PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Volume 13 · 2020 · Pages 20–29

Accelerating the Lawson-Hanson NNLS solver for large-scale Tchakaloff regression designs

Monica Dessole[a] · Fabio Marcuzzi[a] · Marco Vianello[a]

Communicated by L. Bos

Abstract

We deal with the problem of computing near G-optimal compressed designs for high-degree polynomial regression on fine discretizations of 2d and 3d regions of arbitrary shape. The key tool is Tchakaloff-like compression of discrete probability measures, via an improved version of the Lawson-Hanson NNLS solver for the corresponding full and large-scale underdetermined moment system, that can have for example a size order of 10^3 (basis polynomials) $\times$ 10^4 (nodes).

2010 AMS subject classification: 65C60, 65K05

Keywords: near-optimal regression designs, sparse recovery, Tchakaloff compression, nonnegative least squares, Lawson-Hanson active set method

1 Near-optimal Tchakaloff regression

In this paper we are concerned with the problem of optimizing (possibly high-degree) weighted polynomial regression (optimal design) on a high-cardinality point cloud

$$X = \{x_1, \dots, x_M\} \subset K \subset \mathbb{R}^d \,, \;\; d = 2, 3 \,, \tag{1}$$

which could typically be a fine discretization of a compact region or surface K (the point cloud could in principle be generated by any form of discretization, such as grids, triangulations, low-discrepancy sets, even scattered sets). Regression optimization is here considered from the point of view of finding probability weights on X that sparsify the support and at the same time nearly minimize the estimate of the regression uniform operator norm (or the maximum regressor variance in statistical terms).

Below, we shall denote by $\mathbb{P}^d_n(X)$ the space of d-variate polynomials with total degree not exceeding n, restricted to X, and by $N_n = dim(\mathbb{P}^d_n(X))$ its dimension. As known, if X is $\mathbb{P}^d_n$-determining, i.e. a polynomial in $\mathbb{P}^d_n$ vanishing there vanishes everywhere in $\mathbb{R}^d$, the dimension is $N_n = \binom{n+d}{d} = dim(\mathbb{P}^d_n)$. This holds true for example when X is a polynomial mesh of a $\mathbb{P}^d_n$-determining compact set; cf. the seminal paper [7] for the first general setting of the theory of polynomial meshes.

However, the dimension can be smaller (even for $M > dim(\mathbb{P}^d_n)$ if $d \geq 2$). This certainly happens when X is a subset of an algebraic variety, for example of the 2-sphere $S^2 \subset \mathbb{R}^3$ we have that $N_n \leq (n+1)^2 = dim(\mathbb{P}^d_n(S^2)) < (n+1)(n+2)(n+3)/6 = dim(\mathbb{P}^d_n(\mathbb{R}^3))$.

Now, let the weight array $\mathbf{u} \geq \mathbf{0}$, $\|\mathbf{u}\|_1 = 1$, define a probability measure on X (often called a *design* in statistics), whose support is $\mathbb{P}^d_n(X)$-determining, and let us denote by

$$K^{\mathbf{u}}_n(x, x) = \sum_{j=1}^{N_n} \pi_j^2(x) \in \mathbb{P}^d_{2n}(X) \tag{2}$$

the corresponding Christoffel polynomial, where $\{\pi_j\}$ is any $\mathbf{u}$-orthonormal basis of $\mathbb{P}^d_n(X)$ (the diagonal of the reproducing kernel of the measure, which can be proved to be independent of the basis). Then, the following fundamental estimate holds

$$\|p\|_{\ell^\infty(X)} \leq \sqrt{\max_{x \in X} K^{\mathbf{u}}_n(x, x)}\, \|p\|_{\ell^2_{\mathbf{u}}(X)} \,, \tag{3}$$

that is tight since $(\sqrt{\max_{x\in X} K^{\mathbf{u}}_n(x,x)} = \sup_{p\in\mathbb{P}^d_n(X)} \{\|p\|_{\ell^\infty(X)}/\|p\|_{\ell^2_{\mathbf{u}}(X)}\}$, cf. e.g. [3]), which entails the following estimate for the uniform norm of the weighted regression operator $f \mapsto L^{\mathbf{u}}_n f$ (the $\mathbf{u}$-orthogonal projection of a function f on $\mathbb{P}^d_n(X)$, cf. [4])

$$\|L^{\mathbf{u}}_n\| = \sup_{f \neq 0} \frac{\|L^{\mathbf{u}}_n f\|_{\ell^\infty(X)}}{\|f\|_{\ell^\infty(X)}} \leq \sqrt{\max_{x \in X} K^{\mathbf{u}}_n(x, x)} \,. \tag{4}$$

