Regression for sets of polynomial equations

Franz J. Király*, Paul von Bünau, Jan S. Müller, Duncan A. J. Blythe, Frank C. Meinecke, Klaus-Robert Müller
Berlin Institute of Technology (TU Berlin), Machine Learning group, Franklinstr. 28/29, 10587 Berlin

Abstract

We propose a method called *ideal regression* for approximating an arbitrary system of polynomial equations by a system of a particular type. Using techniques from approximate computational algebraic geometry, we show how we can solve ideal regression directly without resorting to numerical optimization. Ideal regression is useful whenever the solution to a learning problem can be described by a system of polynomial equations. As an example, we demonstrate how to formulate Stationary Subspace Analysis (SSA), a source separation problem, in terms of ideal regression, which also yields a consistent estimator for SSA. We then compare this estimator in simulations with previous optimization-based approaches for SSA.

1 Introduction

Regression analysis explains the relationship between covariates and a target variable by estimating the parameters of a function which best fits the observed data. The function is chosen to be of a particular type (e.g. linear) to facilitate interpretation or computation. In this paper, we introduce a similar concept to sets of polynomials equations: given arbitrary input polynomials, the aim is to find a set of polynomials of a particular type that best approximates the set of solutions of the input. As in ordinary regression, these polynomials are parameterized to belong to a certain desired class. This class of polynomials is usually somewhat simpler than the input polynomials. We call this approach *ideal regression*, inspired by the algebraic concept of an ideal in a ring. In fact, the algorithm that we derive is based on techniques from approximate computational algebraic geometry. In machine learning, ideal regression is useful whenever the solution to a learning problem can be written in terms of a set of polynomial equations. We argue that our ideal regression framework has several advantages: (a) it allows to naturally formulate regression problems with intrinsical algebraic structure; (b) our algorithm solves ideal regression directly instead of resorting to less efficient numerical optimization; and (c) the algebraic formulation is amenable to a wide range of theoretical tools from algebraic geometry. We will demonstrate these points in an application to a concrete parametric estimation task.

More formally, the analogy between ordinary and ideal regression is as follows. In ordinary regression, we are given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^D \times \mathbb{R}$ and the aim is to determine parameters θ for a regression function $f_\theta : \mathbb{R}^D \to \mathbb{R}$ such that $f(x_i)$ fits the target y_i according to some criterion, i.e. informally speaking,

$$f_\theta(x_i) \approx y_i \text{ for all samples } (x_i, y_i) \in \mathcal{D}. \qquad (1)$$

In ideal regression, the input consists of a set of n arbitrary polynomial equations $q_1(T) = \cdots = q_n(T) = 0$ in the vector of variables $T = (T_1, \ldots, T_D)$ which may e.g. correspond to the coordinates of data. That is, the data set $\mathcal{D} = \{q_1, \ldots, q_n\}$ consists of the coefficients of the input polynomials. Let $\mathrm{V}(\mathcal{D}) \subset \mathbb{C}^d$ be the set of solutions to the input equations, i.e.

$$\mathrm{V}(\mathcal{D}) = \{x \in \mathbb{C}^D \mid q(x) = 0 \,\forall p \in \mathcal{D}\},$$

where $p(x)$ denotes the evaluation of the polynomial p on the values x. The aim of ideal regression is to determine another set of polynomials $p_\theta^{(1)}, \ldots, p_\theta^{(m)}$ parametrized by θ that best approximate the input polynomials $\mathcal{D}$ in terms of their set of solutions; that is, informally,

$$\mathrm{V}(p_\theta^{(1)}, \ldots, p_\theta^{(m)}) \approx \mathrm{V}(\mathcal{D}). \qquad (2)$$

* Franz J. Király is also at Discrete Geometry Group (FU Berlin), Arnimallee 2, 14195 Berlin

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

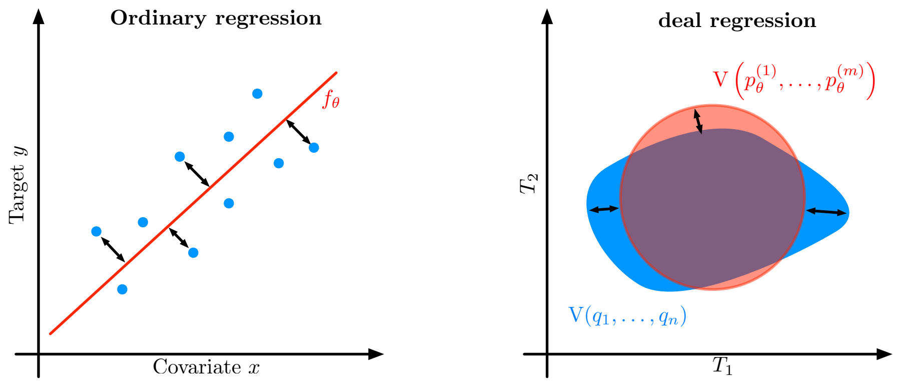
Ordinary regression
Target y
Covariate x
f_θ
deal regression
$\mathrm{V}\left(p_\theta^{(1)},\dots,p_\theta^{(m)}\right)$
T_2
$\mathrm{V}(q_1,\dots,q_n)$
T_1

Figure 1: Ordinary regression (left panel) fits the data (blue points) by a function f_θ (red line) parametrized by θ. Ideal regression (right panel) approximates a set of arbitrary input polynomials $q_1,\dots,q_n$ by a set of polynomials $p_\theta^{(1)},\dots,p_\theta^{(m)}$ that are of a special type parameterized by θ. The approximation is in terms of their set of solutions: the parameter θ is chosen such that $\mathrm{V}(p_\theta^{(1)},\dots,p_\theta^{(m)})$ (red shape) best fits the the solutions to the input (blue shape), in the space of T_1 and T_2.

