Computational Optimization and Applications (2023) 84:421–447
https://doi.org/10.1007/s10589-022-00420-9

Loss functions for finite sets

Jiawang Nie[1] · **Suhan Zhong**[2]

Received: 1 July 2022 / Accepted: 3 October 2022 / Published online: 14 October 2022

Abstract
This paper studies loss functions for finite sets. For a given finite set S, we give sum-of-square type loss functions of minimum degree. When S is the vertex set of a standard simplex, we show such loss functions have no spurious minimizers (i.e., every local minimizer is a global one). Up to transformations, we give similar loss functions without spurious minimizers for general finite sets. When S is approximately given by a sample set T, we show how to get loss functions by solving a quadratic optimization problem. Numerical experiments and applications are given to show the efficiency of these loss functions.

Keywords Loss function · Finite set · Polynomial · Spurious minimizer · Optimization

Mathematics Subject Classification 90C23 · 65K05 · 90C30

1 Introduction

This paper studies loss functions for finite sets. The questions of concerns are: for a finite set, how do we construct a convenient loss function for it? When does the loss function have no spurious optimizers, i.e., every local optimizer is also a global one? We discuss these topics in this paper. Let n, k be positive integers. Suppose S is a set of k distinct points in the n-dimensional real Euclidean space $\mathbb{R}^n$. A function f in $x := (x_1, \ldots, x_n)$ is said to be a loss function for S if the global minimizers of f are precisely the points in S. For convenience, we often select f such that f is

✉ Suhan Zhong
suzhong@tamu.edu

Jiawang Nie
njw@math.ucsd.edu

[1] Department of Mathematics, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA

[2] Department of Mathematics, Texas A &M University, College Station, TX 77843-3368, USA

nonnegative in $\mathbb{R}^n$ and the minimum value is zero. Mathematically, this is equivalent to that

$$f(x) = 0 \quad \text{if and only if} \quad x \in S. \tag{1.1}$$

When $S = \{u_1, \ldots, u_k\}$, a straightforward choice for the loss function is $f = \|x - u_1\|^2 \cdots \|x - u_k\|^2$, where $\|\cdot\|$ is the standard Euclidean norm. This loss function is a polynomial of degree $2k$ in the variable x. It requires to use all points of S. In applications, the cardinality k may be big. Moreover, the set S often has noises and it may be given by a large number of samplings around the points in S. For this reason, the above choice of loss function may not be convenient in computational practice.

A frequently used loss function is the class of sum-of-squares (SOS) polynomials. That is, the loss function f is in the form

$$f = p_1^2 + \cdots + p_m^2,$$

where each p_i is a polynomial in x. Then f is a loss function for S if and only if each $p_i \equiv 0$ on S. For convenience of computation, we prefer that f and each p_i have degrees as low as possible. A more preferable function is that every local minimizer of f is a global minimizer (i.e., a zero of f). That is, we wish that the loss function f has no *spurious minimizer*.[1] Optimization without spurious minimizers is studied in [22, 26]. Polynomial loss functions have good mathematical properties and are convenient computationally (see [1, 9, 12]). In particular, polynomial optimization problems (especially nonconvex ones) can be efficiently solved by Moment-SOS relaxations. We refer to Fan et al. [8], Lasserre [18–20], Laurent [23, 24], Nie [28, 29] for recent work on polynomial optimization.

In applications, the set S may not be given explicitly. It is often approximately given by a sample set

$$T = \{v_1, \ldots, v_N\},$$

where each v_i is a sample for a point in S and the sample size $N \gg k$. For such a case, we can choose a family $\mathcal{F}$ of loss functions, which is parameterized to represent the set S. Since S is approximated by T, we choose a loss function $f \in \mathcal{F}$ such that the average value of f on T is minimum. Mathematically, this is equivalent to solving the optimization

$$\min_{f \in \mathcal{F}} \quad \frac{1}{N} \sum_{i=1}^{N} f(v_i). \tag{1.2}$$

The optimization (1.2) requires that we choose parameters for f such that the average loss on T is minimum. The set S can be determined by parameters for f in the family $\mathcal{F}$.

[1] A local minimizer that is not a global minimizer is called a spurious minimizer.

Loss functions are useful in data science optimization. There are broad applications of loss functions [2, 4, 5, 10, 17, 31, 32, 35, 38]. Selection of loss functions needs to consider application purposes and data structures. There are various types of loss functions for different applications. We refer to the survey [37] for loss functions in machine learning. Polynomial loss functions are used in optimal control [13, 14]. Linear loss functions are used for network blocking games [21]. Loss functions obtained via statistical averaging are given in [3]. For inverted beta loss functions, their properties and applications are given in [25]. Some properties of Erlang loss functions are given in [15]. Properties of correntropic loss functions are given in [36].

1.1 Contributions

The paper studies loss functions for finite sets. We focus on the SOS type loss functions with minimum degrees. Let S be a given finite set in $\mathbb{R}^n$. We characterize loss functions that satisfy (1.1). When S is approximately given by a set T of larger cardinality, we look for loss functions by solving the optimization (1.2). Let $x := (x_1, \ldots, x_n)$. We consider the loss function f such that $f = p_1^2 + \cdots + p_m^2$, where every p_i is a polynomial in x. The f is a loss function for S if and only if S precisely consists of common real zeros of polynomials $p_1, \ldots, p_m$. Mathematically, this is equivalent to that

$$S = \{v \in \mathbb{R}^n : p_1(v) = \cdots = p_m(v) = 0\}. \tag{1.3}$$

For the polynomial p_i to have minimum degrees, we consider *generating polynomials* for the S, which are introduced for symmetric tensor decomposition [30, 31]. Let Φ be the set of all generating polynomials for S. It is interesting to note that Φ has the minimum degree, such that (1.3) holds. In particular, when S is given by vertices of a standard simplex, the resulting loss function f does not have spurious minimizers. Up to transformations, we can get loss functions without spurious minimizers, for general finite sets. In computational practice, we choose such loss functions of degree four.

When the set S is approximately given by a set T of larger size, we propose to solve the optimization (1.2) to get the loss function. Equivalently, we determine parameters for f from a family $\mathcal{F}$ of loss functions of S. Each $f \in \mathcal{F}$ is determined by a set of parameters, and vice versa. By solving (1.2), we not only get a loss function, but also get a set S^* of k points that are approximations for the points in S. Once S^* is determined, up to transformations, we can use S^* to get loss functions that have no spurious minimizers.

In summary, our major results are:

- For a given finite set S, we give an SOS type loss function of minimum degree, such that S is precisely the set of global minimizers.
- When S consists of the vertices of a standard simplex, we show that the selected loss function has no spurious minimizers. For more general finite sets, we give these loss functions by applying transformations.

- When the set S is approximately given by a sample set T, we solve the optimization (1.2) to get loss functions of similar properties, i.e., they are in SOS type and have minimum degrees.

The paper is organized as follows. In Sect. 2, we briefly review some backgrounds for polynomial ideals. In Sect. 3, we show how to get SOS type loss functions for finite sets, with desired properties. In Sect. 4, when the set S consists of vertices of a standard simplex, we show that the constructed loss functions have no spurious minimizers. For more general S, we show how to get similar loss functions by applying transformations. In Sect. 5, we show how to get loss functions when the set S is approximately given by a sample set T. Some numerical experiments are given in Sect. 6.

2 Preliminaries

2.1 Notation

The symbol $\mathbb{R}$ (resp., $\mathbb{C}$, $\mathbb{N}$) denotes the set of real (resp., complex, nonnegative integer) numbers respectively. The symbol $\mathbb{N}^n$ (resp., $\mathbb{R}^n$, $\mathbb{C}^n$) stands for the set of n-dimensional vectors with entries in $\mathbb{N}$ (resp., $\mathbb{R}$, $\mathbb{C}$) respectively. For an integer $k > 0$, $[k] := \{1, \ldots, k\}$. We use 0 to denote the vector of all zeros and e to denote the vector of all ones. The symbol e_i stands for the unit vector such that the ith entry is one and all other entries are zeros. For a vector v, the $\|v\|$ denotes its Euclidean norm. For a vector $u \in \mathbb{R}^n$ and $\delta \geq 0$, $B(u, \delta) := \{x \in \mathbb{R}^n : \|x - u\| \leq \delta\}$ denotes the closed ball centered at u with radius δ. The symbol I_n denotes the n-by-n identity matrix. The superscript ^{T} (resp., $^{\mathbb{H}}$) denotes the operation of matrix transpose (resp., Hermitian). A square matrix A is said to be positive semidefinite (resp., positive definite) if $x^T Ax \geq 0$ (resp., $x^T Ax > 0$) for all nonzero vectors x. For two square matrices X, Y of the same dimension, their commutator is

$$[X, Y] := XY - YX.$$

That is, X commutes with Y if and only if $[X, Y] = 0$. For a function f which is continuously differentiable in $x = (x_1, \ldots, x_n)$, the ∇f denotes its gradient in x and $\nabla^2 f$ denotes its Hessian.

Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$. Denote by $\mathbb{F}[x] := \mathbb{F}[x_1, \ldots, x_n]$ the ring of polynomials in $x := (x_1, \ldots, x_n)$ with coefficients in $\mathbb{F}$. For every $d \in \mathbb{N}$, $\mathbb{F}[x]_d$ denotes the subspace of $\mathbb{F}[x]$ which contains all polynomials of degree at most d. For every $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, denote the monomial $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Its total degree is $|\alpha| := \alpha_1 + \cdots + \alpha_n$.

A subset $I \subseteq \mathbb{F}[x]$ is an ideal of $\mathbb{F}[x]$ if $p \cdot q \in I$ for all $p \in I$, $q \in \mathbb{F}[x]$, and $p_1 + p_2 \in I$ for all $p_1, p_2 \in I$. For an ideal I, its *radical* is the set

$$\sqrt{I} := \{f \in \mathbb{F}[x] : f^k \in I \text{ for some } k \in \mathbb{N}\}.$$

The set $\sqrt{I}$ is also an ideal and $I \subseteq \sqrt{I}$. The ideal I is said to be radical if $I = \sqrt{I}$. Each ideal I determines the *variety* in $\mathbb{F}^n$ as

$$V_{\mathbb{F}}(I) := \{x \in \mathbb{F}^n \,:\, p(x) = 0\,(p \in I)\}.$$

For a polynomial tuple $p := (p_1, \ldots, p_m)$, we similarly denote that

$$V_{\mathbb{F}}(p) := \{x \in \mathbb{F}^n \,:\, p(x) = 0\}.$$

The tuple p generates the ideal

$$\mathrm{Ideal}(p) := p_1 \cdot \mathbb{F}[x] + \cdots + p_m \cdot \mathbb{F}[x].$$

Clearly, $V_{\mathbb{F}}(\mathrm{Ideal}(p)) = V_{\mathbb{F}}(p)$.

For a set $S \subseteq \mathbb{C}^n$, its vanishing ideal is

$$I(S) := \{q \in \mathbb{C}[x] \,:\, q(u) = 0\,(u \in S)\}.$$

If $S = V_{\mathbb{C}}(p)$ for some polynomial tuple p in x, then $\mathrm{Ideal}(p) \subseteq I(S)$ but the equality may not hold. For every $I \subseteq \mathbb{C}[x]$, we have $I(V_{\mathbb{C}}(I)) = \sqrt{I}$. This is Hilbert's Nullstellensatz [7].

For a given ideal $I \subseteq \mathbb{C}[x]$, it determines an equivalence relation $\sim$ on $\mathbb{C}[x]$ such that $p \sim q$ if $p - q \in I$, or equivalently, $p \equiv q \mod I$. Then every $p \in \mathbb{C}[x]$ corresponds to an equivalence class with the module of I, i.e.,

$$[p] = \{q \in \mathbb{C}[x] \,:\, q \equiv p \mod I\}.$$

The set of all equivalent classes is the quotient ring

$$\mathbb{C}[x]/I := \{[p] \,:\, p \in \mathbb{C}[x]\}.$$

3 A class of loss functions

In this section, we give a general framework of constructing loss functions for finite sets. For convenience, we assume the finite sets are real. Suppose $S \subseteq \mathbb{R}^n$ is a finite set of cardinality k, say,

$$S = \{u_1, \ldots, u_k\}.$$

A function f is a loss function for S if and only if the global minimizers of f are precisely the points of S. In computational practice, we often consider the sum-of-squares loss functions

$$f = p_1^2 + \cdots + p_m^2, \tag{3.1}$$

where each p_i is a polynomial in x. Denote the tuple

$$p = (p_1, \ldots, p_m).$$

Without loss of generality, one can assume that the minimum value of f is zero, up to shifting of a constant. Note that $f(x) = 0$ if and only if $p(x) = 0$. Therefore, f is a loss function for S if and only if

$$S = \{x \in \mathbb{R}^n : p_1(x) = \cdots = p_m(x) = 0\}. \tag{3.2}$$

The above observation gives the following lemma.

Lemma 3.1 *Let S, f be as above. Then f is a loss function for S if and only if S is the real zero set of p, i.e., $S = V_{\mathbb{R}}(p)$.*

The existence of p such that $S = V_{\mathbb{R}}(p)$ is obvious. For instance, one can choose p_i to be a product like

$$(x_{j_1} - (u_1)_{j_1}) \cdot (x_{j_2} - (u_2)_{j_2}) \cdots (x_{j_k} - (u_k)_{j_k}),$$

for all possible $j_1, \ldots, j_k \in \{1, \ldots, n\}$. However, for such a choice of p, each p_i has degree k and f has degree $2k$. The degree is high if the cardinality k is big, and there are n^k such products. This is not practical in applications. In particular, if the set S is approximately given by a sample set of large size, then the resulting p is not convenient for usage. In applications, people prefer loss functions of low degrees.

In the following, we show how to choose a computationally efficient loss function for S. Let $\mathbb{B}_0$ be the set of first k vectors in the nonnegative power set $\mathbb{N}^n$, in the graded lexicographic ordering, i.e.,

$$\mathbb{B}_0 := \Big\{ \underbrace{0,\, e_1,\, \ldots,\, e_n,\, 2e_1,\, e_1 + e_2, \ldots,}_{\text{first k of them}} \Big\}. \tag{3.3}$$

Then, we consider the set

$$\mathbb{B}_1 := \Big((e_1 + \mathbb{B}_0) \cup \cdots \cup (e_n + \mathbb{B}_0) \Big) \setminus \mathbb{B}_0. \tag{3.4}$$

For convenience of notation, denote the monomial vectors

$$[x]_{\mathbb{B}_0} := \big(x^\alpha\big)_{\alpha \in \mathbb{B}_0}, \quad [x]_{\mathbb{B}_1} := \big(x^\alpha\big)_{\alpha \in \mathbb{B}_1}.$$

Since S is a finite set of cardinality k, we wish to select $\mathbb{B}_0$ so that the set of equivalent classes of monomials in $\{x^\beta : \beta \in \mathbb{B}_0\}$ is a basis for the quotient space $\mathbb{R}[x]/I(S)$, where $I(S)$ is the vanishing ideal of S. This requires that x^α ($\alpha \in \mathbb{B}_1$) is a linear combination of monomials x^β ($\beta \in \mathbb{B}_0$), modulo $I(S)$. Equivalently, there exist scalars $G(\beta, \alpha)$ such that

$$\varphi[G, \alpha](x) := x^\alpha - \sum_{\beta \in \mathbb{B}_0} G(\beta, \alpha) x^\beta \equiv 0 \mod I(S) \tag{3.5}$$

for each $\alpha \in \mathbb{B}_1$. Let $G := (G(\beta, \alpha)) \in \mathbb{R}^{\mathbb{B}_0 \times \mathbb{B}_1}$ be the matrix of all such scalars $G(\beta, \alpha)$. The polynomial $\varphi[G, \alpha]$ has coefficients that are linear in entries of G. For convenience, denote that

$$\begin{aligned} \varphi[G] &= \big(\varphi[G,\alpha]\big)_{\alpha\in\mathbb{B}_1}, \\ X_0 &= \big[[u_1]_{\mathbb{B}_0} \quad \cdots \quad [u_k]_{\mathbb{B}_0}\big], \\ X_1 &= \big[[u_1]_{\mathbb{B}_1} \quad \cdots \quad [u_k]_{\mathbb{B}_1}\big]. \end{aligned} \tag{3.6}$$

The X_0 is a square matrix, which is nonsingular if the points in S are in generic positions. For $\varphi[G]$ to vanish on S, the Eq. (3.5) implies that

$$X_1 - G^T X_0 = 0.$$

If X_0 is nonsingular, then the matrix G is given as

$$G = X_0^{-T} X_1^T. \tag{3.7}$$

We look for conditions on G such that $\varphi[G]$ has k common zeros in $\mathbb{C}^n$. For each $i = 1, \ldots, n$, define the multiplication matrix $M_{x_i}(G)$ such that

$$[M_{x_i}(G)]_{\mu,\nu} = \begin{cases} 1 & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_0,\ \mu = \nu + e_i, \\ 0 & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_0,\ \mu \neq \nu + e_i, \\ G(\mu, \nu + e_i) & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_1. \end{cases} \tag{3.8}$$

The rows and columns of $M_{x_i}(G)$ are labelled by monomial powers $\mu, \nu \in \mathbb{B}_0$. The following proposition characterizes when $\varphi[G]$ has k common zeros.

Proposition 3.2 [30, *Proposition* 2.4] *Let* $\mathbb{B}_0$, $\mathbb{B}_1$ *be as in* (3.3)–(3.4). *Then, the polynomial tuple* $\varphi[G]$ *has k common complex zeros* (*counting multiplicities*) *if and only if the multiplication matrices* $M_{x_1}(G), \ldots, M_{x_n}(G)$ *commute, i.e.,*

$$[M_{x_i}(G), M_{x_j}(G)] = 0 \quad (1 \le i < j \le n). \tag{3.9}$$

In particular, $\varphi[G]$ *has k distinct complex zeros if and only if* $M_{x_1}(G), \ldots, M_{x_n}(G)$ *are simultaneously diagonalizable.*

The polynomial tuple $\varphi[G]$ generates the vanishing ideal $I(S)$ of S and $p = \varphi[G]$ has minimum degrees for (3.2) to hold.

