Estimation methods in the errors-in-variables context

Summary of PhD dissertation

Levente Hunyadi

Budapest University of Technology and Economics
Department of Automation and Applied Informatics

Advisor:

Dr. István Vajk, DSc
professor, head of department
Budapest University of Technology and Economics
Department of Automation and Applied Informatics

Budapest, Hungary
2013

1 Introduction

A recurring problem in engineering is constructing a computer model from a possibly large set of data. In fields such as computer vision, pattern recognition, image reconstruction, speech and audio processing, signal processing, modal and spectral analysis, data mining, system identification, econometrics or time series analysis, the goal is often to identify or describe the internal laws that govern a system rather than to predict its future behavior. In other words, one is interested in reconstructing how the measured variables are related given a set of observations. The amount and dimensionality of the observed data may be large yet one is often able to express the equation that relates data points in a succinct manner. In a system identification context, for instance, one could be interested in the parameters of a discrete-time dynamic system model, based on a measured input and output data sequence, contaminated with noise. In a pattern recognition context, on the other hand, one would seek to capture a set of unorganized data points with a number of simple shapes, such as lines, ellipses, parabolas, etc. While these tasks appear to be wildly different, they share some common properties, which outline the characteristics of *parametric errors-in-variables* systems:

- The relationships that characterize the system under investigation admit a structure. Even if the data accumulated may be large, the number of parameters is limited: the system can be explained by a set of relatively simple relationships, each known up to the parameters. A discrete-time dynamic system may be described by a low-order polynomial that relates the output variable with past values of the input and output variables, even if the data sequences may comprise of thousands of observations. Points obtained from a manufactured object by a laser scanner could be modeled with a couple of quadric surfaces.

- There are no distinguished variables in general. The equation that relates data points has the implicit form $f(\mathbf{x}) = 0$ rather than an explicit form $y = f(\mathbf{x})$. In a 3D scan, there is no special significance of any of the coordinates x, y or z, data may be rotated and translated.

- The measured data is contaminated with noise. Unlike the usual assumption in statistics, there is in most cases no meaningful distinction between independent (noise-free) and dependent (noisy) variables. In the errors-in-variables context all variables are assumed to be measured quantities, hence contaminated with noise. Both the input and the output of the dynamic system is treated as a sequence of noisy measurements, and the points that are captured with a scanner are polluted with noise in all coordinates.

Identifying the underlying relationship that governs a system gives us a compact representation that is easier to manipulate and a key to understanding. For instance, a pattern recognition algorithm that exploits that an approximated 85% of manufactured objects can be modeled with quadratic surfaces [CJ93], and subsequently reduces to fitting these surfaces, uses far fewer parameters than a primarily non-parametric approach that uses only locality information. Furthermore, the reverse-engineered model lends itself better to future transformations such as constructive solid geometry operations.

When the relationship is linear, standard tools in mathematics and statistics, such as singular value decomposition, can be applied to static systems, and iterative algorithms may be formulated for dynamic systems. Nonlinearity in the system, however, makes it difficult to draw conclusions for the original (noise-free) relationship based on (noisy) measured variables; traditional approaches may lead to substantial bias as additive noise contaminates nonlinear variables of the system. The objective of the dissertation is to extend the results for linear errors-in-variables systems to a relatively rich set of the nonlinear case when the system is captured by polynomial functions.

2 Research objectives and methodology

The dissertation investigates algorithms related to identification and recognition problems that belong to the domain of nonlinear parametric errors-in-variables systems. We look into both static and dynamic (time-dependent) systems, with focus on low-order polynomial functions, and assume the noise that contaminates measurements is Gaussian. Contributions address both the pure parametric case, where the system is known up to the unknown parameters, as well as the machine learning case, where we assume the system is known to comprise of a set of constituents, each captured with a well-known structure defined by a set of unknown parameters.

First, we venture onto the field of static systems where the estimation may be subject to constraints. In a computer vision application, for instance, one may be interested in fitting an ellipse rather than a general quadratic curve to an unorganized point set. Previous work in the computer vision domain showed how to integrate constraints such as the quadratic curve representing an ellipse rather than a hyperbola into the estimation but failed to adequately take into account the nonlinear distortions induced by noise. The method proposed in the dissertation takes a step further, and incorporates constraints into parameter estimation while canceling the effect of noise.

Second, we investigate a machine learning application, namely clustering, where the system is originally built up of several constituents, each described by a polynomial relationship, more restrictively quadratic curves and surfaces, but only an unorganized set of noisy data is at our disposal, and we aim to discover the original structure of the system and estimate parameters of the constituents. Several algorithms exist that can tackle the so-called (multi-)subspace clustering or hybrid linear modeling problem, where the objective is to build a model where each partition of the data is captured by a linear relationship and the entire model is a composition of these partitions. Not every data set, however, admits a decomposition into linear relationships, and applying linear methods to essentially nonlinear relationships loses the simplicity and explanation power of these methods. A natural generalization of subspace clustering is (multi-)manifold clustering, which we explore in the dissertation, where each curved manifold is captured by some nonlinear (or polynomial) relationship.

Finally, we look into dynamic systems, where we consider discrete-time systems, and combine a generalization of the Koopmans–Levin (GKL) method [Vaj05] with a nonlinear extension to the original Koopmans (NK) method [VH03]. The GKL method is an approach to estimate parameters of a linear dynamic system with a scalable balance between accuracy and computational cost, whereas the NK method gives a non-iterative approach to estimate

parameters of a static system described by a polynomial function. The algorithm proposed in the dissertation alloys the two approaches, and estimates parameters of dynamic systems whose variables are related with a polynomial function, using a set of data polluted with Gaussian noise.

A working implementation with several examples is essential to the popularization of any new methods. The results in this dissertation are augmented with source code, which demonstrate the feasibility of the proposed algorithms.

3 Novel contributions

Coupled with the outlined research objectives, the results are grouped into three theses. The first thesis investigates static systems and proposes estimation methods for (zero) level sets, both without and with ancillary constraints. The primary assumption of the first thesis is that the entire data set derives from a single relationship (a single partition), with a known polynomial structure, albeit with unknown parameters. The second thesis builds on the first, and extends the fitting problem to multiple partitions, where the goal is both to discover appropriate partitions and to estimate parameters within the partitions. Finally, the third thesis uses results from the first, and generalizes estimation methods for static systems to dynamic systems.

The proposed methods use a statistical approach and compute parameters from sample data covariance matrices. From a numerical perspective, they are based on generalized eigenvalue decomposition, which caters for straightforward implementation on modern hardware.