The class of polynomials (parametrized by θ) by which we approximate arbitrary input is chosen such that it has certain desirable properties, e.g. is easy to interpret or is of a particular type prescribed by the context of the application. Thus, in ordinary regression we fit a parametrized function to arbitrary data and in ideal regression we fit a parametrized system of polynomial equations to arbitrary systems of polynomial equations. Note that even if $\mathrm{V}(\mathcal{D})$ has no exact solution (e.g. due to noise, over-determinedness, or when reducing degrees of freedom), we can still find an approximate regression system close to the inputs. Figure 1 illustrates the analogy between ordinary regression and ideal regression.

The natural algebraic object to parameterize sets of equations up to additive and multiplicative ambiguities are ideals in polynomial rings[1]. The parametric family $p_\theta^{(1)},\dots,p_\theta^{(m)}$ corresponds to a parametric ideal $\mathcal{F}_\theta$ in the polynomial ring $\mathbb{C}[T_1,\dots,T_d]$. In ring theoretic language, the informal regression condition reformulates to

$$\mathcal{F}_\theta \ni_{\text{approx}} \{q_1,\dots,q_n\}, \tag{3}$$

where $\ni_{\text{approx}}$ stands for being approximately contained. Intuitively, this means that the equations q_i are well-approximated by the parametric ideal $\mathcal{F}_\theta$.

The ideal regression setting is fairly general. It can be applied to a wide range of learning settings, including the following.

- **Linear dimension reduction and feature extraction.** When linear features of data are known, ideal regression provides the canonical way to estimate the target parameter with or without "independent" or "dependent" labels. This subsumes PCA dimensionality reduction, linear regression with positive codimension and linear feature estimation.
- **Non-linear polynomial regression.** When the regressor is a set of polynomials with specific structure, as e.g. in positive codimensional polynomial regression or reduced rank regression, ideal regression allows to estimate the coefficients for the regressor polynomials simultaneously.
- **Comparison of moments and marginals.** Ideal regression is the canonical tool when equalities or projections of cumulants are involved. The example we will pursue in the main part of the paper will be of this type, as it is possibly the simplest one where non-linear polynomials occur naturally.
- **Kernelized versions.** Non-linear feature mapping and kernelization is natural to integrate in the regression process as the presented estimator builds on least-squares estimates of vector spaces.

[1]that is, sets of polynomials closed under addition in the set, and under multiplication with arbitrary polynomials

1.1 Example: finding common marginals

In this paper, we will demonstrate ideal regression in a non-linear example: we will reformulate a statistical marginalization task as ideal regression. Namely, we study the following problem:

Problem 1.1 *Given D-variate random variables $X_1,\dots,X_m$, find a projection $P\in\mathbb{R}^{d\times D}$ to a d-dimensional subspace under which which the X_i are*

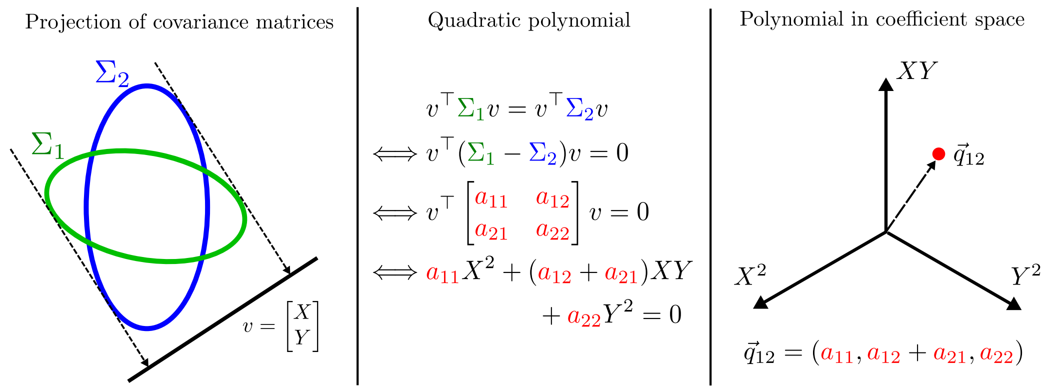
Projection of covariance matrices
Σ_2
Σ_1
$v = \begin{bmatrix} X \\ Y \end{bmatrix}$
Quadratic polynomial
$v^\top \Sigma_1 v = v^\top \Sigma_2 v$
$\iff v^\top(\Sigma_1 - \Sigma_2)v = 0$
$\iff v^\top \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} v = 0$
$\iff a_{11}X^2 + (a_{12} + a_{21})XY + a_{22}Y^2 = 0$
Polynomial in coefficient space
XY
$\vec{q}_{12}$
X^2
Y^2
$\vec{q}_{12} = (a_{11}, a_{12} + a_{21}, a_{22})$

Figure 2: Representation of the problem: the left panel shows the covariance matrices Σ_1 and Σ_2 with the desired projection v. In the middle panel, this projection is defined as the solution to a quadratic polynomial. This polynomial is embedded in the vector space of coefficients spanned by the monomials X^2, Y^2 and XY shown in the right panel.

identically distributed, i.e.

$$PX_1 \sim \cdots \sim PX_m.$$

For example, the X_i can model different data clusters or epochs, for which we want to find an informative common projection P. A special subcase of this problem, where the X_i are approximated by Gaussians, is Stationary Subspace Analysis [23, 19] which has been applied successfully to Brain-Computer-Interfacing [25], Computer Vision [18], domain adaptation [10], geophysical data analysis and feature extraction for change point detection [24, 2]. Previous SSA algorithms [23, 10, 12] have addressed this task by finding the minimum of an objective function that measures the difference of the projected cumulants on the sought-after subspace.