Theorem 3.3 *Assume S is a finite set such that* X_0 *is nonsingular. Let G be as in* (3.7). *Then, the ideal* Ideal($\varphi[G]$) *equals the vanishing ideal of S, i.e.,*

$$\text{Ideal}(\varphi[G]) \ = \ \{h \in \mathbb{R}[x] \ : \ h \equiv 0 \ \text{ on } \ S\}. \tag{3.10}$$

In particular, if a polynomial h vanishes on S identically, then there are polynomials p_α $(\alpha \in \mathbb{B}_1)$ *such that*

$$h \ = \ \sum_{\alpha\in\mathbb{B}_1} q_\alpha \varphi[G,\alpha]), \quad \deg(q_\alpha) + |\alpha| \le \deg(h). \tag{3.11}$$

Proof Since X_0 is nonsingular, the set S has k distinct points. Since G is given as in (3.7), the polynomial equation $\varphi[G](x) = 0$ has k distinct solutions. By Proposition 3.2, the multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ are simultaneously diagonalizable. Note that the ideal $\mathrm{Ideal}(\varphi[G])$ is zero-dimensional, because the quotient space $\mathbb{C}[x]/\mathrm{Ideal}(\varphi[G])$ has the dimension k. The ideal $\mathrm{Ideal}(\varphi[G])$ must be radical. This can be implied by Corollary 2.7 of [33]. So (3.10) holds.

Suppose h is a polynomial such that $h \equiv 0$ on S. Then the above shows that $h \in \mathrm{Ideal}(\varphi[G])$. So there exist polynomials q_α ($\alpha \in \mathbb{B}_1$) such that

$$h = \sum_{\alpha \in \mathbb{B}_1} q_\alpha \varphi[G, \alpha].$$

The multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ commute. One can check that the set of polynomials in the tuple $\varphi[G]$ is a Gröbner basis for $\mathrm{Ideal}(\varphi[G])$, with respect to the graded lexicographical ordering. This can also be implied by the proof of Lemma 2.8 in [30]. Therefore, we can further select polynomials $q_\alpha \in \mathbb{R}[x]$ with degree bounds as in (3.11). □

The condition that X_0 is nonsingular holds when the points of S are in generic positions. The Eq. (3.11) shows that the polynomial tuple $\varphi[G]$ is a minimum-degree generating set for the vanishing ideal $I(S)$. The following are some examples.

Example 3.4

(i) Consider the set S in $\mathbb{R}^3$ such that

$$S = \left\{ \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 4 \end{bmatrix} \right\},$$

$$\mathbb{B}_0 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}, \quad \mathbb{B}_1 = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\}.$$

The matrix G as in (3.7) and $\varphi[G]$ are

$$G = \begin{bmatrix} -1 & \frac{11}{3} & 2 & 2 & -\frac{2}{3} \\ 1 & -\frac{1}{3} & 1 & 0 & \frac{10}{3} \end{bmatrix}, \quad \varphi[G] = \begin{bmatrix} x_2 - x_1 + 1 \\ \frac{x_1}{3} + x_3 - \frac{11}{3} \\ x_1^2 - x_1 - 2 \\ x_1 x_2 - 2 \\ x_1 x_3 - \frac{10x_1}{3} + \frac{2}{3} \end{bmatrix}.$$

(ii) Consider the set S in $\mathbb{R}^2$ such that

$$S = \left\{ \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\},$$
$$\mathbb{B}_0 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \quad \mathbb{B}_1 = \left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \right\}.$$

The matrix G as in (3.7) and $\varphi[G]$ are

$$G = \frac{1}{19}\begin{bmatrix} 58 & -14 & 82 \\ 3 & -23 & -20 \\ -12 & -22 & 23 \end{bmatrix}, \quad \varphi[G] = \begin{bmatrix} x_1^2 + \frac{12x_2}{19} - \frac{3x_1}{19} - \frac{58}{19} \\ x_1x_2 + \frac{22x_2}{19} + \frac{23x_1}{19} + \frac{14}{19} \\ x_2^2 - \frac{23x_2}{19} + \frac{20x_1}{19} - \frac{82}{19} \end{bmatrix}.$$

(iii) Consider the set S in $\mathbb{R}^2$ such that

$$S = \left\{ \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ -1 \end{bmatrix} \right\},$$
$$\mathbb{B}_0 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}, \quad \mathbb{B}_1 = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\}.$$

The matrix G in (3.7) and the polynomial vector $\varphi[G]$ are

$$G = \begin{bmatrix} 20 & -5 & -36 & 22 \\ \frac{7}{2} & -\frac{3}{2} & -2 & \frac{9}{2} \\ -7 & 3 & 12 & -5 \\ -\frac{9}{2} & \frac{3}{2} & 9 & -\frac{11}{2} \end{bmatrix}, \quad \varphi[G] = \begin{bmatrix} x_1x_2 + \frac{9x_1^2}{2} + 7x_2 - \frac{7x_1}{2} - 20 \\ x_2^2 - \frac{3x_1^2}{2} - 3x_2 + \frac{3x_1}{2} + 5 \\ x_1^3 - 9x_1^2 - 12x_2 + 2x_1 + 36 \\ x_1^2x_2 + \frac{11x_1^2}{2} + 5x_2 - \frac{9x_1}{2} - 22 \end{bmatrix}.$$

For given S, the polynomial tuple $\varphi[G]$ with G as in (3.7), gives the loss function $f = \|\varphi[G]\|^2$ whose global minimizers are precisely the points in S. However, the loss function f may have spurious minimizers.

Example 3.5 Consider the $S = \left\{ \begin{bmatrix} 5 \\ -2 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix} \right\}$ in $\mathbb{R}^2$. The loss function $f = \|\varphi[G]\|^2$ is

$$f(x) = (x_2 + 5x_1 - 23)^2 + (x_1^2 - 9x_1 + 20)^2 + (x_1x_2 + 22x_1 - 100)^2.$$

Its total gradient ∇f is

$$\begin{bmatrix} 4x_1^3 - 54x_1^2 + 2x_1x_2^2 + 88x_1x_2 + 1260x_1 - 190x_2 - 4990 \\ 2x_2 - 190x_1 + 2x_1^2x_2 + 44x_1^2 - 46 \end{bmatrix}$$

and its Hessian $\nabla^2 f$ is

$$\begin{bmatrix} 12x_1^2 - 108x_1 + 2x_2^2 + 88x_2 + 1260 & 88x_1 + 4x_1x_2 - 190 \\ 88x_1 + 4x_1x_2 - 190 & 2x_1^2 + 2 \end{bmatrix}.$$

By checking the optimality conditions $\nabla f(x) = 0$, $\nabla^2 f(x) \succeq 0$, we get a local minimizer $(-2.2588, -49.7911)$, which is not a global one.

4 Simplicial loss functions

In this section, we study loss functions when S is the vertex set of a standard simplex. For such a case, we show that the loss function $f = \|\varphi[G]\|^2$ has no *spurious minimizers*, i.e., every local minimizer of f is also a global minimizer. Moreover, when S is not the vertex set of a standard simplex, we apply a transformation and get similar loss functions.

4.1 Simplicial loss functions

For a vector $a := (a_1, \ldots, a_n)$, with each scalar $a_i \neq 0$, consider the standard simplex vertex set

$$\Delta_n(a) := \{0,\ a_1e_1, \ldots, a_ne_n\}. \tag{4.1}$$

For the special case that $a = (1, \ldots, 1)$, we denote

$$\Delta_n := \{0,\ e_1, \ldots,\ e_n\}. \tag{4.2}$$

When the dimension n is clear in the context, we just write $\Delta = \Delta_n$ for convenience. In this subsection, we consider the special case that $S = \Delta_n(a)$. Then the monomial power sets $\mathbb{B}_0, \mathbb{B}_1$ are respectively

$$\begin{aligned} \mathbb{B}_0 &= \{0, e_1, \ldots, e_n\}, \\ \mathbb{B}_1 &= \{2e_1, e_1 + e_2,\ \ldots, 2e_n\}. \end{aligned}$$

For the matrix $G \in \mathbb{R}^{\mathbb{B}_0 \times \mathbb{B}_1}$ given as in (3.7), we have that

$$\begin{array}{lll} \varphi[G, 2e_i] & = x_i^2 - a_ix_i & (i \in [n]), \\ \varphi[G, e_i + e_j] & = x_ix_j & (i < j). \end{array} \tag{4.3}$$

The resulting loss function for the set $\Delta_n(a)$ is

$$f(x) = \sum_{i=1}^{n} x_i^2 (x_i - a_i)^2 + \sum_{1 \le i < j \le n} x_i^2 x_j^2. \tag{4.4}$$

In particular, the above loss function for Δ_n is

$$F(x) := \sum_{i=1}^{n} x_i^2 (x_i - 1)^2 + \sum_{1 \le i < j \le n} x_i^2 x_j^2. \tag{4.5}$$

A nice property is that the simplicial loss function as in (4.4) has no spurious minimizers.

