f(i) \geq 0)$
    end for
    $\mathbf{J} \leftarrow jacobian([f(1 : polyNo)]^T, \vec{x}(1 : spaceDim))$
    constraints $= \mathbf{C}^2 - \mathbf{J}^T * \mathbf{J}$
    for $i \leftarrow 1, spaceDim$ **do**
      for $j \leftarrow 1, spaceDim$ **do**
        $F \leftarrow F \cup (\textbf{constraints}(i, j) \geq 0)$
        $F \leftarrow F \cup (-\textbf{constraints}(i, j) \geq 0)$
      end for
    end for
    $solved \leftarrow optimize(F, obj)$
  until $solved = True$
  $eval \leftarrow replace(\mathbf{J}, \vec{x}(1 : spaceDim), value(\vec{x}(1 : spaceDim)))$
  $varietyDim \leftarrow spaceDim - rank(full(eval))$
  return $varietyDim, f$
end procedure

Table 2: Determination of Dimension for Test Case Varieties in the Inner Polynomial Optimization Step of the PVA. For numerical considerations, all varieties are centered at $(1, \ldots, 1)$.

	CASE	ALGEBRAIC FORMULA	DIM	TIME (S)
CURVES IN $\mathbb{R}^3$[1]	CLOVER	$(x-1)^4+2(x-1)^2(y-1)^2+(y-1)^4-(x-1)^3+3(x-1)(y-1)^2$	1	9.4529
	AMPERSAND	$\left[(y-1)^2-(x-1)^2\right](x-2)\left[2(x-1)-3\right]-4\left[(x-1)^2+(y-1)^2-2(x-1)\right]^2$	1	0.4239
	BICUSPID	$\left[(x-1)^2-1\right](x-2)^2+\left[(y-1)^2-1\right]^2$	1	0.1332
	BOW	$(x-1)^4-(x-1)^2(y-1)+(y-1)^3$	1	6.2361
	ELLIPTIC	$(y-1)^2-(x-1)^3+x-2$	1	1.0470
	WATT'S CURVE	$\left[(x-1)^2+(y-1)^2\right]\left[(x-1)^2+(y-1)^2-1\right]^2+4(y-1)^2\left[(x-1)^2+(y-1)^2-1\right]$	1	0.3598
SURFACES	CALYPSO	$(x-1)^2+(y-1)^2(z-1)-(z-1)^2$	2	1.2318
	DINGDONG	$(x-1)^2+(y-1)^2+(z-1)^3-(z-1)^2$	2	6.3090
	DISTEL	$(x-1)^2+(y-1)^2+(z-1)^2+1000\left[(x-1)^2+(y-1)^2\right]\left[(x-1)^2+(z-1)^2\right]\left[(y-1)^2+(z-1)^2\right]-1$	2	8.6266
	DURCHBLICK	$(x-1)^3(y-1)+(x-1)(z-1)^3+(y-1)^3(z-1)+(z-1)^3+5(z-1)$	2	7.4589
	EISTÜTE	$\left[(x-1)^2+(y-1)^2\right]^3-4(x-1)^2(y-1)^2\left[(z-1)^2+1\right]$	2	8.5421
	EVE	$5(x-1)^2+2(x-1)(z-1)^2+5(y-1)^6+15(y-1)^4+5(z-1)^2-15(y-1)^5-5(y-1)^3$	2	5.5867
	FLIRT	$(x-1)^2-(x-1)^3+(y-1)^2+(y-1)^4+(z-1)^3-10(z-1)^4$	2	18.1745
	GEISHA	$(x-1)^2(y-1)(z-1)+(x-1)^2(z-1)^2-(y-1)^3(z-1)-(y-1)^3$	2	3.9184
	HARLEKIN	$(x-1)^3(z-1)+10(x-1)^2(y-1)+(x-1)(y-1)^2+(y-1)(z-1)^2-(z-1)^3$	2	7.4701
	HELIX	$6(x-1)^2-2(x-1)^4-(y-1)^2(z-1)^2$	2	7.1149
	HERZ	$(y-1)^2+(z-1)^3-(z-1)^4-(x-1)^2(z-1)^2$	2	6.5586
	KOLIBRI	$(x-1)^3+(x-1)^2(z-1)^2-(y-1)^2$	2	27.0135
	LEOPOLD	$(x-1)^2(y-1)^2(z-1)^2+3(x-1)^2+3(y-1)^2+(z-1)^2-1$	2	8.5942
	OCTDONG	$(x-1)^2+(y-1)^2+(z-1)^4-(z-1)^2$	2	2.0404
	PLOP	$(x-1)^2+\left[(z-1)+(y-1)^2\right]^3$	2	0.3425
	SOFA	$(x-1)^2+(y-1)^3+(z-1)^5$	2	6.6046
	TORUS	$\left[(x-1)^2+(y-1)^2+(z-1)^2+3\right]^2-16\left[(x-1)^2+(y-1)^2\right]$	2	7.1438
HYPER-SURFACES	3D SPHERE	$(x_1-1)^2+(x_2-1)^2+(x_3-1)^2+(x_4-1)^2-4$	3	4.3512
	3D CALABI-YAU	$(x_1-1)^5+(x_2-1)^5+(x_3-1)^5+(x_4-1)^5+(x_1-1)(x_2-1)(x_3-1)(x_4-1)-1$	3	7.7645
	4D SPHERE	$(x_1-1)^2+(x_2-1)^2+(x_3-1)^2+(x_4-1)^2+(x_5-1)^2-4$	4	21.9351
	4D CALABI-YAU	$(x_1-1)^5+(x_2-1)^5+(x_3-1)^5+(x_4-1)^5+(x_5-1)^5+(x_1-1)(x_2-1)(x_3-1)(x_4-1)(x_5-1)-1$	4	13.3629
	5D SPHERE	$(x_1-1)^2+(x_2-1)^2+(x_3-1)^2+(x_4-1)^2+(x_5-1)^2+(x_6-1)^2-4$	5	7.3037
	5D CALABI-YAU	$(x_1-1)^5+(x_2-1)^5+(x_3-1)^5+(x_4-1)^5+(x_5-1)^5+(x_6-1)^5+(x_1-1)(x_2-1)(x_3-1)(x_4-1)(x_5-1)(x_6-1)-1$	5	15.9658

[1] CURVES IN $\mathbb{R}^3$ HAVE BEEN INTERSECTED BY z PLANE.