Under the assumption that such a linear map P exists, we can describe the set of all maps yielding common marginals. A necessary (and in practice sufficient) condition is that the projections of the cumulants under P agree. Thus the coefficients of the polynomial equations are given by the coefficients of the cumulants of the X_i (see Section 2 for details). The output ideals correspond to the possible row-spans of P; note that the fact that the PX_i have identical distribution depends only on the row-span of P. Equivalently, the regression parameter θ ranges over the set all sub-vector spaces of dimension d in D-space, i.e. over the Grassmann manifold $\mathrm{Gr}(d, D)$. The regression ideal $\mathcal{F}_\theta$ is then just $\mathrm{I}(\theta)$, the ideal of the vector space θ, considered as a subset of complex D-space.

1.2 Outline of the algorithm

In the application of ideal regression studied in this paper, the aim is to determine linear polynomials that have approximately the same vanishing set as the input polynomials derived from differences of cumulants. The representation in which the algorithm computes is the vector space of polynomials: each polynomial is represented by a vector of its coefficients as shown in the right panel of Figure 2. The coefficient vector space is spanned by all monomials of a particular degree, e.g. in Figure 2 the axis correspond to the monomials X^2, XY and Y^2 because all homogeneous polynomials of degree two in two variables can be written as a linear combination of these monomials.

The algorithm uses an algebraic trick: We first generate more and more equations by making the given ones more complicated; then, when their number suffices, we can make them simpler again to end at a system of the desired type. Playing on analogies, we thus call the algorithm the *Münchhausen* procedure[2].

In the first part of the Münchhausen algorithm, we generate the said larger system of equations, having a particular (higher) degree, but the same vanishing set. This is done by multiplying the input equations by all monomials of a fixed degree, as illustrated in the right panel of Figure 3. We will show that for any (generic) input, there always exists a degree such that the resulting polynomials span a certain linear subspace of the coefficient vector space (up to noise), i.e. that the red curve will always cross the blue curve. That is, we show that the coefficient vectors lie approximately on a linear subspace of known dimension (subspace (a) in Figure 3). This is the case because we have assumed that there exists a subspace of the data space on which

[2]After the eponymous and semi-fictional Baron Hieronymus Carl Friedrich Münchhausen, who purportedly pulled himself and his horse out of the swamp by his own hair; compare the Münchhausen trilemma in epistemiology.

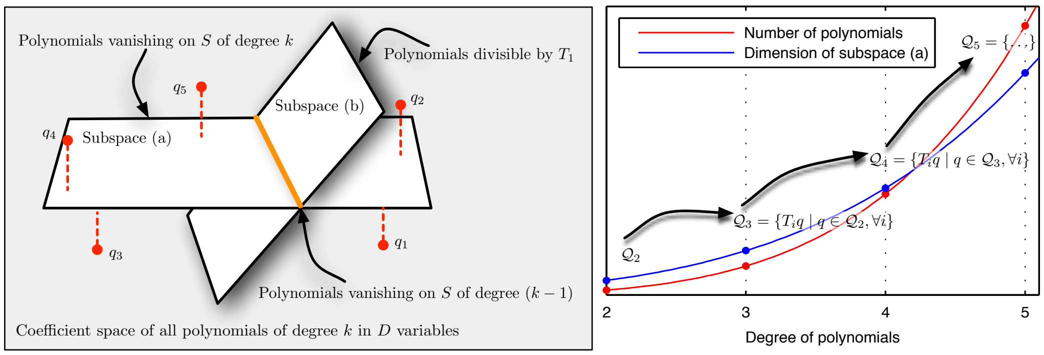
Polynomials vanishing on S of degree k
Polynomials divisible by T_1
Subspace (a)
Subspace (b)
q_1
q_2
q_3
q_4
q_5
Polynomials vanishing on S of degree $(k-1)$
Coefficient space of all polynomials of degree k in D variables
Number of polynomials
Dimension of subspace (a)
$\mathcal{Q}_5 = \{\dots\}$
$\mathcal{Q}_4 = \{T_i q \mid q \in \mathcal{Q}_3, \forall i\}$
$\mathcal{Q}_3 = \{T_i q \mid q \in \mathcal{Q}_2, \forall i\}$
$\mathcal{Q}_2$
2
3
4
5
Degree of polynomials

Figure 3: The left panel shows the vector space of coefficients; the input polynomials (red points) lie approximately on the subspace (a) of polynomials vanishing on S. In order to reduce the degree of the polynomial, we determine the intersection (orange) with the subspace (b) of polynomials that are divisible by a variable T_1. The polynomials in this intersection are isomorphic to the polynomials of degree $k-1$ that vanish on S. The right panel shows the Münchhausen process of multiplying the set of polynomials $\mathcal{Q}_i$ up to a degree so that we have enough elements $|\mathcal{Q}_i|$ to determine the basis for the subspace (a). In the shown case, where we start with $|\mathcal{Q}_2| = 5$ input polynomials in $D = 6$ variables and $\dim S = 3$, we need to go up to degree 5. From $\mathcal{Q}_5$ we then descend to a linear form by repeatedly dividing out a single variable as shown in the left panel.

the marginals agree. After this, we obtain an approximate basis for this subspace in coefficient space by applying PCA dimension reduction; this provides us with an approximate basis for subspace (a) in the left panel of Figure 3.