[a] Department of Mathematics, University of Padova, Italy

In view of *Tchakaloff's theorem* (a cornerstone of quadrature theory, cf. e.g. [20], valid for any measure with finite moments on $\mathbb{R}^d$), for every $k>0$ if $M=card(X)>N_k$ there exists a probability measure with weight array $\mathbf{w}=\{w_1,\dots,w_m\}$, supported at $T=T_k=\{t_1,\dots,t_m\}\subset X$ with cardinality $m=m_k\le N_k$, such that $\sum_{i=1}^M u_i\,p(x_i)=\sum_{j=1}^m w_j\,p(t_j)$ for any $p\in\mathbb{P}_k^d(X)$.

Taking $k=2n$ we get that $K_n^{\mathbf{u}}(x,x)=K_n^{\mathbf{w}}(x,x)$ in X, since both the weight arrays generate the same scalar product in $\mathbb{P}_n^d(X)$, the same 2-norms and hence the same orthogonal polynomials, and thus the weigthed regression operator $L_n^{\mathbf{w}}$ has the same norm estimate of $L_n^{\mathbf{u}}$, namely

$$\|L_n^{\mathbf{w}}\|=\sup_{f\neq 0}\frac{\|L_n^{\mathbf{w}}f\|_{\ell^\infty(X)}}{\|f\|_{\ell^\infty(X)}}\le\sqrt{\max_{x\in X}K_n^{\mathbf{w}}(x,x)}=\sqrt{\max_{x\in X}K_n^{\mathbf{u}}(x,x)}\,. \tag{5}$$

If $M\gg N_{2n}\ge m_{2n}$, this means that the Tchakaloff weights preserve the polynomial moments up to degree $2n$ and the quality of the regressor (measured by its uniform operator norm), with a *sparse support* contained in X. Observe that necessarily $m_{2n}=card(T_{2n})\ge N_n$, since the Tchakaloff points T_{2n} are $\mathbb{P}_n^d(X)$-determining by (3). We shall refer to polynomial regression by Tchakaloff points and weights as to *Tchakaloff (compressed) regression*.

In particular, if the design with weights $\mathbf{u}$ is G-optimal, in the sense that $\max_{x\in X}K_n^{\mathbf{u}}(x,x)=N_n$ (it is easily seen that such a maximum is not smaller than N_n for any design), then the corresponding Tchakaloff regression is by constrution G-optimal itself. The same is true for a near G-optimal regression design, i.e. $\max_{x\in X}K_n^{\mathbf{u}}(x,x)=\theta N_n$ with $\theta\in(0,1)$ close to 1 (the parameter θ is usually called G-efficiency of the regressor in statistics), since by construction Tchakaloff regression preserves G-efficiency.

Indeed, in [4, 5] it has been recently shown that a near G-optimal and sparse regression design can be computed by few iterations of the basic Titterington's multiplicative algorithm [16, 23], which gives cheaply a design with say $\theta=0.95$ but a still large support, followed by extraction of the corresponding Tchakaloff points and weights. It is worth recalling that the computation of optimal regression designs is still an active research subject, cf. e.g. [8] with the references therein. For some deep connections of approximation theoretic with statistical properties of optimal designs we may quote, e.g., [2, 3].

In order to proceed with the discussion on how to implement Tchakaloff's theorem (which in principle is only a nonconstructive existence result) and the computation of near G-optimal Tchakaloff regression on X, we fix a polynomial basis, say $span(p_1,\dots,p_{N_k})=\mathbb{P}_k^d(X)$, and consider the corresponding Vandermonde-like matrix $\mathcal{V}$ for degree k together with the diagonal weight matrix D

$$\mathcal{V}_k=\mathcal{V}_k(X)=(p_j(x_i))\in\mathbb{R}^{M\times N_k}\,,\quad D=diag(\mathbf{u})\in\mathbb{R}^{M\times M}\,. \tag{6}$$