Theorem 4.1 *Fix nonzero scalars $a_1, \dots, a_n$, the function f in (4.4) has no spurious minimizers, i.e., every local minimizer of f is also a global minimizer.*

Proof Suppose $z = (z_1, \dots, z_n)$ is a local minimizer of f. Then z satisfies the optimality conditions

$$\nabla f(z) = 0, \quad \nabla^2 f(z) \succeq 0.$$

This implies that for $i = 1, \dots, n$,

$$\frac{\partial f}{\partial x_i}(z) = 2z_i\big(2z_i^2 - 3a_i z_i + (z^T z - z_i^2 + a_i^2)\big) = 0, \tag{4.6}$$

$$\frac{\partial^2 f}{\partial x_i^2}(z) = 12z_i^2 - 12a_i z_i + 2(z^T z - z_i^2 + a_i^2) \ge 0. \tag{4.7}$$

Denote $\delta_i(z) := a_i^2 - 8(z^T z - z_i^2)$. The real solutions for (4.6) are $z_i = 0$ and

$$z_i = \frac{3a_i \pm \sqrt{\delta_i(z)}}{4} \quad \text{if} \quad \delta_i(z) \ge 0. \tag{4.8}$$

If each $z_i = 0$, then $z = 0$ is a global minimizer. Suppose some z_i is nonzero, then it satisfies $\delta_i(z) \ge 0$ and $2z_i^2 - 3a_i z_i + (z^T z - z_i^2 + a_i^2) = 0$. So (4.7) can be reformulated as

$$\frac{\partial^2 f}{\partial x_i^2}(z) = 8z_i^2 - 6a_i z_i = 2z_i(4z_i - 3a_i) \ge 0.$$

Plug (4.8) into the above inequality. Since $\sqrt{\delta_i(z)} \le |a_i| < |3a_i|$ (note $a_i \ne 0$),

$$z_i = \begin{cases} \frac{3a_i - \sqrt{\delta_i(z)}}{4} & \text{if } a_i < 0, \\ \frac{3a_i + \sqrt{\delta_i(z)}}{4} & \text{if } a_i > 0. \end{cases}$$

It is clear that $|z_i| \geq |3a_i/4|$. If z_i is the only nonzero entry of z, then $\sqrt{\delta_i(z)} = |a_i|$ and $z = a_i e_i$, which is a global minimizer. Suppose z has another nonzero entry z_j. By a similar argument, we can get $\delta_j(z) \geq 0$ and $|z_j| \geq |3a_j/4|$. Note that $2a_i^2 - 9a_j^2 \geq 0$ since

$$a_i^2 - 8 \cdot \left|\frac{3a_j}{4}\right|^2 \geq a_i^2 - 8z_j^2 \geq \delta_i(z) \geq 0.$$

Similarly, $2a_j^2 - 9a_i^2 \geq 0$, so

$$2a_j^2 - 9a_i^2 \geq 2a_j^2 - 9 \cdot \frac{9}{2}a_j^2 = -\frac{77}{2}a_j^2 \geq 0.$$

The above holds if and only if $a_j = 0$, which contradicts that all $a_1, \ldots, a_n$ are nonzero. Therefore, every local minimizer of f is a global minimizer, i.e., f has no spurious minimizers. □

4.2 Transformation for general sets

When S is not a simplicial vertex set, we can still use the function F in (4.5) to get new loss functions, up to a transformation. These new functions have no spurious minimizers. They are called *transformed* simplicial loss functions. Consider that S is given as

$$S = \{u_1, \ldots, u_k\}. \tag{4.9}$$

We discuss the transformation for two different cases.

4.3 Case I: $k \leq n+1$

Consider the vertex set of a standard simplex set in $\mathbb{R}^{k-1}$

$$\Delta_{k-1} = \{0, e_1, \ldots, e_{k-1}\}.$$

The loss function as in (4.5) for Δ_{k-1} is

$$F_{k-1}(z) := \sum_{i=1}^{k-1} z_i^2(z_i - 1)^2 + \sum_{1 \leq i < j \leq k-1} z_i^2 z_j^2, \tag{4.10}$$

in the variable $z = (z_1, \ldots, z_{k-1})$. Consider the linear map

$$\ell : \mathbb{R}^{k-1} \to \mathbb{R}^n, \quad \ell(e_i) = u_i - u_k, \ i = 1, \ldots, k-1. \tag{4.11}$$

The representing matrix for the linear map ℓ is

$$U = \begin{bmatrix} u_1 - u_k & \cdots & u_{k-1} - u_k \end{bmatrix}. \tag{4.12}$$

When $u_1, \ldots, u_k$ are in generic positions, the matrix U has full column rank. Let

$$U^\dagger := (U^T U)^{-1} U^T$$

be the Pseudo inverse of U. For $x = (x_1, \ldots, x_n)$, consider the loss function

$$f(x) = F_{k-1}\big(U^\dagger (x - u_k)\big). \tag{4.13}$$

Recall that $\mathrm{Null}(U^\dagger)$ denotes the null space of the matrix $U^\dagger$.

Theorem 4.2 *Suppose* $k \le n+1$ *and* $\mathrm{rank}\ U = k-1$. *Then, the function* f *as in* (4.13) *is a loss function for the set*

$$S + \mathrm{Null}(U^\dagger) := \{x + y : x \in S, U^\dagger y = 0\}.$$

Moreover, f *has no spurious minimizers.*

Proof The function f as in (4.13) is nonnegative everywhere. Note that $f(x) = 0$ if and only if $U^\dagger(x - u_k) \in \Delta_{k-1}$. It holds that

$$\Delta_{k-1} = \{U^\dagger (x - u_k) : x \in S\}.$$

For $x \in \mathbb{R}^n$, we have $U^\dagger(x - u_k) \in \Delta_{k-1}$ if and only if $x \in S + \mathrm{Null}(U^\dagger)$. This shows that f is a loss function for $S + \mathrm{Null}(U^\dagger)$ in $\mathbb{R}^n$.

The gradient and Hessian of f can be written as

$$\nabla_x f(x) = (U^\dagger)^T \nabla_z F_{k-1}(z), \quad \nabla_x^2 f(x) = (U^\dagger)^T \nabla_z^2 F_{k-1}(z) U^\dagger.$$

Note that $U^\dagger$ has full row rank. If u is a local minimizer of f, then $\nabla_x f(u) = 0$, $\nabla_x^2 f(u) \succeq 0$. Let $z = U^\dagger(u - u_k)$, then the above implies that

$$\nabla_z F_{k-1}(z) = 0, \quad \nabla_z^2 F_{k-1}(z) \succeq 0.$$

As in the proof of Theorem 4.1, one can show that $z \in \Delta_{k-1}$. This implies that z is a global minimizer of F_{k-1} and hence u is a global minimizer of f. So f has no spurious minimizers. □

4.4 Case II: $k > n+1$

Let $\omega : \mathbb{R}^n \to \mathbb{R}^{k-1}$ be the monomial function such that

$$[x]_{\mathbb{B}_0} = \begin{bmatrix} 1 \\ \omega(x) \end{bmatrix}, \tag{4.14}$$

where $\mathbb{B}_0$ is the power set in (3.3). For the set S as in (4.9), denote

$$\hat{S} := \left\{ \omega(u_1), \ldots, \omega(u_k) \right\} \subseteq \mathbb{R}^{k-1}. \tag{4.15}$$

Define the linear map $\mathcal{L}$ such that

$$\mathcal{L} : \mathbb{R}^{k-1} \to \mathbb{R}^{k-1}, \quad \mathcal{L}(e_i) = \omega(u_i) - \omega(u_k),\ i = 1, \ldots, k-1.$$

The representing matrix for the linear map $\mathcal{L}$ is

$$L = \left[\omega(u_1) \ \cdots \ \omega(u_{k-1}) \right] - \left[\omega(u_k) \ \cdots \ \omega(u_k) \right]. \tag{4.16}$$

When $u_1, \ldots, u_n$ are in generic positions, the matrix L is nonsingular. For such a case, define the function

$$\hat{f}(z) := F_{k-1}\left(L^{-1}(z - \omega(u_k)) \right), \tag{4.17}$$

in the $z = (z_1, \ldots, z_{k-1})$, where F_{k-1} is the simplicial loss function as in (4.10). The above $\hat{f}$ is called a transformed simplicial loss function for $\hat{S}$. The following theorem follows from Theorem 4.2.