3.1 Constrained fitting with noise cancellation

Related publications: [8, 10, 11]

Fitting quadratic curves and surfaces, in particular, ellipses and ellipsoids, occurs frequently in computer vision, pattern recognition and image processing applications. Given a parametric zero level set function

$$P(\mathbf{x}, \mathbf{g}) = 0$$

where P represents a quadratic function (a point set), $\mathbf{x}$ stands for the point co-ordinate vector (a free parameter) and $\mathbf{g}$ denotes curve or surface parameters, the goal is to estimate $\mathbf{g}$ from noisy samples $\mathbf{x}_i = \bar{\mathbf{x}}_i + \tilde{\mathbf{x}}_i$ where $\bar{\mathbf{x}}_i$ is noise-free data and $\tilde{\mathbf{x}}_i$ is a noise contribution such that

$$P(\bar{\mathbf{x}}_i, \mathbf{g}) = 0.$$

One possible way to estimate $\mathbf{g}$ from $\mathbf{x}_i$ is to use geometric distance [CM11], i.e. maximum likelihood estimation, and minimize

$$e = \frac{1}{N} \sum_{i=1}^{N} d_i^2$$

where d_i measures the distance from the noisy point $\mathbf{x}_i$ to the curve or surface $P(\mathbf{x}, \mathbf{g}) = 0$. While highly accurate, this approach (and its approximations [MM00, C$^+$05]) yields iterative

methods that may converge slowly (or even diverge) in some situations, and always require a feasible initialization. A more robust approach is to use algebraic distance

$$e = \frac{1}{N}\sum_{i=1}^{N}\left(P(\mathbf{x}_i, \mathbf{g})\right)^2$$

where noisy points are substituted into the curve or surface equation, and the algebraic fit minimizes this substitution error, which yields a fast, non-iterative approach. Let a linearization (lifting) of 2D observations be

$$\mathbf{z}^\top = \begin{bmatrix} x^2 & xy & y^2 & x & y & 1 \end{bmatrix} \tag{1}$$

and a linearization (lifting) of 3D observations be

$$\mathbf{z}^\top = \begin{bmatrix} x^2 & y^2 & z^2 & xy & xz & yz & x & y & z & 1 \end{bmatrix} \tag{2}$$

such that the sample covariance matrix is

$$\mathbf{D} = \frac{1}{N}\sum_i \left(\mathbf{z}_i - \frac{1}{N}\sum_j \mathbf{z}_j\right)\left(\mathbf{z}_i - \frac{1}{N}\sum_j \mathbf{z}_j\right)^\top.$$

The least squares solution then solves the eigenvalue problem

$$\mathbf{Dg} = \lambda\mathbf{g}$$

for the minimum eigenvalue λ.

However, in many cases we are interested in fitting a quadratic curve or surface subject to constraints, i.e. estimating the parameter $\mathbf{g}$ such that it represents a particular class of quadratic curves or surfaces such as ellipses or ellipsoids. Direct least squares fitting of ellipses [FPF99, HF98] uses a constraint (normalization matrix)

$$\mathbf{Q} = \mathrm{diag}\left(\begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix}, \mathbf{0}_{3\times 3}\right)$$

which represents the constraint

$$\mathbf{g}^\top\mathbf{Qg} > 0$$

and solves the eigenvalue problem

$$\mathbf{Dg} = \lambda\mathbf{Qg}$$

for the only positive eigenvalue λ.

Unfortunately, simple as it is, directly minimizing algebraic error is statistically inaccurate. An approach to maintain the simplicity of least squares methods yet combat their statistical inaccuracy is to use a noise cancellation scheme, and remove the distortion effects of noise (as much as possible) before attempting a least squares fit. As apparent from the observation vectors (1) and (2), the effect of noise on various components may nonlinear (such as for x^2),

which must be taken into account in devising such estimation schemes. This leads to a matrix polynomial

$$\mathbf{D} - \mathbf{C}(\mu)$$

where μ denotes noise magnitude and $\mathbf{C}(\mu)$ is a matrix estimated from noise observations. The corrected matrix $\mathbf{D}-\mathbf{C}(\mu)$ is then subject to constrained minimization where the quadratic curve or surface class (e.g. ellipse or ellipsoid) is taken into account.

The proposed estimation approach comprises of the following consecutive steps:

1. noise cancellation

2. constrained quadratic least squares fitting

This approach guarantees consistent estimation (step 1) while ensuring robustness (step 2), the algorithm, unlike others, enforces constraints even if data would suggest a shape other than that to be fitted.

Thesis 1. Fitting ellipses, parabolas, hyperbolas and ellipsoids with noise cancellation

I have proposed an errors-in-variables parameter estimation method fitting quadratic curves and surfaces subject to constraints. The method incorporates a noise cancellation step in existing constrained quadratic least squares fitting algorithms. The noise cancellation step can be written as a quadratic eigenvalue problem on symmetric matrices, where the eigenvalue problem yields a data covariance matrix with noise distortions being accounted for, and may be formulated without the statistically invariant terms in the data covariance matrix, leading to a simpler expression. Operating on a noise-compensated data covariance matrix, direct ellipse-specific, parabola-specific, hyperbola-specific and ellipsoid-specific least squares fitting exhibit substantially improved accuracy compared to their original formulation without noise cancellation.

Given a set of data $\mathbf{z}_i$, a sample covariance matrix $\mathbf{D}$ of these data $\mathbf{z}_i$, an estimated noise covariance matrix $\mathbf{C}(\mu)$ can be developed in a series of simple steps. In particular, $\mathbf{C}(\mu)$ for estimating parameters of a quadratic curve or surface will take the form $\mathbf{C}(\mu) = \mu^2\mathbf{C}_2 + \mu\mathbf{C}_1$ where $\mathbf{C}_2$ is a matrix corresponding to the quadratic part whose entries depend on noise variances σ_x^2 and σ_y^2 (and σ_z^2 in three dimensions), while those of $\mathbf{C}_1$ corresponding to the linear part depend on both the noise variances as well as terms approximated from finite noisy data samples. In two dimensions, this means $\mathbb{E}\left(x^2\right)$, $\mathbb{E}\left(y^2\right)$, $\mathbb{E}\left(xy\right)$, $\mathbb{E}\left(x\right)$ and $\mathbb{E}\left(y\right)$. Since knowledge of noise magnitude alone does not give additional insight by itself, we write $\sigma_x^2 = \mu\bar{\sigma}_x^2$, $\sigma_y^2 = \mu\bar{\sigma}_y^2$ with $\bar{\sigma}_x^2 + \bar{\sigma}_y^2 = 1$ where $\bar{\sigma}_x^2$ and $\bar{\sigma}_y^2$ represent (known) noise "directionality" (relative distribution of noise w.r.t. variables) and μ represents (unknown) noise magnitude. As a result, we arrive at a quadratic eigenvalue problem (QEP)

$$\mathbf{\Psi}(\mu)\mathbf{g} = \left(\mathbf{D} - \mu\mathbf{C}_1 - \mu^2\mathbf{C}_2\right)\mathbf{g} = \mathbf{0}.$$

One way to solve the QEP is to apply linearization, thereby eliminating the polynomial dependence on μ at the expense of increasing the size of coefficient matrices, which is analogous

to companion matrices constructed from polynomials where the eigenvector of the companion matrix yields the roots of the polynomial. In particular, transformations that preserve symmetry are especially favored for their numerical stability. A well-known result [TM01] for linearizing the QEP

$$\mathbf{D} - \mu\mathbf{C}_1 - \mu^2\mathbf{C}_2$$

is with the symmetric first companion form

$$\Xi(\lambda) = \Xi_1 - \lambda\Xi_2 = \begin{bmatrix} \mathbf{0} & -\mathbf{D} \\ -\mathbf{D} & \mathbf{C}_1 \end{bmatrix} - \lambda \begin{bmatrix} -\mathbf{D} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_2 \end{bmatrix}$$

which yields a generalized eigenvalue problem with symmetric matrices.

The following algorithm summarizes the cancellation scheme for a general quadratic curve or surface.

Algorithm 1. *Noise cancellation scheme for quadratic curves and surfaces.*

1. *Input:* noisy samples $\mathbf{x}_i$ and the relative noise magnitude vector $\bar{\sigma}^2_{\mathbf{x}}$ for each dimension.
2. Estimate the data covariance matrix $\mathbf{D}$ from noisy samples.
3. Estimate the noise covariance matrix polynomial coefficients $\mathbf{C}_1$ and $\mathbf{C}_2$ from noisy samples.
4. Find the eigenvalue μ that solves $\det\left(\mathbf{D} - \mu\mathbf{C}_1 - \mu^2\mathbf{C}_2\right) = 0$.
5. *Output:* the noise-compensated (singular) matrix $\mathbf{R} = \mathbf{D} - \mu\mathbf{C}_1 - \mu^2\mathbf{C}_2$.