In the second part of the algorithm, we reduce the degree of the system of equations, i.e. the approximate basis for subspace (a), by repeatedly dividing out single variables[3]. In each step, we reduce the degree by one as illustrated by the left panel of Figure 3. We compute a basis for the intersection (orange line) of subspace (a) and the subspace (b) of polynomials of degree k that are divisible by the variable T_1. By dividing each basis element by T_1 we have obtained a system of equations of lower degree, which has approximately the same set of solutions as the input. We repeat this process until we arrive at a system of linear equations.

1.3 Relation to other work

The ideal regression approach draws inspiration from and integrates several concepts from different fields of research. The first important connection is with computational algebra, as the estimation procedure is essentially a ring theoretic algorithm which can cope with noise and inexact data. In the noiseless case, estimating the regression parameter θ is essentially calculating the radical of a specific ideal, or, more specifically, computing the homogenous saturation of an ideal. These tasks are notoriously known to be very hard in general[4]; however, for generic inputs, the computational complexity somewhat drops into feasible regions, as implied by the results on genericity in the appendix. The best known algorithms for computations of radicals are those of [8], implemented in AXIOM and REDUCE, the algorithm of [6], implemented in Macaulay 2, the algorithm of [3], currently implemented in CoCoA, and the algorithm of [16] and its modification by [17], available in SINGULAR. Closely related to homogenous saturation is also the well-known Buchberger's algorithm for computation of reduced Gröbner basis, which can be seen as the inhomogenous counterpart of homogenous saturation when a degree-compatible term order is applied.

The second contribution comes from the field of approximate and numerical algebra, as the exact algorithms from computational algebra are numerically unstable even under small variations of the inputs and thus unfeasible for direct application to our case. The first application of numerical linear algebra to vector spaces of polynomials can be found in [4], the numerical aspects of noisy polynomials have been treated in [20]. Also, the nascent field of approximate algebra has developed tools to deal with noise, see [15]. In particular, the approximate vanishing ideal algorithm [11] fits polynomial equations to noisy data points with a method that essentially applies a sequence of weighted polynomial kernel regressions. The estimator for ideal regression given in this paper is essentially a deterministic algorithm using approximate computational

[3] i.e. saturating with the homogenizing variable

[4] namely, doubly exponential in the number of variables, see e.g. [17, section 4.]

algebra.

On the conceptual level, the idea of using (linear and commutative) algebra as an ingredient in statistical modelling and for solution of statistical problems is natural, as algebra is the science of structure. Therefore, studying structure in a statistical context makes algebra under stochastical premises a canonical tool. This idea gives rise to a plethora of different approaches, which today are subsumed as the field of algebraic statistics - in the broader meaning of the term. Standard references in the field include [21], [5] and [9], or [1] and [26]. Also, there is a range of machine learning methods dealing with non-linear algebraic structure or symmetries in data, e.g. [13, 14] or [22]. In many applications, the concept of genericity also arises as the algebraic counterpart of statistical nondegeneracy; where it is mostly applied to the choice of model parameters and the study of a general such. Also, algebraic geometry and commmutative algebra are already successfully applied there to obtain theoretical results on statistical objects and methods.

2 The algorithm

The probability distribution of every smooth real random variable X can be fully characterized in terms of its *cumulants*, which are the tensor coefficients of the cumulant generating function. Before we continue, we introduce a useful shorthand notation for linearly transforming tensors, i.e. cumulants.

Definition 2.1 *Let $A \in \mathbb{C}^{d\times D}$ be a matrix. For a tensor $T \in \mathbb{R}^{D^{(\times k)}}$, we will denote by $A\circ T$ the application of A to T along all tensor dimensions, i.e.*

$$(A \circ T)_{i_1\dots i_k} = \sum_{j_1=1}^{D} \cdots \sum_{j_k=1}^{D} A_{i_1 j_1} \cdot \ldots \cdot A_{i_k j_k} T_{j_1 \dots j_k}.$$

Using this, we can now define the cumulants of a D-dimensional smooth real random variable X via the Taylor expansion of the cumulant generating function.

Definition 2.2 *Let X be a smooth real D-dimensional random variable. Then the cumulant generating function of X is defined as $\chi_X(\tau) = \log\left(\mathbb{E}\left[e^{i\tau^\top X}\right]\right) = \sum_{k=1}^{\infty}(i\tau)\circ\frac{\kappa_k(X)}{k!}$, where $\tau \in \mathbb{R}^D$. The coefficient tensors $\kappa_k(X)$ are called the k-th cumulants of X.*

For the problem addressed in this paper, cumulants are a particularly suitable representation because the cumulants of a linearly transformed random variable are the multilinearly transformed cumulants, as a classical and elementary calculation shows:

Proposition 2.3 *Let X be a smooth real D-dimensional random variable and let $A \in \mathbb{R}^{d\times D}$ be a matrix. Then the cumulants of the transformed random variable $A \cdot X$ are the transformed cumulants, $\kappa_k(AX) = A \circ \kappa_k(X)$ where $\circ$ denotes the application of A along all tensor dimensions.*

We now derive an algebraic formulation for Problem 1.1: note that $PX_i \sim PX_j$ if and only if $vX_i \sim vX_j$ for all row vectors $v \in \operatorname{span} P^\top$.