Theorem 4.3 *Suppose $k > n+1$ and L is nonsingular. Then, the function $\hat{f}$ as in (4.17) is a loss function for $\hat{S}$ and it has no spurious minimizers.*

For $x = (x_1, \ldots, x_n)$, define the function

$$f(x) = F_{k-1}\left(L^{-1}(\omega(x) - \omega(u_k)) \right). \tag{4.18}$$

Corollary 4.4 *Suppose $k > n+1$ and L in (4.16) is nonsingular, then the function f in (4.18) is a loss function for S.*

Proof The function f as in (4.18) is nonnegative everywhere. By Theorem 4.3, we know $f(x) = 0$ if and only if $\omega(x) \in \hat{S}$. Since ω is a one-to-one map, the f is a loss function for S. □

The transformed simplicial loss functions in (4.13) and (4.17) have no spurious minimizers. The following are some examples of transformed simplicial loss functions.

Example 4.5

(i) Consider the set S in $\mathbb{R}^3$ such that

$$S = \left\{ \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ -5 \end{bmatrix} \right\}.$$

The matrix U as in (4.12) and its Pseudo inverse are

$$U = \begin{bmatrix} 5 \\ -5 \\ 6 \end{bmatrix}, \quad U^{\dagger} = \frac{1}{86} \begin{bmatrix} 5 \\ -5 \\ 6 \end{bmatrix}^T.$$

Since $k = 2$, the simplicial loss function for Δ_{k-1} is $F_1 = z^2(z-1)^2$ in the univariate variable z. Then, the transformed simplicial loss function as in (4.13) is

$$f(x) = \left(\frac{5x_1}{86} - \frac{5x_2}{86} + \frac{3x_3}{43} + \frac{25}{43} \right)^2 \cdot \left(\frac{5x_1}{86} - \frac{5x_2}{86} + \frac{3x_3}{43} - \frac{18}{43} \right)^2.$$

(ii) Consider the set S in $\mathbb{R}^2$ such that

$$S = \left\{ \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right\}.$$

Since $k = 4 > n + 1$, the set $\hat{S}$ in (4.15) is

$$\hat{S} = \left\{ \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \\ 4 \end{bmatrix} \right\}.$$

The matrix L as in (4.16) and its inverse are

$$L = \begin{bmatrix} 4 & 1 & 3 \\ 1 & -4 & -5 \\ 0 & -3 & -3 \end{bmatrix}, \quad L^{-1} = \frac{1}{18} \begin{bmatrix} 3 & 6 & -7 \\ -3 & 12 & -23 \\ 3 & -12 & 17 \end{bmatrix}.$$

Since $k = 4$, the simplicial loss function for Δ_{k-1} is

$$F_3(z) = z_1^2(z_1 - 1)^2 + z_1^2 z_2^2 + z_2^2(z_2 - 1)^2 + z_2^2 z_3^2 + z_3^2(z_3 - 1)^2.$$

in the variable $z = (z_1, z_2, z_3)$. Then, the transformed simplicial loss function as in (4.17) is $\hat{f}(z) = F_3(L^{-1}(z - \omega(u_4))$, with

$$L^{-1}(z - \omega(u_4)) = \frac{1}{18} \begin{bmatrix} 3z_1 + 6z_2 - 7z_3 + 22 \\ -3z_1 + 12z_2 - 23z_3 + 62 \\ 3z_1 - 12z_2 + 17z_3 - 38 \end{bmatrix}.$$

5 Finite sets with noises

In this section, we study loss functions for finite sets that are given with noises. In many applications, the finite set S, with the cardinality k, is often approximately given by another finite set T, with the cardinality $N \gg k$. For instance, each point of

S is often approximated by a number of samplings, and T consists of all such samplings. The cardinality N is the total number of samplings. We look for good loss functions for such approximately given sets. This kind of questions have important applications in clustering and classification.

5.1 Best approximation sets

Suppose S is approximately given by a sampling set T, say,

$$T = \{v_1, \ldots, v_N\}. \tag{5.1}$$

Each point of S is sampled by a certain number of points in T. We discuss how to recover the k points of S from sampling points in T.

A finite set can be represented as the optimizer set of a loss function. For convenience, we consider loss functions whose minimum values are zeros. Let $\mathcal{F}$ be a family of loss functions such that each $f \in \mathcal{F}$ has k common zeros. The loss function family $\mathcal{F}$ is parameterized by some parameters. For such given $\mathcal{F}$, we look for the best loss function in $\mathcal{F}$ such that its average value on T is the smallest. This leads to the following definition.

Definition 5.1 Let $\mathcal{F}$ be a family of loss functions such that each $f \in \mathcal{F}$ is nonnegative and it has k common zeros. A set $S^* = \{u_1^*, \ldots, u_k^*\}$ is called the best $\mathcal{F}$-approximation set for T as in (5.1) if S^* is the zero set of f^*, where f^* is the minimizer of the optimization

$$\begin{cases} \min \ \mu(f) := \frac{1}{N} \sum_{i=1}^{N} f(v_i) \\ \text{s.t. } f \in \mathcal{F}. \end{cases} \tag{5.2}$$

For a given set S, if the matrix G is as in (3.7), then S is the common zero set of the polynomial tuple $\varphi[G]$, given as in (3.5). In fact, Ideal$(\varphi[G])$ is the vanishing ideal $I(S)$ and $\varphi[G]$ gives the minimum-degree generating set for $I(S)$. The relation between S and $\varphi[G]$ is characterized by Theorem 3.3. As shown in Proposition 3.2, $\varphi[G]$ has k common zeros (counting multiplicities and all complex ones) if and only if the multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ commute with each other. Moreover, $\varphi[G]$ has k distinct zeros if and only if $M_{x_1}(G), \ldots, M_{x_n}(G)$ are simultaneously diagonalizable. So, one can use the matrix G and the polynomial tuple $\varphi[G]$ to represent the finite set S. As in Sect. 3, we consider the family of the following loss functions

$$f_G := \|\varphi[G]\|^2, \tag{5.3}$$

parameterized by G. We look for the matrix G such that the average of the values of f_G on T is minimum and $\varphi[G]$ has k common zeros.

In view of the above, we consider the following matrix optimization problem

$$\begin{cases} \min\ \vartheta(G) := \frac{1}{N}\sum_{j=1}^{N} f_G(v_j) \\ \text{s.t. } [M_{x_i}(G), M_{x_j}(G)] = 0\,(1 \le i < j \le n). \end{cases} \tag{5.4}$$

The value $\varphi[G](v_i)$ is linear in the matrix G. The feasible set of (5.4) is given by a set of quadratic equations. The optimization (5.4) is the specialization of (5.2) such that $\mathcal{F}$ is the family of loss function f_G, with $\varphi[G]$ having k common zeros.

5.2 Approximation analysis

Suppose G^* is the minimizer of (5.4). Let S_0 denote the common zero set of $\varphi[G^*]$. We can use S_0 to approximate the points in S. In some applications, the set S contains only real points and people like to get a real set approximation for S.

First, we study the approximation quality of the optimization (5.4). For each $\alpha \in \mathbb{B}_1$, the sub-Hessian of the objective $\vartheta(G)$ with respect to the αth column $G(:,\alpha)$ is the matrix

$$H := \frac{2}{N}\sum_{j=1}^{N} [v_j]_{\mathbb{B}_0}([v_j]_{\mathbb{B}_0})^{\mathtt{H}}.$$

In the above, the superscript $^{\mathtt{H}}$ denotes the Hermitian transpose.

Theorem 5.2 *Let T be as in (5.1) and let $S = \{u_1, \ldots, u_k\}$ be such that the matrix X_0 as in (3.6) is nonsingular. Assume there exists $\delta > 0$ such that $H \succeq 2\delta I_k$. Suppose the set T is such that*

$$T \subseteq S + B(0,\epsilon), \quad T \cap B(u_i,\epsilon) \neq \emptyset\,(i = 1, \ldots, k), \tag{5.5}$$

for some $\epsilon > 0$. Then, as $\epsilon \to 0$, the optimizer G^* of (5.4) converges to $\hat{G} := X_0^{-T} X_1^T$, and the common zero set S_0 of $\varphi[G^*]$ converges to S.

In particular, when $S, T \subseteq \mathbb{R}^n$, if $\epsilon > 0$ is sufficiently small, the common zero set S_0 contains k distinct real points.