Once the effects of noise have been accounted for, we may write the constrained parameter optimization problem as

$$\begin{aligned} \min_{\mathbf{g}} \quad & \mathbf{g}^\top \mathbf{R}\mathbf{g} \\ \text{s.t.} \quad & \mathbf{g}^\top \mathbf{Q}\mathbf{g} = 1 \end{aligned}$$

which leads to a generalized eigenvalue problem

$$\mathbf{R}\mathbf{g} = \lambda\mathbf{Q}\mathbf{g}.$$

The matrix $\mathbf{Q}$ represents constraints. Let the vectors and matrices be decomposed as

$$\begin{aligned} \mathbf{g}^\top &= \begin{bmatrix} \mathbf{g}_1^\top & \mathbf{g}_2^\top \end{bmatrix}^\top \\ \mathbf{R} &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{R}_2^\top & \mathbf{R}_3 \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned}$$

where $\mathbf{Q}_1$ represents the quadratic constraint that confines the parameters to be e.g. an ellipse or hyperbola such that

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{R}_2^\top & \mathbf{R}_3 \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}.$$

As in [HF98, HOZM08], let

$$\begin{aligned} \mathbf{T} &= -\mathbf{R}_3^{-1}\mathbf{R}_2^\top \\ \mathbf{S} &= \mathbf{R}_1 + \mathbf{R}_2\mathbf{T} \end{aligned}$$

where the inverse $\mathbf{R}_3^{-1}$ always exists unless points are co-linear or co-planar. Thus,

$$\mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{T}\mathbf{g}_1 \end{bmatrix}$$

where we solve the estimation problem by finding the eigenvector $\mathbf{g}_1$ of the reduced scatter matrix $\mathbf{S}$ that satisfies $\mathbf{g}_1^\top \mathbf{Q}_1 \mathbf{g}_1 = 1$, i.e. we find the eigenvector decomposition of

$$\mathbf{S}\mathbf{g}_1 = \lambda \mathbf{Q}_1 \mathbf{g}_1. \tag{3}$$

Ellipse and hyperbola fitting have the constraint

$$q = \mathbf{g}_1^\top \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \mathbf{g}_1 = \mathbf{g}_1^\top \mathbf{Q}_1 \mathbf{g}_1 \tag{4}$$

where $q > 0$ for fitting ellipses and $q < 0$ for fitting hyperbolas. For ellipse-specific fitting [FPF99], the solution $\mathbf{g}_1$ corresponds ideally (in the absence of numerical errors) to the smallest positive eigenvalue λ of the eigenvalue problem (3), whereas for hyperbola-specific fitting [OZM04], the best choice of $\mathbf{g}_1$ among the eigenvectors is found by back-substitution into (4). The generalized eigenvalue problem (3) has therefore three valid solutions with respect to the constraints: one elliptical and two hyperbolic solutions to the conic fitting problem, depending on the eigenvalue λ. There is a symmetry between only one of the hyperbolic solutions and the elliptical solution, which enables us to uniquely identify the correct solutions [OZM04].

Parabola fitting cannot directly utilize the ellipse and hyperbola constraint; fitting a parabola would require $q = 0$ but this would lead to a trivial solution. Instead, as in [HOZM08], we compute a regular eigenvector decomposition of

$$\mathbf{S}\mathbf{g}_1 = \lambda \mathbf{g}_1$$

which has the incorporated implicit constraint $\mathbf{Q} = \mathbf{I}$, confining the estimation to produce an unspecified but quadratic curve, and we enforce the parabola constraint explicitly rather than via the eigenvalue problem. Given the decomposition of $\mathbf{S}$, the solution we seek can be written as a linear combination of eigenvectors

$$\mathbf{g}_1 = \mathbf{v}_1 + s\mathbf{v}_2 + t\mathbf{v}_3$$

where s and t are scalars and we seek $\mathbf{g}_1$ with the minimum norm

$$\begin{aligned}\mathscr{F} &= \mathbf{g}_1^\top \mathbf{S}^\top \mathbf{S}\mathbf{g}_1 \\ &= (\mathbf{v}_1 + s\mathbf{v}_2 + t\mathbf{v}_3)^\top \mathbf{S}^\top \mathbf{S}(\mathbf{v}_1 + s\mathbf{v}_2 + t\mathbf{v}_3) \\ &= \mathbf{v}_1^\top \mathbf{S}^\top \mathbf{S}\mathbf{v}_1 + s^2\mathbf{v}_2^\top \mathbf{S}^\top \mathbf{S}\mathbf{v}_2 + t^2\mathbf{v}_3^\top \mathbf{S}^\top \mathbf{S}\mathbf{v}_3 \\ &= \lambda_1^2 + s^2\lambda_2^2 + t^2\lambda_3^2\end{aligned}$$

in which we have used the orthogonality property of eigenvectors $\mathbf{v}_i^\top \mathbf{v}_j = 0$ for all $i \neq j$. Meanwhile, we express the parabola constraint with the eigenvectors

$$\mathscr{C} = \left(v_{1,2} + sv_{2,2} + tv_{3,2}\right)^2 - 4\left(v_{1,1} + sv_{2,1} + tv_{3,1}\right)\left(v_{1,3} + sv_{2,3} + tv_{3,3}\right) = 0$$

where $v_{i,j}$ is the jth element of the ith eigenvector. This leads us to the Lagrangian

$$\mathscr{L}(s,t,\alpha) = \mathscr{F}(s,t) + \alpha\mathscr{C}(s,t)$$

which yields a fourth-order polynomial in α after equating the derivatives w.r.t. the parameters with zero. Solving the polynomial for α and back-substituting it to get s and t, we eventually obtain $\mathbf{g}_1$ and thus the parameter vector $\mathbf{g}$.

For ellipsoid-specific fitting [LG04], we choose the constraint in (3) as

$$\mathbf{Q}_1 = \operatorname{diag}\left(\frac{1}{2}\begin{bmatrix} 0 & k & k \\ k & 0 & k \\ k & k & 0 \end{bmatrix} - \begin{bmatrix} k & k & k \\ k & k & k \\ k & k & k \end{bmatrix}, -\frac{1}{4}\begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{bmatrix}\right)$$

where $k \geqq 4$. Unlike the constraint matrix for ellipses and hyperbolas, which express simple constraints, the ellipsoid-specific constraint depends on a parameter k. Let

$$\mathbf{g}^\top = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 & g_8 & g_9 & g_{10} \end{bmatrix}$$

which pair with components of

$$\mathbf{z}^\top = \begin{bmatrix} x^2 & y^2 & z^2 & xy & xz & yz & x & y & z & 1 \end{bmatrix}$$

and

$$\begin{aligned} I &= g_1 + g_2 + g_3 \\ J &= g_1g_2 + g_2g_3 + g_1g_3 - \frac{1}{4}g_4^2 - \frac{1}{4}g_5^2 - \frac{1}{4}g_6^2. \end{aligned}$$

When $4J - I^2 > 0$, the parameter vector $\mathbf{g}$ represents an ellipsoid. On the other hand, when the short radius of an ellipsoid is at least half of its major radius, we have $4J - I^2 > 0$ (corresponding to $k = 4$). For any ellipsoid, there always exists a k such that $kJ - I^2 > 0$. It can be shown [LG04] that when the short radius of an ellipsoid is at least $1/\sqrt{k}$ multiple of its major radius, then we will have $kJ - I^2 > 0$. With a bisection method, starting from a large k, estimating

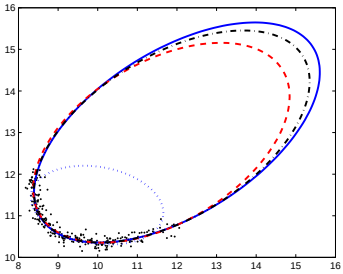

Figure 1: Comparing the accuracy of direct ellipse fit (dotted line), hyper-accurate ellipse fit (dashed line) and the estimation method with noise cancellation (dash dot line). Data points (shown as dots) are contaminated with noise.

parameters $\mathbf{g}_1$ given $\mathbf{Q}_1(k)$, and iterating until $\mathbf{g}_1$ identifies an ellipsoid, we may guarantee ellipsoid-specific fitting.