First, we observe that the Tchakaloff theorem can be reformulated as the existence of a nonnegative solution to the *underdetermined moment system*

$$\mathcal{V}_k^t\mathbf{v}=\mathbf{b}\,,\quad \mathbf{b}=\mathcal{V}_k^t\mathbf{u} \tag{7}$$

with at least $M-N_k$ zero components, that in this discrete case is guaranteed by the well-known Caratheodory theorem on conical linear combinations of finite-dimensional vectors, applied to the columns of $\mathcal{V}_k^t$. Observe that since $1\in\mathbb{P}_n^d(X)$, then $\|\mathbf{v}\|_1=\|\mathbf{u}\|_1$, so that if $\mathbf{u}$ is a probability measure, such is also $\mathbf{v}$. Such a solution can be conveniently computed via quadratic programming, solving the NNLS (NonNegative Least Squares) problem

$$\operatorname*{argmin}_{\mathbf{v}\ge\mathbf{0}}\|\mathcal{V}_k^t\mathbf{v}-\mathbf{b}\|_2 \tag{8}$$

by the *Lawson-Hanson iterative active-set method*, which seeks a sparse solution with the appropriate cardinality. The nonzero components of the solution vector are the Tchakaloff weights and determine the Tchakaloff subset $T_k\subset X$.

This approach has been applied in several recent papers on the compression of cubature and least squares, on 2d and 3d domains with different shape, cf., e.g., [17, 18] with the references therein. Indeed, Tchakaloff regression for degree n can be implemented by the Lawson-Hanson method applied to the NNLS problem (8) with $k=2n$. We shall deepen the Lawson-Hanson algorithm in the next section.

2 Sparse recovery through non negative least squares problems

Let $A\in\mathbb{R}^{N\times M}$ and $\mathbf{b}\in\mathbb{R}^N$. The NNLS problem consists in seeking $\mathbf{x}\in\mathbb{R}^M$ that solves

$$\min_{\mathbf{x}\ge\mathbf{0}}\|A\mathbf{x}-\mathbf{b}\|_2^2. \tag{9}$$

This is a convex optimization problem with linear inequality constraints that define the *feasible region*, that is the positive orthant $\left\{\mathbf{x}\in\mathbb{R}^M:\ x_i\ge 0\right\}$. The very first algorithm is due to Lawson and Hanson [14] and it is still one of the most used. Let $\mathbf{x}^\star$ be a solution for this problem. The algorithm is based on the observation that the variable index set $I=\{1,\dots,M\}$ can be partitioned into two sets: the optimal passive set $P^\star=\{j:\ x_j^\star>0\}$ (also called *slack set*), i.e. the support of the optimal solution $\mathbf{x}^\star$, and its complement $Z^\star=\{j:\ x_j^\star=0\}$ the optimal active set. Given a generic set $P\subseteq I$, define $\mathbf{x}_P=(x)_{i\in P}$, and let A_P be the submatrix of A obtained by keeping only columns whose index belongs to P. Then the optimum $\mathbf{x}^\star$ also solves in a least square sense the following Unconstrained Least Squares (ULS) subproblem

$$\mathbf{x}_{P^\star}^\star=\operatorname*{argmin}_{\mathbf{y}}\|A_{P^\star}\mathbf{y}-\mathbf{b}\|_2^2, \tag{10}$$

while the remaining entries are null, that is $\mathbf{x}_{Z^\star}^\star=0$. Starting from a null initial guess $\mathbf{x}=0$ (which is feasible), corresponding to the the choice $P=\emptyset$ and $Z=\{1,\dots,M\}$ for the passive and actives sets respectively, the algorithm incrementally builds an optimal solution by moving indices from the active set Z to the passive set P and vice versa, while keeping the iterates within the feasible region. More precisely, at each iteration first order information is used to detect a column of the matrix A such that the

corresponding entry in the new solution vector will be strictly positive; the index of such a column is moved from the active set Z to the passive set P. Since there's no guarantee that the other entries corresponding to indices in the former passive will stay positive, an inner loop ensures the new solution vector falls into the feasible region, by moving from the passive set P to the active set Z all those indices corresponding to violated constraints. The algorithm terminates in a finite number of steps, since the possible combinations of passive/active set are finite and the sequence of objective function values is strictly decreasing [14]. When the algorithm terminates we have an optimal pair of passive and active sets $P = P^\star$ and $Z = Z^\star$, and a corresponding minimum point $\mathbf{x}^\star$ of (9).