Proof First, we show the convergence $G^* \to \hat{G}$ as $\epsilon \to 0$. Since the set $\hat{B} := \cup_{i=1}^{k} B(u_i, 1)$ is compact, the polynomial function $\varphi[\hat{G}](x)$ is Lipschitz continuous on $\hat{B}$. There exists $R > 0$ such that for all $i \in [k]$ and for all $x \in B(u_i, \epsilon)$,

$$\|\varphi[\hat{G}](x) - \varphi[\hat{G}](u_i)\| \le R\|x - u_i\| \le R\epsilon.$$

Since $T \subseteq S + B(0,\epsilon)$, each $v_j \in T$ belongs to some $B(u_{i_j}, \epsilon)$ for $i_j \in \{1, \ldots, k\}$. So the above inequality implies that (note that each $\varphi[\hat{G}](u_{i_j}) = 0$)

$$\begin{aligned}\vartheta(\hat{G}) &= \frac{1}{N}\sum_{j=1}^{N} \|\varphi[\hat{G}](v_j)\|^2 \\ &= \frac{1}{N}\sum_{j=1}^{N} \|\varphi[\hat{G}](v_j) - \varphi[\hat{G}](u_{i_j})\|^2 \le (R\epsilon)^2.\end{aligned}$$

Since G^* is the minimizer of (5.4), we have

$$0 \le \vartheta(G^*) \le \vartheta(\hat{G}) \le (R\epsilon)^2. \tag{5.6}$$

Moreover, it holds that

$$\begin{aligned}\vartheta(G^*) &= \frac{1}{N}\sum_{j=1}^{N} \|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j) + \varphi[\hat{G}](v_j)\|^2, \\ &\ge \frac{1}{N}\sum_{j=1}^{N} \Big(\|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| - \|\varphi[\hat{G}](v_j)\|\Big)^2 \\ &\ge \frac{1}{N^2}\Big(\sum_{j=1}^{N} \|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| - \sum_{j=1}^{N} \|\varphi[\hat{G}](v_j)\|\Big)^2.\end{aligned}$$

In the above, the first inequality follows from that $\|a+b\|^2 \ge (\|a\| - \|b\|)^2$ and the second inequality follows from the Cauchy-Schwartz inequality. Then, we have

$$\sum_{j=1}^{N} \|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| \le N\sqrt{\vartheta(G^*)} + \sum_{j=1}^{N} \|\varphi[\hat{G}](v_j)\|$$

By the formula of $\varphi[G](x)$ and using Cauchy-Schwartz inequality again, we get

$$\sum_{j=1}^{N} \|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\| \le N\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big).$$

Since $\sum_{j=1}^{N} \|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|^2 \le \big(\sum_{j=1}^{N} \|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|\big)^2$, we have

$$\frac{1}{N}\sum_{j=1}^{N} \|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|^2 \le N\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big)^2.$$

By the assumption $H \succeq 2\delta I_k$, the above implies

$$\|G^* - \hat{G}\| \le \sqrt{\frac{N}{\delta}}\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big).$$

Therefore, as $\epsilon \to 0$, we have G^* converges to $\hat{G}$.

In the following, we assume that $S, T \subseteq \mathbb{R}^n$. Since X_0 is nonsingular, S has k distinct real points. Recall the multiplication matrices $M_{x_i}(G^*), M_{x_i}(\hat{G})$ given as in (3.8). Since $G^* \to \hat{G}$, the common zero set of $\varphi[G^*]$ converges to that of $\varphi[\hat{G}]$. The zero set

of $\varphi[\hat{G}]$ is S, which consists of k distinct real points. Hence, $\varphi[G^*]$ also has k distinct common zeros when $\epsilon > 0$ is sufficiently small. Then it remains for us to show that all common zeros of $\varphi[G^*]$ are real. For a vector $\xi = (\xi_1, \ldots, \xi_n)$, define the matrices

$$M_1 = \sum_{i=1}^{n} \xi_i M_{x_i}(G^*), \quad M_2 = \sum_{i=1}^{n} \xi_i M_{x_i}(\hat{G}).$$

Their characteristic polynomials are

$$p_1(\lambda) := \det(M_1 - \lambda I), \quad p_2(\lambda) := \det(M_2 - \lambda I).$$

Fix a generic real value for ξ so that M_2 has k distinct real eigenvalues. This is because $\varphi[\hat{G}](x)$ has real distinct solutions and by the Stickelberger's Theorem (see (5.8) as in [23, 33]). Note that both $p_1(\lambda)$, $p_2(\lambda)$ have degree k and all coefficients are real. The $p_2(\lambda)$ has k distinct real roots. They are ordered as

$$\hat{\lambda}_1 < \hat{\lambda}_2 < \cdots < \hat{\lambda}_k.$$

We can choose real scalars $b_0, \ldots, b_k$ such that

$$b_0 < \hat{\lambda}_1 < b_1 < \cdots < b_{k-1} < \hat{\lambda}_k < b_k.$$

As $\epsilon \to 0$, the coefficients of p_1 converge to those of p_2. So, when $\epsilon > 0$ is small enough, $p_1(b_j)$ has the same sign as $p_2(b_j)$ does. Since each $p_2(b_{j-1})p_2(b_j) < 0$, we have

$$p_1(b_{j-1})p_1(b_j) < 0, \quad j = 1, \ldots, k+1.$$

This implies that p_1 has k distinct real roots. Equivalently, M_1 has k distinct real eigenvalues for $\epsilon > 0$ sufficiently small. By Proposition 3.2, the multiplication matrices $M_{x_1}(G^*), \ldots, M_{x_n}(G^*)$ are simultaneously diagonalizable. Also note that M_1 is diagonalizable and there is a unique real eigenvector (up to scaling) for each real eigenvalue. This shows that $M_{x_1}(G^*), \ldots, M_{x_n}(G^*)$ can be simultaneously diagonalized by common real eigenvectors. All $M_{x_1}(G^*), \ldots, M_{x_n}(G^*)$ have real entries, so they have only real eigenvalues. Therefore, by Stickelberger's Theorem, $\varphi[G^*]$ has k distinct real common zeros if $\epsilon > 0$ is sufficiently small. □

5.3 Loss functions for noisy sets

When the set S is approximately given by the sampling set T, we can solve (5.4) for an optimizer matrix G^*, to get loss functions. Let S_0 be the common zero set of the polynomial tuple $\varphi[G^*]$. If T is far from S, S_0 may have non-real points. If real points are wanted, we can choose the real part set

$$S^{re} := \{ \text{Re}\,(u) : u \in S_0 \}. \tag{5.7}$$

First, we show how to compute the common zero set S_0. By Stickelberger's Theorem (see [23, 33]), the set S_0 can be expressed as

$$S_0 = \left\{ (\lambda_1, \ldots, \lambda_n) \,\middle|\, \begin{array}{l} \exists q \in \mathbb{C}^k \setminus \{0\} \text{ such that} \\ M_{x_i}(G^*)q = \lambda_i q,\ i = 1, \ldots, n \end{array} \right\}. \tag{5.8}$$

To get S_0 numerically, people often use Schur decompositions. Let

$$M_1 = \xi_1 M_{x_1}(G^*) + \cdots + \xi_n M_{x_n}(G^*), \tag{5.9}$$

where $\xi_1, \ldots, \xi_n$ are generically chosen scalars. Then, compute the Schur decomposition for M_1:

$$Q^{\mathrm{H}} M_1 Q = P, \quad Q = \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}. \tag{5.10}$$

In the above, $Q \in \mathbb{C}^{k\times k}$ is a unitary matrix and $P \in \mathbb{C}^{k\times k}$ is upper triangular. Based on the Schur decomposition (5.10), the common zeros $\hat{u}_1, \ldots, \hat{u}_k$ of $\varphi[G^*]$ can be given as

$$\hat{u}_i := \left(q_i^{\mathrm{H}} M_{x_1}(G^*) q_i, \ldots, q_i^{\mathrm{H}} M_{x_n}(G^*) q_i\right), \quad i = 1, \ldots, k. \tag{5.11}$$

We refer to Corless et al. [6] for how to use Schur decompositions to compute common zeros of zero-dimensional polynomial systems. For general cases, the set S_0 contains k distinct points. It holds when $S, T \subseteq \mathbb{R}^n$ and the points in T are close to S; see Theorem 5.2.

Based on the above discussions, we get the following algorithm for obtaining loss functions when S is approximately given by the sampling set T.

Algorithm 1 For the given set T as in (5.1) and the cardinality k, do the following:

Step 1 Solve quadratic optimization (5.4) for the optimizer G^*.
Step 2 Compute the common zero set $S_0 = \{\hat{u}_1, \ldots, \hat{u}_k\}$ of $\varphi[G^*]$. Let S^* be the set S_0 or S^{re} be as in (5.7) if the real points are wanted.
Step 3 Get a loss function for the set S^*, by the method in Sect. 3 or Sect. 4.

In Step 1, the optimization (5.4) has a convex quadratic objective, but its constraints are given by quadratic equations, in the matrix variable G. So (5.4) is a quadratically constrained quadratic program (QCQP). It can be solved as a polynomial optimization problem (e.g., by the software `GloptiPoly 3` [11]). The classical nonlinear optimization methods, (e.g., Gauss-Newton, trust region, and Levenberg-Marquardt type methods) can also be applied to solve (5.4). We refer to Kelley [16], More [27] and Yuan [39] for such references.

In Step 2, the common zero set S_0 can be computed as in (5.11), by using the Schur decomposition (5.10) for the matrix M_1 in (5.9), for generically chosen scalars $\xi_1, \ldots, \xi_n$.

In Step 3, there are two options for obtaining loss functions for the set S^*, given in Sects. 3 and 4 respectively. One is to choose $f = \|\varphi[G]\|^2$; the other one is to apply

Table 1 The numerical results of Example 6.1

N_i	ϵ	$\|S - S^*\|$	$\max_{u \in S} f(u)$
50	0.05	0.0064	1.27×10^{-4}
	0.1	0.0145	2.98×10^{-4}
	0.5	0.1821	0.0862
100	0.05	0.0055	8.06×10^{-5}
	0.1	0.0067	1.89×10^{-4}
	0.5	0.1080	0.0359

a transformation first and then choose f similarly. After the transformation, there are no spurious optimizers for the loss function.