Figure 1 compares three estimation methods: a least-squares-based approach called direct ellipse fit [FPF99] (shown with dotted line), hyper-accurate ellipse fit [KR11] (shown with dashed line) and the estimation based on the noise compensation approach (shown with dash dot line). 250 original data points sampled evenly along a section of an ellipse (continuous line) are contaminated with Gaussian noise of $\sigma_{\mathbf{x}} = 0.1$ (plotted as black dots). The original ellipse that has generated the data points has center $(12, 13)$, semi-axes lengths of 4 and 2, and angle of tilt $\frac{\pi}{6}$. It can be seen how the estimator we use can achieve an accuracy strikingly similar to that of hyper-accurate ellipse fit (an equation error method that completely cancels second-order error), while outperforming direct ellipse fit, which has a substantial low-eccentricity bias. Unlike our method, hyper-accurate ellipse fit requires the computation of several matrix pseudo-inverses, and despite what its name suggests, is not ellipse-specific.

3.2 Structure discovery by fitting low-order manifolds

Related publications: [1, 2, 6, 7, 12, 13, 14]

With the increasing availability of data from various sources and modalities, there has been escalated interest in acquiring, compressing, storing, transmitting and processing massive amounts of data. Most of these advances rely on the observation that even though data sets are often high-dimensional, their intrinsic dimension is often (much) smaller. For example, we may have a data set of three dimensional data points, yet we are interested in discovering lines, planes or ellipsoids in the data set. Conventional techniques, such as singular value decomposition, seek a low-dimensional representation of the higher dimensional space, a compact model that explains the data points using a limited set of base vectors, assuming all data points originate from a single low-dimensional space. In most situations, however,

data points are seldom drawn from a single low-dimensional space. Rather, they belong to multiple subspaces, and modeling involves both identifying subspace membership as well as subspace fitting. Subspace clustering, as the problem is typically referred to, is a significantly more difficult problem with a number of challenges:

1. Data segmentation and model parameter estimation are strongly coupled. Were either the segmentation or the estimation known, the other could be easily derived using conventional estimation methods or projection. Simultaneous segmentation and estimation is, however, more demanding.

2. Uneven distribution of data points and the relative position of subspaces with respect to one another, including subspace intersections, seeks for more involved algorithms.

3. Model selection, i.e. choosing the right dimension for each of the subspaces, can be difficult if the subspaces admit more complex structures.

4. Noise corruption introduces more uncertainty to model accuracy compared to the case of a single subspace.

An intuitive way to address the problem of subspace discovery is to employ iterative refinement. Given an initial segmentation, a subspace can be fit to each group using classical techniques such as singular value decomposition. Then, given a model for each subspace, each data point is assigned to its closest subspace. By alternating these two steps until convergence, an estimate of the subspaces and segmentation may be discovered.

Despite their simplicity, iterative algorithms are sensitive to initialization. If data points are randomly assigned, several restarts are required before a near-optimum solution is found. In order to reduce the number of restarts, algorithms that produce feasible results without any preliminary input are required. Spectral Local Best-fit Flats (SLBF) [ZSWL10] is based on the observation that points and their nearest neighbors often belong to the same subspace, which provides estimates for the parameters of the subspace. A distance between a point and the locally estimated subspace around the other can serve as an affinity between two data points. Extending the principles of SLBF to the nonlinear case, and fitting manifolds described by polynomial functions to data rather than subspaces, and employing appropriate projections, leads to an effective manifold clustering algorithm.

Thesis 2.1. Iterative algorithm for manifold clustering

I have generalized linear grouping algorithms [BM00, Tse00, AM04, AWZZ06] for finding linear patterns in a data set to a grouping algorithm fitting quadratic curves and surfaces. Alternating between an update (parameter estimation) and an assignment (data mapping) step, the method can discover a structural decomposition of data where members of each group are related by a low-order (linear or quadratic) implicit polynomial function.

The outline of the proposed iterative grouping algorithm resembles the iterative algorithm of the standard k-means procedure. The primary difference lies in the use of parameters instead of mean values, and simple point-to-point distance (between cluster center and data points) replaced with data point projection. As with the standard k-means algorithm, the choice of initial groups affects convergence to an optimal solution.

Algorithm 2. *Iterative grouping algorithm fitting polynomial functions.*

1. Initialization. Choose k randomly chosen initial points $\mathbf{x}_i$ with $i = 1, \ldots, k$, and for each data point $\mathbf{x}_i$
   (a) start with an initial neighborhood $\mathcal{N}(\mathbf{x}_i)$ around $\mathbf{x}_i$
   (b) estimate the parameters $\boldsymbol{\theta}_{i,(0)}$ that best capture points in $\mathcal{N}(\mathbf{x}_i)$
   (c) enlarge $\mathcal{N}(\mathbf{x}_i)$ by adding nearest-neighbor points
   (d) compute new estimates $\boldsymbol{\theta}_{i,(n)}$ and compare them to the estimates $\boldsymbol{\theta}_{i,(n-1)}$ obtained in the previous iteration
   (e) repeat until the neighborhood $\mathcal{N}(\mathbf{x}_i)$ cannot be enlarged without worsening the accuracy of $\boldsymbol{\theta}_{i,(n)}$
   (f) let $\boldsymbol{\theta}_i$ be the best $\boldsymbol{\theta}_{i,(n)}$ that belongs to the optimum $\mathcal{N}(\mathbf{x}_i)$ around $\mathbf{x}_i$.
2. Initial grouping.
   (a) project each data point $\mathbf{x}_j$ with $j = 1, \ldots, N$ to all candidate $\boldsymbol{\theta}_i$ with $i = 1, \ldots, k$
   (b) form initial groups $\mathcal{S}_i$ with $i = 1, \ldots, k$ such that the distance is minimized.
3. Alternating optimization.
   (a) Update step. Each group of data points $\mathcal{S}_i$ is used to estimate new surface parameters $\boldsymbol{\theta}_i$.
   (b) Assignment step. Each data point $\mathbf{x}_j$ is assigned to the surface $\boldsymbol{\theta}_i$ it lies in the vicinity of.
4. Finalization. Each data point is assigned to a single shape $\boldsymbol{\theta}_i$ it lies closest to.
5. Re-sampling. Repeat the algorithm with different randomly chosen initial locations.

Thesis 2.2. Non-iterative algorithm for manifold clustering

I have proposed a clustering method fitting implicit polynomial functions that finds an initial segmentation without a preliminary assignment of data points. The clustering problem is modeled with a complete directed weighted graph where the nodes are data points, and the edge weights a_{ij} are an affinity measure reflecting the distance between a data point i and the curve or surface estimated from points in the neighborhood of the other point j. A best weighted cut algorithm is applied to split the graph into k components, and thereby produce an asymmetric spectral clustering of the data set with the given distance measure. The proposed method surpasses other manifold clustering methods that do not explicitly incorporate function fitting to data in clusters.

Grouping with spectral clustering remedies the issue of wrongly chosen initial locations. Spectral clustering eliminates the need for re-sampling and provides a clustering that is already close to an optimum solution.