Problem 2.4 *Find all d-dimensional linear subspaces in the set of vectors*

$$\begin{aligned} S &= \{v \in \mathbb{R}^D \mid v^\top X_1 \sim \cdots \sim v^\top X_m\} \\ &= \{v \in \mathbb{R}^D \mid v^\top \circ \kappa_k(X_i) = v^\top \circ \kappa_k(X_j), \\ &\qquad\qquad k \in \mathbb{N}, 1 \le i, j \le m\} \\ &= \{v \in \mathbb{R}^D \mid v^\top \circ (\kappa_k(X_i) - \kappa_k(X_m)) = 0, \\ &\qquad\qquad k \in \mathbb{N}, 1 \le i < m\}. \end{aligned}$$

The equivalence of the problems then follows from the fact that the projection P can be characterized by its row-span which is a d-dimensional linear subspace in S. Note that while we are looking for linear subspaces in S, in general S itself is not a vector space. Apart from the fact that S is homogeneous, i.e. $\lambda S = S$ for all $\lambda \in \mathbb{R}\backslash\{0\}$, there is no additional structure that we utilize. To use the tools from computational algebra, we now only need to consider the left hand side of each of the equations as polynomials $f_1, \dots, f_n$ in the variables $T_1, \dots, T_D$,

$$f_j = \left[T_1 \cdots T_D\right] \circ (\kappa_k(X_i) - \kappa_k(X_m)),$$

with j running through n combinations of i and k. The f_j are formally elements of the polynomial ring over the complex numbers $\mathbb{C}[T_1, \dots, T_D]$. In particular, if we restrict ourselves to a finite number of cumulants, we can write S as the set of solutions to a finite number n of polynomial equations,

$$S = \left\{v \in \mathbb{R}^D \mid f_1(v) = \cdots = f_n(v) = 0,\ 1 \le j \le k\right\},$$

which means that S is an *algebraic set* or an *algebraic variety.* Thus, in the language of algebraic geometry, we can reformulate the problem as follows.

Problem 2.5 *Find all d-dimensional linear subspaces in the algebraic set*

$$S = \mathrm{V}(f_1, \dots, f_n).$$

In order to describe in algebraic terms how this can be done we need to assume that a unique solution indeed exists, while assuming as little as possible about the given polynomials. Therefore we need to employ the concept of *generic polynomials*, which is defined rigorously in the supplemental material. Informally, a polynomial is generic when it does not fulfill any other conditions than the imposed ones and those which follow logically. This can be modelled by assuming that the coefficients are independently sampled from a sufficiently general probability distribution (e.g. Gaussians), only subject to the imposed constraints. The statements following the assumption of genericity are then probability-one statements under those sampling conditions.

In our context, we can show that under genericity conditions on the $f_1, \ldots, f_n$, and when their number n is large enough, the solution is unique and equivalent to finding a linear generating set for the radical of the ideal $\langle f_1, \ldots, f_n \rangle$, which equals its homogenous saturation, as Corollary C.7 in the supplemental material asserts:

Problem 2.6 *Let $f_1, \ldots, f_n$ with $n \geq D+1$ be generic homogenous polynomials vanishing on a linear d-dimensional subspace $S \subseteq \mathbb{C}^D$. Find a linear generating set $\ell_1, \ldots, \ell_{D-d}$ for the radical ideal*

$$(\langle f_1, \ldots, f_n \rangle : T_D) = \sqrt{\langle f_1, \ldots, f_n \rangle} = \mathrm{I}(S).$$

3 Approximate Algebraic Computations

In this section we present the Münchhausen algorithm which computes the homogenous saturation in Problem 2.6 and thus computes the ideal regression in the marginalization problem. The algorithm for the general case is described in the appendix in section C.2.

The efficiency of the algorithm stems largely from the fact that we operate with linear representations for polynomials. That is, we first find enough elements in the ideal $\langle f_1, \ldots, f_n \rangle$ which we then represent in terms of coefficient vectors. In this vector space we can then find the solution by means of linear algebra. An illustration of this representation is shown in Figure 2. Let us first introduce tools and notation.

Notation 3.1 *We will write $R = \mathbb{C}[T_1, \ldots, T_D]$ for the ring of polynomials over $\mathbb{C}$ in the variables $T_1, \ldots, T_D$. We will denote the ideal of the d-dimensional linear space $S \subseteq \mathbb{C}^D$ by $\mathfrak{s} = \mathrm{I}(S)$.*

In order to compactly write sub-vector spaces of certain degree, we introduce some notation.

Notation 3.2 *Let $\mathcal{I}$ be an ideal of R. Then we will denote the vector space of homogenous polynomials of degree k in $\mathcal{I}$ by $\mathcal{I}_k$.*

The dimension of these sets can be later written compactly in terms of simplex numbers, for which we introduce an abbreviating notation.

Notation 3.3 *We denote the b-th a-simplex number by $\Delta(a,b) = \binom{a+b-1}{a}$, which is defined to be zero for $a < 0$.*

Since the polynomials arise from estimated cumulants we need to carry all algebraic computations out approximately. The crucial tool is to minimize distances in coefficient space using the singular value decomposition (SVD).

Definition 3.4 *Let $A \in \mathbb{C}^{m \times n}$ be a matrix, let $A = UDV^\top$ be its SVD. The approximate row span of A of rank k is the row span of the first k rows of V; the approximate left null space of A of rank k is the row span of the last k rows of U.*

These approximate spaces can be represented by matrices consisting of row vectors spanning them, the so-called approximate left null space matrix and approximate row span matrix.