6 Numerical experiments

In this section, we present numerical experiments for loss functions. The computation is implemented in `MATLAB` R2018a, in a Laptop with CPU 8th Generation Intel® Core^{TM} i5-8250U and RAM 16 GB. The optimization problem (5.4) can be solved by the polynomial optimization software `GloptiPoly 3` (with the SDP solver `SeDuMi` [34]), or it can be solved by classical nonlinear optimization solvers (e.g., the `MATLAB` function `fmincon` can be used for convenience).

First, we explore the numerical performance of Algorithm 1.

Example 6.1 Consider the set

$$S = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 1.5 \\ 2.5 \end{bmatrix}, \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix} \right\}.$$

Suppose T is a sampling set of S such that

$$T \subseteq S + \epsilon[-1, 1]^2, \quad \text{and}$$
$$|T \cap \{u_i + \epsilon[-1, 1]^2\}| = N_i \, (i = 1, \ldots, 6).$$

We apply Algorithm 1 for cases $N_i \in \{50, 100\}$ and $\epsilon \in \{0.05, 0.1, 0.5\}$. The samples are generated with `MATLAB` function `randn`. We summarize the computational results in Table 1 and Fig. 1. In Table 1, the symbol S^* denotes the computed approximation set as in (5.7). We use the distance

$$\|S - S^*\| := \max_{v \in S^*} \min_{u_i \in S} \|v - u_i\|$$

to measure the approximation quality of S^* to S. The loss function for S^* is in form of $f = \|\varphi[G]\|^2$, whose maximum value on S is shown in the fourth column. In Fig. 1,

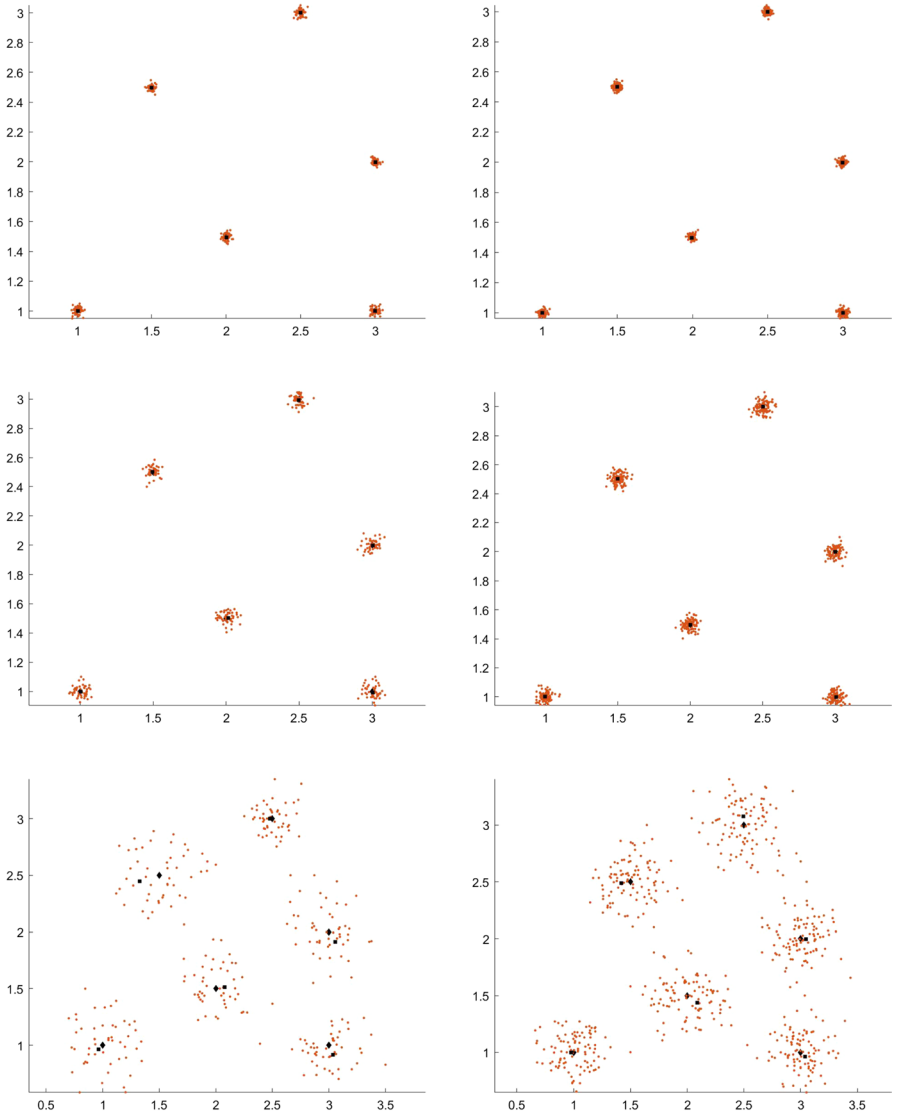

Fig. 1 The performance of Algorithm 1 for Example 6.1. The left column is for $N_i = 50$, and the right column is for $N_i = 100$. The first row is for $\epsilon = 0.05$, the second row is for $\epsilon = 0.1$, and the third row is for $\epsilon = 0.5$

the sampling points in T are plotted in dots, the points in S are plotted in diamonds and the points in S^* are plotted in squares. The left column from top to bottom shows cases for $N_i = 50$ and $\epsilon = 0.05, 0.1, 0.5$ respectively. The right column shows cases for $N_i = 100$ accordingly.

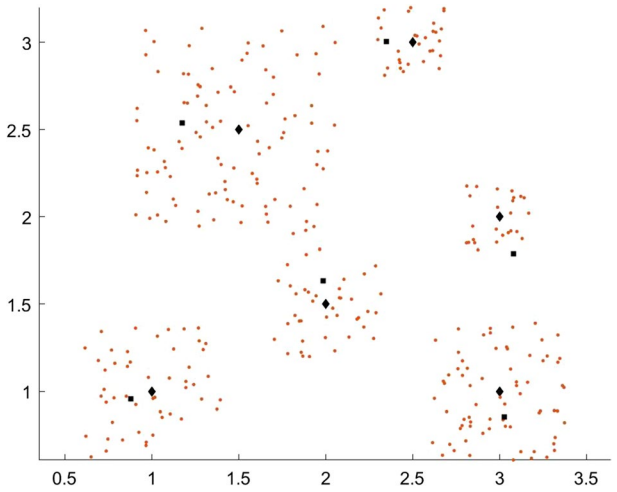

Fig. 2 The performance of Algorithm 1 for Example 6.2

We explore the performance of Algorithm 1 for sampling sets T that are not evenly distributed around S.

Example 6.2 Let S be the same set given as in Example 6.1. Suppose T is a sampling set of S such that for each $i = 1, \ldots, 6$,

$$T \subseteq S + a_i[-1, 1]^2, \quad |T \cap \{u_i + a_i[-1, 1]^2\}| = b_i,$$

where $a = (a_1, \ldots, a_6)$ and $b = (b_1, \ldots, b_6)$ are given as

$$a = (0.4,\ 0.2,\ 0.6,\ 0.2,\ 0.32,\ 0.4),$$
$$b = (50,\ 25,\ 100,\ 30,\ 40,\ 70).$$

We apply Algorithm 1 for samples generated with the `MATLAB` function `randn`. The computational results are summarized as follows. The computed approximation set is

$$S^* = \left\{ \begin{bmatrix} 0.8820 \\ 0.9557 \end{bmatrix}, \begin{bmatrix} 3.0807 \\ 1.7892 \end{bmatrix}, \begin{bmatrix} 1.1759 \\ 2.5383 \end{bmatrix}, \begin{bmatrix} 2.3481 \\ 3.0050 \end{bmatrix}, \begin{bmatrix} 1.9854 \\ 1.6354 \end{bmatrix}, \begin{bmatrix} 3.0292 \\ 0.8541 \end{bmatrix} \right\}.$$

We have that

$$\|S - S^*\| = 0.3264, \quad \max_{u \in S} f(u) = 0.2147,$$

where $f(x) = \|\varphi[G](x)\|^2$ is the loss function for S^*. The visualization of Example 6.2 is given in Fig. 2, where the points in S are plotted in diamonds and the points in S^* are plotted in squares.

Table 2 The computational results for Example 6.3

n	k	Accuracy rate		CPU time	
		Diagonal (%)	Non-diagonal (%)	Diagonal	Non-diagonal
4	4	77.66	85.34	66.14	68.28
	5	88.73	98.92	93.32	90.76
5	3	80.93	84.04	73.35	75.25
	4	82.40	89.58	132.88	129.19

Then, we apply loss functions to study Gaussian mixture models. For a given sampling set T, we compute the finite set S^* and its loss function by Algorithm 1. The loss function in Sect. 4 are used, so there are no spurious minimizers. For a point $v \in T$, apply a nonlinear optimization method (we use `MATLAB` function `fminunc`) to minimize f with the starting point v. Once a minimizer u is returned, we cluster v to the group labeled by the point $u \in S^*$.