Algorithm 3. *Grouping algorithm fitting polynomial functions, driven by spectral clustering*

1. Initialization. For each data point $\mathbf{x}_i$ with $i = 1, \ldots, N$
   (a) start with an initial neighborhood $\mathcal{N}(\mathbf{x}_i)$ around $\mathbf{x}_i$
   (b) estimate the parameters $\boldsymbol{\theta}_{i,(0)}$ that best capture points in $\mathcal{N}(\mathbf{x}_i)$
   (c) enlarge $\mathcal{N}(\mathbf{x}_i)$ by adding nearest-neighbor points
   (d) compute new estimates $\boldsymbol{\theta}_{i,(n)}$ and compare them to the estimates $\boldsymbol{\theta}_{i,(n-1)}$ obtained in the previous iteration
   (e) repeat until the neighborhood $\mathcal{N}(\mathbf{x}_i)$ cannot be enlarged without worsening the accuracy of $\boldsymbol{\theta}_{i,(n)}$
   (f) let $\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_i)}$ be the best $\boldsymbol{\theta}_{i,(n)}$ that belongs to the optimum $\mathcal{N}(\mathbf{x}_i)$ around $\mathbf{x}_i$.
2. Compute asymmetric distances. For each pair of data points $\mathbf{x}_i$ and $\mathbf{x}_j$, obtain their asymmetric distances
   (a) project $\mathbf{x}_i$ to the manifold parametrized by $\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_j)}$ and calculate a_{ij}
   (b) project $\mathbf{x}_j$ to the manifold parametrized by $\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_i)}$ and calculate a_{ji}.
3. Spectral clustering.
   (a) build an affinity matrix from asymmetric distances a_{ij}
   (b) assign data points to groups based on the affinity matrix eigenvectors.

Figure 2 illustrates the effectiveness of the proposed algorithm, comparing it to other manifold clustering algorithms that use locality and connectivity information. The proposed algorithm significantly outperforms the K-manifolds algorithm [SP05] and Kernel Spectral Curvature Clustering [CAL09]. The fact that the proposed algorithm uses structural information and estimates parametric curves and surfaces significantly improves its explanation power and in turn its fitting capabilities.

The key idea of the non-iterative algorithm is constructing an asymmetric distance matrix $\mathbf{A}$ whose entry a_{ij} represents the distance of data point $\mathbf{x}_i$ from its foot point obtained by projecting $\mathbf{x}_i$ onto the surface around $\mathbf{x}_j$, defined by parameters $\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_j)}$.

Definition 4. Asymmetric distance matrix. Let

$$a_{ij} = d\left(\mathbf{x}_i, \mathbf{p}\left(\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_j)}, \mathbf{x}_i\right)\right)$$

where d denotes Euclidean distance and $\mathbf{p}\left(\boldsymbol{\theta}, \mathbf{x}_i\right)$ denotes the projection of the point $\mathbf{x}_i$ to the curve or surface defined by $\boldsymbol{\theta}$, and $\boldsymbol{\theta}_{\mathcal{N}(\mathbf{x}_j)}$ are the parameters of the curve or surface estimated

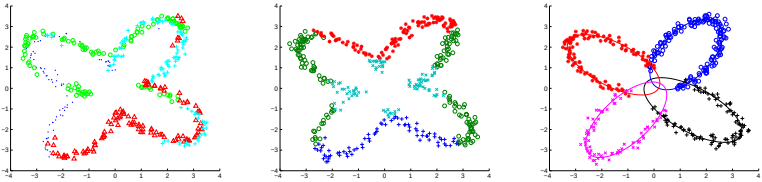

Figure 2: Comparison of manifold clustering algorithms; left: K-manifolds algorithms, center: Kernel Spectral Curvature Clustering, right: the proposed algorithm.

from the set of points $\mathcal{N}(\mathbf{x}_j)$, which is the local neighborhood of $\mathbf{x}_j$. Unlike the standard notion of distance, we have in general $a_{ij} \neq a_{ji}$. The asymmetric distance matrix is defined as

$$\mathbf{A} = \left[a_{ij}\right].$$

A non-iterative approach to split data into k groups is spectral clustering. Most spectral clustering methods, however, assume a symmetric distance matrix. While the standard k-means algorithm uses point-to-point distances, which are inherently symmetric, our polynomial grouping algorithm employs projections onto manifolds, which is not typically symmetric. Asymmetric spectral clustering, an algorithm to find a best cut in a weighted graph [MP07], is an approach to find a decomposition without prematurely forcing symmetry to the model.

Algorithm 5. *Asymmetric spectral clustering.*

Let $\mathbf{S}$ be a(n asymmetric) matrix initialized with entries s_{ij} between 0 and 1, not at all similar and most similar, respectively. The steps of the algorithm [MP07] are as follows:

1. Normalize the matrix $\mathbf{S}$. Let
$$d_i = \sqrt{\frac{1}{\sum_j [\mathbf{S}]_{ij}}}$$
and $\mathbf{D} = \operatorname{diag}\left(d_i\right)$ such that
$$\mathbf{H} = \mathbf{I} - \frac{1}{2}\mathbf{D}(\mathbf{S} + \mathbf{S}^\top)\mathbf{D}.$$

2. Compute the smallest eigenvectors of the scatter matrix $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U}$ is a matrix of eigenvectors and a $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues.

3. Let $\mathbf{U}_k$ be a matrix of the bottom k smallest eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots$ arranged in a matrix.

4. Normalize the rows of $\mathbf{U}_k$ to unit length.

5. Perform a k-means clustering on normalized $\mathbf{U}_k$.

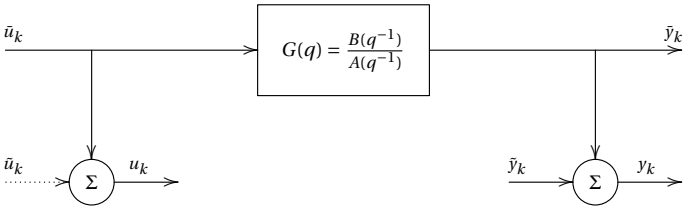

Figure 3: A linear dynamic discrete-time errors-in-variables system.

The algorithm allows us to start with an asymmetric distance matrix such as

$$\mathbf{S} = \exp(-\mathbf{A})$$

where the operator $\exp(\bullet)$ is to be understood element-wise.

3.3 Nonlinear extension to the generalized Koopmans–Levin method

Related publications: [9, 15, 3, 18, 19, 20, 21, 24, 25, 28]

Single-input single-output (SISO) discrete-time (DT) dynamic linear systems are at the core of system identification and control. Figure 3 shows how such an errors-in-variables system compares to a classical setup in system identification. In the classical case, only system output is measured with noise, system input can be observed noise-free (i.e. no noise contribution by the dashed line), which is a well-understood problem. The case is more subtle when both the system input and output is contaminated with noise. While there has been more substantial research on the parameter estimation of dynamic linear errors-in-variables systems, nonlinear systems are not nearly as well discussed. The dissertation proposes a new method that extends the generalized Koopmans–Levin method for linear system identification method to systems where model parameters are polynomial functions of input and output data. This means that a function such as

$$f\left(\bar{u}_{k-1}, \bar{u}_{k-1}^2, \bar{y}_k, \bar{y}_{k-1}, \bar{y}_{k-1}^2, \ldots\right)$$

takes the place of $G(q)$ in Figure 3.