The key to the problem is the fact that there exists a degree N such that the ideal generated by the $f_1, \ldots, f_n$ contains all homogenous polynomials of degree N in $\mathfrak{s}$. This allows us to increase the degree until we arrive at a vector space where it suffices to operate linearly (see Figure 3).

Theorem 3.5 *Let $f_1, \ldots, f_n \in \mathfrak{s}$ be generic homogenous polynomials in D variables of fixed degrees $d_1, \ldots, d_n$ each, such that $n > D$. Let $\mathcal{I} = \langle f_1, \ldots, f_n \rangle$. Then one has*

$$(\mathcal{I} : T_i) = \mathfrak{s}$$

for any variable T_i. In particular, there exists an integer N such that

$$\mathcal{I}_N = \mathfrak{s}_N.$$

N is bounded from below by the unique index M belonging to the first non-positive coefficient a_M of the power series

$$\sum_{k=0}^{\infty} a_k t^k = \frac{\prod_{i=1}^{n}(1-t^{d_i})}{(1-t)^D} - \frac{1}{(1-t)^d}.$$

If we have $d_i \leq 2$, and $D \leq 11$, then equality holds.

This summarizes Proposition B.36, Corollary B.41 and Theorem B.42 from the supplemental material for the case $\mathfrak{s} = \mathrm{I}(S)$, the proof can be found there. In the appendix, we conjecture that the statement on N is also valid for general d_i and D - this generalizes Fröberg's conjecture on Hilbert series of semi-regular sequences

[7]. In the supplemental material, we give an algorithm which can be used to prove the conjecture for fixed d_i and D and thus give an exact bound for N in those cases. Theorem 3.5 guarantees that given our input polynomials, we can easily obtain a basis for the vector spaces of homogenous polynomials $\mathfrak{s}_k$ with $k \geq N$. In this vector space, we are interested in the polynomials divisible by a fixed monomial T_i; they form the vector space $\mathfrak{s}_k \cap \langle T_i \rangle = (\mathfrak{s} \cap \langle T_i \rangle)_k$. By dividing out the monomials T_i, we can then obtain the vector space of homogenous polynomials $\mathfrak{s}_{k-1}$ of degree one less.

Algorithm 1 The ideal regression estimator.
Input: Generic homogenous polynomials $f_1, \ldots, f_n \in \mathfrak{s}, n \geq D$; the dimension d of S.
Output: An approximate linear generating set $\ell_1, \ldots, \ell_{D-d}$ for the ideal $\mathfrak{s}$.

```
1: Determine some necessary degree $N$ according to Theorem 3.5 (e.g. with Algorithm 2, or via Conjecture B.39/Algorithm 1)
2: Initialize $Q \leftarrow [\,]$ with the empty matrix.
3: for $i = 1 \ldots n$ do
4:     for all monomials $M$ of degree $N - \deg f_i$ do
5:         Add a row vector of coefficients, $Q \leftarrow \begin{bmatrix} Q \\ f_i M \end{bmatrix}$
6:     end for
7: end for
8: for $k = N \ldots 2$ do
9:     Set $Q \leftarrow \text{ReduceDegree}(Q)$
10: end for
11: Compute the approximate row span matrix $A \leftarrow \begin{bmatrix} a_1 \cdots a_{D-d} \end{bmatrix}^\top$ of $Q$ of rank $D - d$
12: Let $\ell_i \leftarrow \begin{bmatrix} T_1 & \cdots & T_D \end{bmatrix} a_i$ for all $1 \leq i \leq D - d$
```

We explain how to proceed in the case where $\mathfrak{s} = \mathrm{I}(S)$ is linear, and the f_i are generic. The case of general $\mathfrak{s}$ can be found in the appendix. So suppose that we have enough quadratic polynomials. Then we can determine an approximate basis for the vector space of homogenous degree 2 polynomials vanishing on S which is $\mathfrak{s}_2$, because our input polynomials lie approximately on that subspace. From this we can obtain the linear homogenous polynomials $\mathfrak{s}_1$ by dividing out any monomial T_i; a basis of $\mathfrak{s}_1$ can be directly used to obtain a basis of S. More generally, if we know a basis of the vector space of homogenous degree k polynomials vanishing on S which is $\mathfrak{s}_k$, we can obtain from it a basis of $\mathfrak{s}_{k-1}$ in a similar way; by repeating this stepwise, we eventually arrive at $\mathfrak{s}_1$. This degree reducing procedure is approximatively applied in Algorithm 2.

We now describe the Münchhausen Algorithm 1 in detail. In Step 1, we calculate a degree N up to which we need to multiply our input polynomials so that we have enough elements to approximately span the whole space $\mathfrak{s}_N$ of polynomials vanishing on S. In the Steps 2 to 7 we multiply the input polynomials up to the necessary degree N and arrange them as coefficient vectors in the matrix Q. This process is illustrated in the right panel of Figure 3. The rows of Q approximately span the space of polynomials vanishing on S, i.e. space (a) in the left panel of Figure 3. In Steps 8 to 10 we invoke Algorithm 2 to reduce the degree of the polynomials in Q until we have reached an approximate linear representation $\mathfrak{s}_1$. Finally, in Step 11 we reduce the set of linear generators in Q to the principal $D - d$ ones.