Example 6.3 We use Algorithm 1 and the transformed simplicial loss functions in Sect. 4 to learn Gaussian mixture models (GMMs). Each GMM has parameters (w_i, μ_i, Σ_i), $i = 1, \ldots, k$, where each weight $w_i > 0$, the mean vector $\mu_i \in \mathbb{R}^n$ and the covariance matrix $\Sigma_i \in \mathcal{S}^n_{++}$ (the cone of real symmetric positive definite n-by-n matrices), such that $w_1 + \cdots + w_k = 1$. We explore the performance of transformed simplicial loss functions for two cases

$$\text{I)}: \ n = 4, \ k \in \{4, 5\}, \quad \text{II)}: \ n = 5, \ k \in \{3, 4\}.$$

In particular, we compare the results for diagonal Gaussian mixture models (each Σ_i is diagonal) and non-diagonal Gaussian mixture models (each Σ_i is non-diagonal). For each instance, 1000 samples are generated. The weights $w_1, \ldots, w_k$ are also computed from sampling: we first use the `MATLAB` command `randi` getting 1000 integers from $[k]$, and then counting each w_i based on the occurrence probability of $i \in [k]$. We generate each covariance matrix as $\Sigma_i = R^T R$, for some randomly generated square matrix R. The clustering accuracy rate counts the percentage of samples belonging to the correct cluster. We run 10 instances for each case and give the average CPU time (in seconds) consumed by the method and the accuracy rate for all instances. The computational results are reported in Table 2. Algorithm 1 together with transformed simplicial loss functions has good performance for both diagonal and non-diagonal Gaussian mixture models. The clustering accuracy rate is higher for non-diagonal Gaussian mixtures than that for diagonal ones. In particular, for $(n, k) = (4, 5)$, the clustering accuracy rate can be as high as 98.92%.

7 Conclusions

This paper studies loss functions for finite sets. We give a framework for loss functions. For a generic finite set S, we show that S can be equivalently given as the zero set of SOS polynomials with minimum degrees. When S is the vertex set of a standard simplex, we show that the given loss function has no spurious minimizers. For general finite sets, after a transformation, we can get similar loss functions that have no spurious minimizers. When S is approximately given by a sampling set T, we show how to get loss functions for S based on sampling points in T. This can be done by solving a quadratic optimization problem. Some examples are given to show the efficiency of the proposed loss functions.

Funding The authors are partially supported by the NSF Grant DMS-2110780.

Data availability statement We do not analyse or generate any datasets, because our work proceeds within a theoretical and mathematical approach.

Declarations

Conflict of interest They have no competing interests to declare that are relevant to the content of this article.

References

1. Babbush, R., Denchev, V., Ding, N., et al.: Construction of non-convex polynomial loss functions for training a binary classifier with quantum annealing. Preprint (2014). arXiv:1406.4203
2. Barron, J.T.: A general and adaptive robust loss function. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
3. Beyhaghi, P., Alimo, R., Bewley, T.: A derivative-free optimization algorithm for the efficient minimization of functions obtained via statistical averaging. Comput. Optim. Appl. **76**(1), 1–31 (2020)
4. Cheng, D., Gong, Y., Zhou, S. et al.: Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
5. Christoffersen, P., Jacobs, K.: The importance of the loss function in option valuation. J. Financ. Econ. **72**(2), 291–318 (2004)
6. Corless, R.M., Gianni, P.M., Trager, B.M.: A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii, pp. 133–140 (1977)
7. Cox, D., Little, J., OShea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer (2013)
8. Fan, J., Nie, J., Zhou, A.: Tensor eigenvalue complementarity problems. Math. Program. **170**(2), 507–539 (2018)
9. Gonzalez, S., Miikkulainen, R.: Optimizing loss functions through multi-variate Taylor polynomial parameterization. In: Proceedings of the Genetic and Evolutionary Computation Conference (2021)
10. Guo, B., Nie, J., Yang, Z.: Learning diagonal Gaussian mixture models and incomplete tensor decompositions. Vietnam J. Math. **50**(2), 421–446 (2022)
11. Henrion, D., Lasserre, J., Lofberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**, 761–779 (2009)

12. Huber, P.J.: Robust estimation of a location parameter. In: Kotz S., Johnson N.L. (eds.) Breakthroughs in Statistics. Springer Series in Statistics (Perspectives in Statistics). Springer, New York (1992).https://doi.org/10.1007/978-1-4612-4380-9_35
13. Ichihara, H.: Optimal control for polynomial systems using matrix sum of squares relaxations. IEEE Trans. Autom. Control **54**(5), 1048–1053 (2009)
14. Ito, Y., Fujimoto, K.: On optimal control with polynomial cost functions for linear systems with time-invariant stochastic parameters. In: American Control Conference (ACC). IEEE (2021)
15. Jagerman, D.L.: Some properties of the Erlang loss function. Bell Syst. Tech. J. **53**(3), 525–551 (1974)
16. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations, Frontiers in Applied Mathematics, vol. 16. SIAM, Philadelphia (1995)
17. Ko, Y.H., Kim, K.J., Jun, C.H.: A new loss function-based method for multiresponse optimization. J. Qual. Technol. **37**(1), 50–59 (2005)
18. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**, 796–817 (2001)
19. Lasserre, J.B.: An Introduction to Polynomial and Semi-Algebraic Optimization. Cambridge University Press (2015)
20. Lasserre, J.B.: The moment-SOS hierarchy. In: Sirakov, B., Ney de Souza, P., Viana, M. (eds.) Proceedings of the International Congress of Mathematicians (ICM 2018), vol. 3, pp. 3761–3784. World Scientific (2019)
21. Laszka, A., Szeszlér, D., Buttyán, L.: Linear loss function for the network blocking game: an efficient model for measuring network robustness and link criticality. In: International Conference on Decision and Game Theory for Security. Springer, Berlin, Heidelberg (2012)
22. Lasserre, J.B.: Homogeneous polynomials and spurious local minima on the unit sphere. Optim. Lett. (2021). https://doi.org/10.1007/s11590-021-01811-3
23. Laurent, M.: Sums of squares, moment matrices and optimization over polynomials. In: Emerging Applications of Algebraic Geometry of IMA Volumes in Mathematics and its Applications, vol. 149, pp. 157–270. Springer (2009)
24. Laurent, M.: Optimization over polynomials: selected topics. In: Jang, S.Y., Kim, Y.R., Lee, D.-W., Yie, I. (eds.) Proceedings of the International Congress of Mathematicians, pp. 843–869 (2014)
25. Leung, B.P.K., Spiring, F.A.: The inverted beta loss function: properties and applications. IIE Trans. **34**(12), 1101–1109 (2002)
26. Li, Z., Cai, J., Wei, K.: Toward the optimal construction of a loss function without spurious local minima for solving quadratic equations. IEEE Trans. Inf. Theory **66**(5), 3242–3260 (2019)
27. More, J.J.: The Levenberg–Marquardt algorithm: implementation and theory. In: Watson, G.A. (ed.) Lecture Notes in Mathematics 630: Numerical Analysis, pp. 105–116. Springer, Berlin (1978)
28. Nie, J.: The hierarchy of local minimums in polynomial optimization. Math. Program. **151**(2), 555–583 (2015)
29. Nie, J., Yang, Z., Zhou, G.: The saddle point problem of polynomials. Found. Comput. Math. **22**(4), 1–37 (2021)
30. Nie, J.: Generating polynomials and symmetric tensor decompositions. Found. Comput. Math. **17**, 423–465 (2017)
31. Nie, J.: Low rank symmetric tensor approximations. SIAM J. Matrix Anal. Appl. **38**(4), 1517–1540 (2017)
32. Schorfheide, F.: Loss function-based evaluation of DSGE models. J. Appl. Economet. **15**(6), 645–670 (2000)
33. Sturmfels, B.: Solving systems of polynomial equations. In: CBMS Regional Conference Series in Mathematics, vol. 97. AMS, Providence (2002)
34. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**, 625–653 (1999)
35. Sudre, C.H., Li, W., Vercauteren, T., et al.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 240–248. Springer, Cham (2017)
36. Syed, M.N., Pardalos, P.M., Principe, J.C.: On the optimization properties of the correntropic loss function in data analysis. Optim. Lett. **8**(3), 823–839 (2014)
37. Wang, Q., Ma, Y., Zhao, K., Tian, Y.: A comprehensive survey of loss functions in machine learning. Ann. Data Sci. (2020). https://doi.org/10.1007/s40745-020-00253-5

38. Wu, Z., Shamsuzzaman, M., Pan, E.S.: Optimization design of control charts based on Taguchi's loss function and random process shifts. Int. J. Prod. Res. **42**(2), 379–390 (2004)
39. Yuan, Y.X.: Recent advances in numerical methods for nonlinear equations and nonlinear least squares. Numer. Algebra Control Optim. **1**, 15–34 (2011)