Thesis 3. Iterative method to estimate parameters of dynamic polynomial systems

I have proposed the polynomial extension to the generalized Koopmans–Levin (PGKL) estimation method for dynamic systems by unifying the nonlinear Koopmans (NK) method for static polynomial systems with the generalized Koopmans–Levin (GKL) method for linear dynamic systems. The algorithm is formulated as an iterative procedure. I have demonstrated how a

generalized eigenvalue problem as used with the iterative approach of the GKL *method extends to a polynomial eigenvalue problem if the errors can be modeled as Gaussian noise. I have utilized a symmetric linearization of the polynomial eigenvalue problem to preserve the symmetry present in the original problem and maintain numerical robustness. The* PGKL *estimation method assumes a known relative distribution of noise between system input and output. I have shown how a noise covariance matching approach can estimate the relative noise distribution if this parameter is unknown.*

Let $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^n$ be a linearization (lifting) function applied to each time-shifted set of terms in the data window $\bar{\mathbf{x}}_{m,k}$ of dimension $(m+1)\cdot d$ corresponding to the discrete time k, producing a set of terms arranged in a vector of dimension $(m+1)\cdot n$. Let

$$\mathbf{G}_q^{\theta_p} = \begin{bmatrix} p_0 & 0 & 0 & \dots & 0 & 0 \\ p_1 & p_0 & 0 & \dots & 0 & 0 \\ p_2 & p_1 & p_0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ p_m & p_{m-1} & p_{m-2} & \dots & 0 & 0 \\ 0 & p_m & p_{m-1} & \dots & 0 & 0 \\ 0 & 0 & p_m & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & p_m & p_{m-1} \\ 0 & 0 & 0 & \dots & 0 & p_m \end{bmatrix}_{q,q-m}$$

where each $p_\bullet$ corresponds to a time-shifted element in a block of the same polynomial term, arranged in decreasing order of time, such that the model parameters are

$$\mathbf{G}_q = \begin{bmatrix} \mathbf{G}_q^{\theta_1} \\ \mathbf{G}_q^{\theta_2} \\ \vdots \end{bmatrix}.$$

Let

$$\mathbf{f}(\bar{\mathbf{x}}_{m,k}) = \tilde{\mathbf{z}} = \begin{bmatrix} \tilde{z}_{0,k} & \tilde{z}_{1,k} & \dots & \tilde{z}_{m,k} \end{bmatrix}$$

where $\tilde{z}_{r,k} = z_{r,k} - \bar{z}_{r,k}$ in which $\bar{z}_{r,k}$ is the unobservable true value of the component. Let

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\left(\tilde{\mathbf{z}}_i - \frac{1}{N}\sum_{j=1}^{N}\tilde{\mathbf{z}}_j\right)\left(\tilde{\mathbf{z}}_i - \frac{1}{N}\sum_{j=1}^{N}\tilde{\mathbf{z}}_j\right)^\top.$$

The linearized noise covariance matrix (and its approximation from finite N data samples) is then $\mathbf{C}_q = \mu\mathbf{C}\otimes\mathbf{I}_q$ where μ represents noise magnitude. Then the objective function for the polynomial extension to the generalized Koopmans–Levin method assuming Gaussian noises is

$$\arg\min_{\boldsymbol{\theta},\mu} \frac{1}{2}\operatorname{trace}\left\{\left(\mathbf{G}_q^\top\mathbf{C}_q(\mu)\mathbf{G}_q\right)^{-1}\left(\mathbf{G}_q^\top\mathbf{D}_q\mathbf{G}_q\right)\right\}.$$

The objective function may be approximated with a simple iterative scheme, which exhibits favorable convergence properties. Let

$$\boldsymbol{\theta}_{(k+1)},\, \mu_{(k+1)} = \arg\min_{\boldsymbol{\theta},\mu} \frac{\operatorname{trace}\left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\left(\mathbf{G}_{q}^{\top}\mathbf{D}_{q}\mathbf{G}_{q}\right)}{\operatorname{trace}\left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\left(\mathbf{G}_{q}^{\top}\mathbf{C}_{q}\mathbf{G}_{q}\right)}$$

where $\mathbf{G}_q = \mathbf{G}_q(\boldsymbol{\theta})$, $\mathbf{G}_{q,(k)} = \mathbf{G}_q(\boldsymbol{\theta}_{(k)})$, $\mathbf{C}_q = \mathbf{C}_q(\mu)$ and $\mathbf{C}_{q,(k)} = \mathbf{C}_q(\mu_{(k)})$ is a means of computing the successor estimates $\boldsymbol{\theta}_{(k+1)}$ and $\mu_{(k+1)}$ given $\boldsymbol{\theta}_{(k)}$ and $\mu_{(k)}$. The iterative scheme

$$\arg\min_{\boldsymbol{\theta},\mu} \frac{\boldsymbol{\theta}^{\top}\mathbf{T}^{\top}\left(\left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\otimes\mathbf{D}_{q}\right)\mathbf{T}\boldsymbol{\theta}}{\boldsymbol{\theta}^{\top}\mathbf{T}^{\top}\left(\left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\otimes\mathbf{C}_{q}(\mu)\right)\mathbf{T}\boldsymbol{\theta}}$$

where $\mathbf{T}$ is a sparse matrix of zeros and ones chosen such that $\operatorname{vec}(\mathbf{G}_q) = \mathbf{T}\boldsymbol{\theta}$ leads to a series of polynomial eigenvector problems (PEPs)

$$\boldsymbol{\Psi}(\mu)\boldsymbol{\theta} = \left(\mathbf{Q} - \mathbf{R}(\mu)\right)\boldsymbol{\theta} = \mathbf{0}$$

with matrix coefficients

$$\begin{aligned}
\boldsymbol{\Psi}(\mu) &= \mathbf{T}^{\top}\left(\mathbf{Q} - \mu\mathbf{R}_1 - \mu^2\mathbf{R}_2 - \ldots - \mu^p\mathbf{R}_p\right)\mathbf{T} \\
\mathbf{Q} &= \left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\otimes\mathbf{D}_q \\
\mathbf{R}_j &= \left(\mathbf{G}_{q,(k)}^{\top}\mathbf{C}_{q,(k)}\mathbf{G}_{q,(k)}\right)^{-1}\otimes\mathbf{C}_q^{(j)}.
\end{aligned}$$

One way to solve the PEP is to apply linearization, thereby eliminating the polynomial dependence on μ at the expense of increasing the size of coefficient matrices, which is analogous to companion matrices constructed from polynomials where the eigenvector of the companion matrix yields the roots of the polynomial. In particular, transformations that preserve symmetry are especially favored for their numerical stability. One such symmetry-preserving linearization [AV04, AV06] of the above problem is

$$\boldsymbol{\Xi}(\mu) = \boldsymbol{\Xi}_1 - \mu\boldsymbol{\Xi}_2$$

which expands for even p as

$$\begin{aligned}
\boldsymbol{\Xi}_1 &= \operatorname{diag}\left\{\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{I} & \mathbf{R}_1\end{bmatrix}, \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{I} & \mathbf{R}_3\end{bmatrix}, \ldots, \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{I} & \mathbf{R}_{p-1}\end{bmatrix}\right\} \\
\boldsymbol{\Xi}_2 &= \operatorname{diag}\left\{\mathbf{R}_0^{-1}, \begin{bmatrix}-\mathbf{R}_2 & \mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}, \ldots, \begin{bmatrix}-\mathbf{R}_{p-2} & \mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}, -\mathbf{R}_p\right\}
\end{aligned}$$

and for odd p as

$$\begin{aligned}\Xi_1 &= \operatorname{diag}\left\{\mathbf{R}_0, \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{R}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{R}_4 \end{bmatrix}, \ldots\right\} \\ \Xi_2 &= \operatorname{diag}\left\{\begin{bmatrix} -\mathbf{R}_1 & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \begin{bmatrix} -\mathbf{R}_3 & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \ldots, -\mathbf{R}_p\right\}\end{aligned}$$

where the operator diag aligns its arguments to bring forth a block diagonal matrix. The resulting problem can then be solved as a generalized eigenvalue problem, which is the linearized equivalent of solving the original PEP. As the linearized problem has eigenvectors $\mathbf{w}$ of dimension mp rather than m, the true polynomial eigenvector that belongs to the eigenvalue μ becomes the column $\mathbf{v}$ of $\operatorname{vec}\mathbf{V} = \mathbf{w}$ of the linearized eigenvector $\Xi_1\mathbf{w} = \mu\Xi_2\mathbf{w}$ that gives the smallest normalized residual.