Algorithm 1 Lawson-Hanson $LH(A, \mathbf{b})$

$P = \emptyset$, $Z = \{1, \dots, M\}$, $\mathbf{x} = 0$, $\mathbf{w} = -A^t\mathbf{b}$
while $Z \neq \emptyset$ and $\max(\mathbf{w}) > 0$ **do**
  $\tau = \operatorname{argmax}_i w_i$
  move τ from Z to P
  $\mathbf{z}_P = \operatorname{argmin} \|A_P\mathbf{x} - \mathbf{b}\|_2$, $\mathbf{z}_Z = 0$
  while $\min(\mathbf{z}_P) \leq 0$ **do**
    $Q = P \cap \{i \ : \ z_i \leq 0\}$
    $\alpha = \min_{i \in Q} \frac{x_i}{x_i - z_i}$
    $\mathbf{x} = \mathbf{x} + \alpha(\mathbf{z} - \mathbf{x})$
    move $\{i \ : \ i \in P, \ x_i \leq 0\}$ from P to Z
    $\mathbf{z}_P = \operatorname{argmin} \|A_P\mathbf{x} - \mathbf{b}\|_2$, $\mathbf{z}_Z = 0$
  end while
  $\mathbf{x} = \mathbf{z}$
  $\mathbf{w} = A^t(A\mathbf{x} - \mathbf{b})$
end while
$P^\star = P$, $Z^\star = Z$, $\mathbf{x}^\star = \mathbf{x}$

Since this seminal work, many modifications have been proposed in order to improve the standard Lawson-Hanson algorithm: Bro and de Jong [6] have proposed a variation specifically designed for use in nonnegative tensor decompositions; their algorithm, called "fast NNLS" (FNNLS), reduces the execution time by avoiding redundant computations in Nonnegative Matrix Factorization (NMF) problems arising in tensor decompositions and performs well with multiple right-hand sides, which is not the case here discussed, thus we omit a comparison. Van Benthem and Keenan [24] presented a different NNLS solution algorithm, namely "fast combinatorial NNLS" (FCNNLS), also designed for the specific case of a large number of right-hand sides. The authors exploited a clever reorganization of computations in order to take advantage of the combinatorial nature of the problems treated (multivariate curve resolution) and introduced a nontrivial initialization of the algorithm by means of unconstrained least squares solution. We compare this initialization strategy, briefly denoted in the following sections as LH-init, with the procedure introduced in the present work. Principal block pivoting method introduced by Portugal et al. [19] is an active-set type algorithm which differs from the standard Lawson-Hanson algorithm, since the sequence of iterates produced does not necessarily fall into the feasible region. The convergence is ensured provided the problem is strictly convex, which is not the case for Tchakaloff Least Squares, therefore this algorithm fails in sparse recovering.

In this paper we propose a new, more general, strategy to accelerate the Lawson-Hanson algorithm, as described in the following subsection. We experimentally validate the proposed modification of the Lawson-Hanson algorithm and observe that it achieves better performances even without taking into account the initialization strategy, thus avoiding extra computations.

2.1 An accelerated Lawson-Hanson algorithm

The algorithm here proposed is based, similarly to the principal block pivoting method [19], on the idea of adding multiple indices to the passive set at each outer iteration of Lawson-Hanson algorithm, i.e. to select a block of new columns to insert in matrix A_P, while keeping the current solution vector within the feasible region in such a way that sparse recovery is possible when dealing with non-strictly convex problems, and the number of total iterations and the resulting computational cost decrease. The motivation is that a linear least squares minimization problem needs to be solved at each iteration: this can be done, for example, by computing a QR decomposition, which is substantially expensive.

Suppose, at each outer iteration, to add a set T of new indices to the passive set, chosen among the set $\{i \ : \ w_i > 0\}$, see Lemma (23.17) in [14], where $\mathbf{w} = \{w_i\}$ is the dual solution vector, that is the gradient of the objective function evaluated at the current solution vector $\mathbf{x}$. The set T is initialized to the index chosen by the standard Lawson-Hanson algorithm: $T = \{\tau \ : \ \tau = \operatorname{argmax} \mathbf{w}\}$. It is then extended, within the same iteration, using a set C of candidate indices, defined as the set of indices i different from t for which the dual variable w_i is "large enough", so that the new entries x_i are likely positive. For instance, we choose

$$C = \{i \ : \ w_i > 0.8 \max \mathbf{w}, \ i \neq \tau, i = 1, \dots, M\}. \tag{11}$$

The elements of C to be added are then chosen carefully: note that if the columns corresponding to the chosen indices are linearly dependent, the submatrix A_P will be rank deficient, leading to numerical difficulties in the solution of minimization subproblems like (10). We propose to add k new indices, where k is a parameter that can be tuned, in such a way that, at the end, for every pair of indices in the set T, the corresponding column vectors form an angle whose cosine in absolute value is below a given threshold. Furthermore, it has to be ensured that the cardinality of the new passive set, namely $\operatorname{card}(T) + \operatorname{card}(P)$, does not

exceed the number of rows M, otherwise the corresponding matrix in the least squares problem $[A_P A_T]$ will not be full column rank. The entire procedure, that we call "Lawson-Hanson Deviation Maximization" (LHDM) is summarized in Algorithm 2.