Algorithm 2 ReduceDegree (Q).
Input: Approximate basis for the vector space $\mathfrak{s}_k$ given as the rows of the $(n \times \Delta(n, D))$-matrix Q; the dimension d of S.
Output: Approximate basis for the vector space $\mathfrak{s}_{k-1}$, given as the rows of the $(n' \times \Delta(n-1, D))$-matrix A

```
1: for $i = 1 \ldots D$ do
2:     Let $Q_i \leftarrow$ the submatrix of $Q$ obtained by removing all columns corresponding to monomials divisible by $T_i$
3:     Compute $L_i \leftarrow$ the approximate left null space matrix of $Q_i$ of rank $m - \Delta(k, D) + \Delta(k, d) + \Delta(k-1, D) - \Delta(k-1, d)$
4:     Compute $L_i' \leftarrow$ the approximate row span matrix of $L_i Q$ of rank $\Delta(k-1, D) - \Delta(k-1, d)$
5:     Let $L_i'' \leftarrow$ the matrix obtained from $L_i'$ by removing all columns corresponding to monomials not divisible by $T_i$
6: end for
7: Let $L \leftarrow$ the matrix obtained by vertical concatenation of all $L_i''$
8: Compute $A \leftarrow$ the approximate row span matrix of $L$ of rank $n' = \min(n, D(\Delta(k-1, D) - \Delta(k-1, d)))$
```

Given a set of polynomials of degree k that vanish approximately S, Algorithm 2 computes another set polynomials of degree $k - 1$ with the same property. This is achieved by dividing out variables *approximately*, in a way that utilizes as much information as possible to reduce the influence of estimation errors in the coefficients of the input polynomials. Approximate division by a variable T_i means that we find linear combinations of our input polynomials such the coefficients of all monomials not divisible by T_i are as small as possible. Given a matrix of coefficient row vectors Q of degree k, for each variable that we divide out the result is a matrix L_i'' of polynomials of degree $k-1$ that also vanish approximately on the set of solutions S. We iterate over all variables in the for-loop and combine the results $L_1'', \ldots, L_D''$ in the final

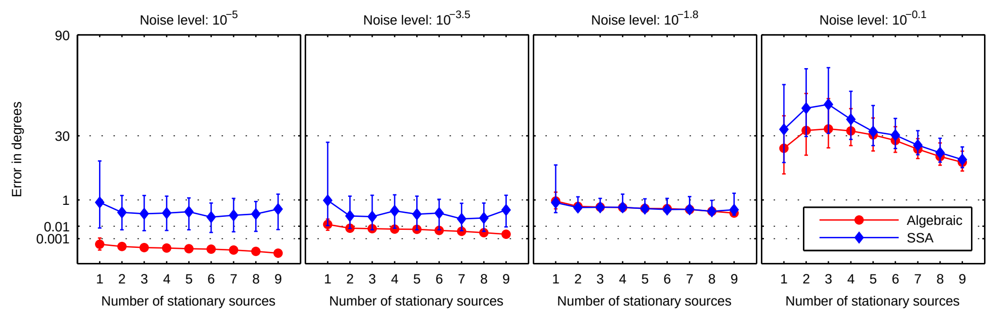

Figure 4: Comparison of the ideal regression algorithm and the SSA algorithm. Each panel shows the median error of the two algorithms (vertical axis) for varying numbers of stationary sources in ten dimensions (horizontal axis). The noise level increases from the left to the right panel; the error bars extend from the 25% to the 75% quantile.

Steps 7 and 8. In order to find L''_i for each variable T_i, we first determine a matrix L_i that minimizes the coefficients for all monomials in Q that are not divisible by T_i. To that end, in Step 2 we remove all monomials not divisible by T_i to get a matrix Q_i for which we then compute the left null space matrix L_i in Step 3. The product L_iQ_i is then a set of polynomials of degree k with minimal coefficients for all monomials not divisible by T_i. These polynomials lie approximately on the span of polynomials vanishing on S. In the next Step 4, we compute an approximate basis L' for this space and in Step 5 we reduce the degree by removing all monomials not divisible by T_i. Finally, in the last Steps 7 and 8 we combine all found solutions $L''_1, \ldots, L''_D$ using PCA. Note that both algorithm are deterministic and consistent estimators.

4 Simulations

We investigate the influence of the noise level and the number of dimensions on the accuracy and the runtime of the ideal regression algorithm in the special case of covariance matrices and compare it to the standard method for this case, the SSA algorithm [23]. We measure the accuracy using the subspace angle between the true and the estimated space of projections. The setup of the synthetic data is as follows: we fix the total number of dimensions to $D = 10$ and vary the dimension d of the subspace with equal probability distribution from one to nine. We also fix the number of random variables to $m = 26$, yielding $n = 25$ quadratic polynomials. For each trial of the simulation, we choose a random basis for the subspace S and random covariance matrices to which we add a disturbance matrix parametrized by the noise level σ.

The results are shown in Figure 4. With increasing noise levels both algorithms become worse. For all noise levels, the algebraic method yields significantly better results than the standard optimization-based approach, over all dimensionalities. In Figure 4, we see that the error level of the ideal regression algorithm decreases with the noise level, converging to the exact solution when the noise tends to zero. In contrast, the error of original SSA decreases with noise level, reaching a minimum error baseline which it cannot fall below. In particular, the algebraic method significantly outperforms SSA for low noise levels, approaching machine precision. At high noise levels, the algebraic method outperforms SSA on average, having lower error variance than SSA.