Given the estimates $\hat{\boldsymbol{\theta}}$ and $\hat{\mu}$ for a polynomial dynamic errors-in-variables system with a relative distribution of noise φ between input and output (i.e. an input/output noise ratio), we may introduce a measure of dissimilarity between the noise present in the data sample and our noise model $\mathbf{C}_q = \mu\mathbf{C}(\varphi) \otimes \mathbf{I}_q$ with our assumption of φ. Let $d_F(\mathbf{A}, \mathbf{B})$ be the Frobenius norm of $\mathbf{A} - \mathbf{B}$ such that

$$d_F(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|$$

and $d_{IS}(\mathbf{A}, \mathbf{B})$ be the Itakura–Saito divergence between matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$d_{IS}(\mathbf{A}, \mathbf{B}) = \operatorname{trace}\left(\mathbf{A}^{-1}\mathbf{B}\right) - \log\det\left(\mathbf{A}^{-1}\mathbf{B}\right).$$

Then, the matrix divergence

$$d\left(\mathbf{G}_q^\top(\hat{\boldsymbol{\theta}})\mathbf{D}_q\mathbf{G}_q(\hat{\boldsymbol{\theta}}), \mathbf{G}_q^\top(\hat{\boldsymbol{\theta}})\mathbf{C}_q(\hat{\mu})\mathbf{G}_q(\hat{\boldsymbol{\theta}})\right)$$

measures how close our noise model is to the noise in the observed data. A covariance matching scheme over φ can then give an estimate for the relative noise distribution

$$\varphi = \arg\min_{\varphi} d\left(\mathbf{G}_q^\top(\hat{\boldsymbol{\theta}})\mathbf{D}_q\mathbf{G}_q(\hat{\boldsymbol{\theta}}), \mathbf{G}_q^\top(\hat{\boldsymbol{\theta}})\mathbf{C}_q(\hat{\mu})\mathbf{G}_q(\hat{\boldsymbol{\theta}})\right)$$

where the estimates $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\varphi)$ and $\hat{\mu} = \hat{\mu}(\varphi)$ are computed for a particular φ by the PGKL method.

4 Applications

The errors-in-variables approach is frequently encountered in several fields where the goal is reconstructing a model from a noisy point cloud. Notably, in the context of computer vision and pattern recognition, an important sub-problem is parametric curve and surface fitting. The proposed methods for unconstrained and constrained fitting are directly applicable to estimating parameters of quadratic curves and surfaces at modest computational cost. The

estimates obtained with the proposed methods are close to those obtained with the maximum likelihood approach. On the other hand, maximum likelihood estimation entails iterations and each iteration involves projection to a nonlinear curve or surface, which are costly operations. As demonstrated by MatLab source code that augments this thesis, the proposed methods, in contrast, are inexpensive to implement.

Taking a step further, a wide variety of problems from computer vision, such as motion segmentation and face clustering on polygonal surfaces, involve data modeling by multiple subspaces. In tracking-based motion segmentation, feature points are clustered according to the different moving objects. Under the affine camera model, the vectors of feature point coordinates corresponding to a moving rigid object lie on an affine subspace; clustering different moving objects is equivalent to clustering different affine subspaces. Similarly, in face clustering, the set of all images of a Lambertian object under a variety of lighting conditions form a convex polyhedral cone in the image space, and this cone can be accurately approximated by a low-dimensional linear subspace. The proposed structure discovery methods, which are data modeling methods by nonlinear manifolds, can be applied in these contexts.

Finally, the system identification method for polynomial dynamic systems is directly applicable to single-input single-output systems where we seek a low-order polynomial representation of a system under study. A polynomial approximation being a natural way to deal with a nonlinear system, the proposed method can be employed in a wide range of contexts.

References

[AM04] Pankaj K. Agarwal and Nabil H. Mustafa. k-means projective clustering. In *Proc. of ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2004.

[AV04] Efstathios N. Antoniou and Stavros Vologiannidis. A new family of companion forms of polynomial matrices. *Electronic Journal of Linear Algebra*, 11:78–87, 2004.

[AV06] Efstathios N. Antoniou and Stavros Vologiannidis. Linearizations of polynomial matrices with symmetries and their applications. *Electronic Journal of Linear Algebra*, 15:107–114, February 2006.

[AWZZ06] Stefan Van Aelst, Xiaogang (Steven) Wang, Ruben H. Zamar, and Rong Zhu. Linear grouping using orthogonal regression. *Computational Statistics and Data Analysis*, 50(5):1287–1312, March 2006.

[BM00] Paul S. Bradley and Olvi L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[C+05] Wojciech Chojnacki et al. FNS, CFNS and HEIV: A unifying approach. *Journal of Mathematical Imaging and Vision*, 23:175–183, 2005.

[CAL09] Guangliang Chen, Stefan Atev, and Gilad Lerman. Kernel spectral curvature clustering (KSCC). In *Proc. of IEEE International Conference on Computer Vision (ICCV 2009)*, pages 765–772, Kyoto, Japan, September 2009.

[CJ93] Pramod N. Chivate and Andrei G. Jablokow. Solid-model generation from measured point data. *Computer-Aided Design*, 25(9):587–600, 1993.

[CM11] Nikolai Chernov and Hui Ma. Least squares fitting of quadratic curves and surfaces. *Computer Vision*, pages 285–302, 2011.

[FPF99] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least squares fitting of ellipses. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 21:476–480, 1999.

[HF98] Radim Halíř and Jan Flusser. Numerically stable direct least squares fitting of ellipses. In Vaclav Skala, editor, *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, pages 125–132, 1998.

[HOZM08] Matthew Harker, Paul O'Leary, and Paul Zsombor-Murray. Direct type-specific conic fitting and eigenvalue bias correction. *Image and Vision Computing*, 26:372–381, 2008.

[KR11] Kenichi Kanatani and Prasanna Rangarajan. Hyper least squares fitting of circles and ellipses. *Computational Statistics and Data Analysis*, 55:2197–2208, 2011.

[LG04] Qingde Li and John G. Griffiths. Least squares ellipsoid specific fitting. In *Proceedings of the Geometric Modeling and Processing*, 2004.

[MM00] Bogdan Matei and Peter Meer. A general method for errors-in-variables problems in computer vision. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18–25. IEEE Computer Society Press, 2000.

[MP07] Marina Meilă and William Pentney. Clustering by weighted cuts in directed graphs. In *Proc. of the 2007 SIAM International Conference on Data Mining*, 2007.

[OZM04] Paul O'Leary and Paul J. Zsombor-Murray. Direct and specific least-square fitting of hyperbolæ and ellipses. *Journal of Electronic Imaging*, 13(3):492–503, 2004.

[SP05] Richard Souvenir and Robert Pless. Manifold clustering. In *Proceedings of the 10th International Conference on Computer Vision*, pages 648–653, 2005.

[TM01] Françoise Tisseur and Karl Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001.

[Tse00] Paul Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

[Vaj05] István Vajk. Identification methods in a unified framework. *Automatica*, 41(8):1385–1393, 2005.

[VH03] István Vajk and Jenő Hetthéssy. Identification of nonlinear errors-in-variables models. *Automatica*, 39:2099–2107, 2003.

[ZSWL10] Teng Zhang, Arthur Szlam, Yi Grace Wang, and Gilad M. Lerman. Hybrid linear modeling via local best-fit flats. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1927–1934, 2010.

Journal articles

[1] Levente Hunyadi and István Vajk. Modeling by fitting a union of polynomial functions to data. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(2), 2013. IF (2011): 0.624.

[2] Levente Hunyadi and István Vajk. Reconstructing a model of implicit shapes from an unorganized point set. *Scientific Bulletin of "Politehnica" University of Timişoara, Transactions on Automatic Control and Computer Science (Buletinul Stiintific al Universitatii "Politehnica" din Timişoara, Romania, Seria Automatica si Calculatoare)*, 56(70):57–64, 2011.

[3] Levente Hunyadi and István Vajk. Identifying dynamic systems with polynomial nonlinearities in the errors-in-variables context. *WSEAS Transactions on Systems*, 8(7):793–802, July 2009.

[4] Levente Hunyadi and István Vajk. An errors-in-variables parameter estimation method with observation separation. *Scientific Bulletin of "Politehnica" University of Timişoara, Transactions on Automatic Control and Computer Science*, 54(68)(2):93–100, 2009.

[5] Levente Hunyadi and István Vajk. An identification approach to dynamic errors-in-variables systems with a preliminary clustering of observations. *Periodica Polytechnica Electrical Engineering*, 52(3-4):127–135, 2008.

Publications in conference proceedings

[6] Levente Hunyadi and István Vajk. Fitting a model to noisy data using low-order implicit curves and surfaces. In *Proc. of 2nd Eastern European Regional Conference on the Engineering of Computer Based Systems*, pages 106–114, Bratislava, Slovakia, September 2011.

[7] Levente Hunyadi and István Vajk. Identifying unstructured systems in the errors-in-variables context. In *Proc. of 18th World Congress of the International Federation of Automatic Control*, pages 13104–13109, Milano, Italy, August–September 2011. Paper ThC23.2.

[8] Levente Hunyadi. Fitting implicitly defined shapes to scattered data points with parameters subject to constraints. In István Vajk and Renáta Iváncsy, editors, *Proc. of Automation and Applied Computer Science Workshop*, pages 197–208, Budapest, Hungary, June 2011.

[9] Levente Hunyadi and István Vajk. Identification of linearizable dynamic systems in the errors-in-variables framework. In *Proc. of 1st International Scientific Workshop on Distributed Control Systems*, pages 37–42, Miskolc–Lillafüred, October 2011.

[10] Levente Hunyadi. Fitting implicit curves to scattered data points. In István Vajk and Renáta Iváncsy, editors, *Proc. of Automation and Applied Computer Science Workshop (AACS)*, pages 37–48, Budapest, Hungary, June 2010.

[11] Levente Hunyadi and István Vajk. Implicit model fitting to an unorganized set of points. In *Proc. of IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics (ICCC-CONTI 2010)*, pages 487–491, Timişoara (Temesvár), Romania, May 2010. paper 85.

[12] Levente Hunyadi and István Vajk. Fitting an implicit model in the errors-in-variables context. In Attila K. Varga and József Vásárhelyi, editors, *Proc. of 11th International Carpathian Control Conference*, pages 359–362, Eger, Hungary, May 2010.

[13] Levente Hunyadi and István Vajk. Model reconstruction from a point cloud in the errors-in-variables context. In Bikfalvi Péter, editor, *Proc. of microCAD International Scientific Conference*, pages 65–70, Miskolc, Hungary, March 2010.

[14] Levente Hunyadi. Implicit function reconstruction in the errors-in-variables context. In Anikó Szakál, editor, *Proc. of 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, pages 601–612, Budapest, Hungary, November 2009.

[15] Levente Hunyadi and István Vajk. An approach to identifying linearizable dynamic errors-in-variables systems. In *Proc. of 10th International PhD Workshop on Systems and Control*, Hluboka nad Vltavou, Czech Republic, September 2009. paper 102.

[16] Levente Hunyadi and István Vajk. Separation methods for dynamic errors-in-variables system identification. In *Proc. of European Control Conference (ECC)*, pages 460–465, Budapest, Hungary, August 2009.

[17] Levente Hunyadi. A cross-validating method for simultaneous model and noise parameter estimation in errors-in-variables systems. In István Vajk, editor, *Proc. of Automation and Applied Computer Science Workshop*, pages 39–52, Budapest, Hungary, June 2009.

[18] Levente Hunyadi. A parameter estimation approach to dynamic errors-in-variables systems with polynomial nonlinearities. In István Vajk, editor, *Proc. of Automation and Applied Computer Science Workshop*, pages 25–38, Budapest, Hungary, June 2009.

[19] Levente Hunyadi and István Vajk. Estimating parameters of dynamic errors-in-variables systems with polynomial nonlinearities. In Metin Demiralp, N. A. Baykara, and N. E. Mastorakis, editors, *Signal Processing Systems, Proc. of 8th WSEAS International Conference on Signal Processing (SIP)*, pages 73–78, Istanbul, Turkey, June 2009.

[20] Levente Hunyadi and István Vajk. Polinomiális jellegű nemlinearitásokból álló dinamikus rendszer paraméterbecslése. In László Lehoczky, editor, *Proc. of microCAD International Scientific Conference*, pages 55–60, Miskolc, Hungary, March 2009.

[21] Levente Hunyadi and István Vajk. Identifying dynamic systems with a nonlinear extension of the generalized Koopmans-Levin method. In László Lehoczky, editor, *Proc. of microCAD International Scientific Conference*, pages 49–54, Miskolc, Hungary, March 2009.

[22] Levente Hunyadi. Separation techniques for errors-in-variables parameter estimation. In Anikó Szakál, editor, *Proc. of 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, pages 467–478, Budapest, Hungary, November 2008.

[23] Levente Hunyadi. An identification approach to dynamic errors-in-variables systems with a preliminary clustering of observations. In *Proc. of 6th Conference of PhD Students in Computer Science (CSCS)*, Szeged, Hungary, July 2008. Awarded "Best talk" in session Operation Research.

[24] Levente Hunyadi. Methods for simultaneous estimation of process and noise parameters in errors-in-variables systems. In István Vajk, editor, *Proc. of Automation and Applied Computer Science Workshop*, pages 115–126, Budapest, Hungary, June 2008.

[25] Levente Hunyadi. A framework for comparing errors-in-variables estimation algorithms. In István Vajk, editor, *Proc. of Automation and Applied Computer Science Workshop*, pages 127–138, Budapest, Hungary, June 2008.

[26] Levente Hunyadi and István Vajk. Identification of errors-in-variables systems using data clustering. In Gregor Rozinaj, Jozef Čepko, Peter Trúchly, Ján Vrabec, and Juraj Vojtko, editors, *Proc. of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 197–200, Bratislava, Slovakia, June 2008.

[27] Levente Hunyadi and István Vajk. Parameter estimation for errors-in-variables systems using partitioned input data. In Octavian Prostean, Gheorghe-Daniel Andreescu, and Dan Pescaru, editors, *Proc. of the 8th International Conference on Technical Informatics (CONTI)*, pages 47–52, Timişoara (Temesvár), Romania, June 2008.

[28] Levente Hunyadi. Identification methods for dynamic errors-in-variables systems. In László Lehoczky, editor, *Proc. of microCAD International Scientific Conference*, pages 57–62, Miskolc, Hungary, March 2008.