5 Conclusion

In this paper, we have presented the framework of ideal regression and an estimator which uses approximate computational algebra. Moreover, we have worked through a specific example: we have shown that the problem of finding common projections of marginals can be reformulated in terms of ideal regression and we have derived a practical algorithm, that we have evaluated in numerical simulations. Also, due to the algebraic formulation of the problem, we were able to derive previously unapproachable theoretical results on the estimation problem. We argue for a cross-fertilization of machine learning and approximate computational algebra: the former can benefit from the wealth of techniques for dealing with uncertainty and noisy data; the machine learning community may find in ideal regression a novel framework for representing learning problems and powerful proof techniques.

Acknowledgments

We thank Gert-Martin Greuel and Bernd Sturmfels for insightful discussion and we gratefully acknowledge support by the Mathematisches Forschungsinstitut Oberwolfach (MFO).

References

[1] Shun'ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of mathematical monographs.* American Mathematical Society, 2000.

[2] D.A.J. Blythe, P. von Bünau, F.C Meinecke, and K.R. Müller. Feature extraction for high dimensional change point detection using stationary subspace analysis. *Online Pre-Print (Submitted to IEEE Transactions on Neural Networks)*, 2011. arXiv:1108.2486.

[3] Massimo Caboara, Pasqualina Conti, and Carlo Traverse. Yet another ideal decomposition algorithm. In Teo Mora and Harold Mattson, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1255 of *Lecture Notes in Computer Science*, pages 39–54. Springer Berlin / Heidelberg, 1997.

[4] R.M. Corless, Patrizia M. Gianni, Barry M. Trager, and S.M. Watt. The singular value decomposition for polynomial systems. *Proc. ISSAC '95*, pages 195–207, 1995.

[5] Mathias Drton, Bernd Sturmfels, and Seth Sullivant. *Lectures on Algebraic Statistics.* Oberwolfach Seminars. Birkhauser Basel, 2010.

[6] David Eisenbud, Craig Huneke, and Wolmer Vasconcelos. Direct methods for primary decomposition. *Inventiones Mathematicae*, 110:207–235, 1992.

[7] Ralf Fröberg and Joachim Hollman. Hilbert series for ideals generated by generic forms. *Journal of Symbolic Computation*, 17(2):149 – 157, 1994.

[8] Patrizia Gianni, Barry Trager, and Gail Zacharias. Groebner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(2-3):149 – 167, 1988.

[9] Paolo Gibilisco, Eva Riccomagno, Maria Piera Rogantin, and Henry P. Wynn. *Algebraic and Geometric Methods in Statistics.* Cambridge University Press, 2010.

[10] Satoshi Hara, Yoshinobu Kawahara, Takashi Washio, and Paul von Bünau. Stationary subspace analysis as a generalized eigenvalue problem. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ICONIP'10, pages 422–429, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, and Hennie Poulisse. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation*, 44(11):1566 – 1591, 2009. In Memoriam Karin Gatermann.

[12] Motoaki Kawanabe, Wojciech Samek, Paul von Bünau, and Frank Meinecke. An information geometrical view of stationary subspace analysis. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 397–404. Springer Berlin / Heidelberg, 2011.

[13] Risi Kondor. The skew spectrum of functions on finite groups and their homogeneous spaces, 2007. Eprint arXiv:0712.4259.

[14] Risi Kondor and Karsten M. Borgwardt. The skew spectrum of graphs. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 496–503, New York, NY, USA, 2008. ACM.

[15] Martin Kreuzer, Hennie Poulisse, and Lorenzo Robbiano. *Approximate Commutative Algebra*, chapter From Oil Fields to Hilbert Schemes. Springer-Verlag Berlin Heidelberg, 2009.

[16] Teresa Krick and Alessandro Logar. An algorithm for the computation of the radical of an ideal in the ring of polynomials. In Harold Mattson, Teo Mora, and T. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 539 of *Lecture Notes in Computer Science*, pages 195–205. Springer Berlin / Heidelberg, 1991.

[17] Santiago Laplagne. An algorithm for the computation of the radical of an ideal. In *Proceedings of the 2006 international symposium on Symbolic and algebraic computation.* ACM, 2006.

[18] Frank C. Meinecke, Paul von Bünau, Motoaki Kawanabe, and Klaus-Robert Müller. Learning invariances with stationary subspace analysis. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 87 –92, 2009.

[19] J.S. Müller, P. von Bünau, F.C. Meinecke, F.J. Király, and Klaus-Robert Müller. The stationary subspace analysis toolbox. *Journal of Machine Learning Research*, 12:3065–3069, 2011.

[20] Hans J. Stetter. *Numerical polynomial algebra.* Society for Industrial and Applied Mathematics, 2004.

[21] Bernd Sturmfels. *Solving Systems of Polynomial Equations*, volume 97 of *CBMS Regional Conferences Series.* Amer. Math. Soc., Providence, Rhode Island, 2002.

[22] René Vidal, Yi Ma, and Sastry Shankar. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

[23] Paul von Bünau, Frank C. Meinecke, Franz J. Király, and Klaus-Robert Müller. Finding stationary subspaces in multivariate time series. *Phys. Rev. Lett.*, 103(21):214101, Nov 2009.

[24] Paul von Bünau, Frank C. Meinecke, Jan S. Müller, Steven Lemm, and Klaus-Robert Müller. Boosting high-dimensional change point detection with stationary subspace analysis. In *Workshop on Temporal Segmentation at NIPS 2009.* 2009.

[25] Paul von Bünau, Frank C. Meinecke, Simon Scholler, and Klaus-Robert Müller. Finding stationary brain sources in EEG data. In *Proceedings of the 32nd Annual Conference of the IEEE EMBS*, pages 2810–2813, 2010.

[26] Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, United Kingdom, 2009.