Algorithm 2 Deviation Maximization $DM(A, P, \mathbf{w}, thres, k)$

Sort $\mathbf{w}$ is descending order
$T = \{\text{argmax}\,\mathbf{w}\}$
$C = \{i \ : \ w_i > 0.8 \max \mathbf{w}\} \setminus T$
sort C in descending order according to the corresponding values in $\mathbf{w}$
Set E equal to A_C with normalized columns
for $c \in C$ **do**
  if $\max(|E_c^t E_T|) < thres$ **then** add c to T
  end if
  if $\text{card}(T)$ is equal to k, or $\text{card}(T) + \text{card}(P)$ is equal to M **then** break
  end if
end for

Just like the classic Lawson-Hanson algorithm, this strategy does not ensure that the new iterate will stay feasible, so an intermediate solution $\mathbf{z}$ is computed and, eventually, an analogous inner loop will keep the iterate $\mathbf{x}$ into the feasible region. The new procedure is shown is Algorithm 3.

Algorithm 3 Lawson-Hanson Deviation Maximization $LHDM(A, \mathbf{b}, thres, k)$

$P = \emptyset$, $Z = \{1, \ldots, M\}$, $\mathbf{x} = 0$, $\mathbf{w} = -A^t\mathbf{b}$
while $Z \neq \emptyset$ and $\max(\mathbf{w}) > 0$ **do**
  $T = DM(A, P, \mathbf{w}, thres, k)$
  move T from Z to P
  $\mathbf{z}_P = \text{argmin}\,\|A_P\mathbf{x} - \mathbf{b}\|_2$, $\mathbf{z}_Z = 0$
  while $\min(\mathbf{z}_P) \leq 0$ **do**
    $Q = P \cap \{i \ : \ z_i \leq 0\}$
    $\alpha = \min_{i \in Q} \frac{x_i}{x_i - z_i}$
    $\mathbf{x} = \mathbf{x} + \alpha(\mathbf{z} - \mathbf{x})$
    move $\{i \ : \ i \in P, \ x_i \leq 0\}$ from P to Z
    $\mathbf{z}_P = \text{argmin}\,\|A_P\mathbf{x} - \mathbf{b}\|_2$, $\mathbf{z}_Z = 0$
  end while
  $\mathbf{x} = \mathbf{z}$
  $\mathbf{w} = A^t(A\mathbf{x} - \mathbf{b})$
end while
$P^\star = P$, $Z^\star = Z$, $\mathbf{x}^\star = \mathbf{x}$

This new algorithm, as will be confirmed by the numerical experiments of section 3, produces a substantial reduction in the number of iterations and, consequently, in the execution time.

2.1.1 Algorithmic details

Let us illustrate the algorithmic improvement more in detail. Consider Lemma (23.17) in [14], which substantially ensures that the variable corresponding to the new index chosen by the Lawson-Hanson algorithm to be inserted in the passive set will be positive. Suppose we aim to add a set T of k new indices to the passive set P. The corresponding columns A_T of A are then added to A_P to form the matrix $[A_P \ A_T]$. Suppose moreover that

$$\begin{bmatrix} A_P^t \\ A_T^t \end{bmatrix} \mathbf{b} = \begin{bmatrix} 0 \\ \mathbf{w}_T \end{bmatrix}, \tag{12}$$

where $\mathbf{w}_T > 0$ belongs to $\mathbb{R}^k$, and let Q be the $N \times N$ orthogonal factor of the QR decomposition of matrix A_P, i.e.

$$A_P = Q \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

Then we have

$$Q^t \left[\begin{array}{cc|c} A_P & A_T & \mathbf{b} \end{array}\right] = \left[\begin{array}{cc|c} R & U & \mathbf{u} \\ 0 & V & \mathbf{v} \end{array}\right], \tag{13}$$

where $U \in \mathbb{R}^{|P| \times k}$, $V \in \mathbb{R}^{N-|P| \times k}$, $\mathbf{u} \in \mathbb{R}^{|P|}$ and $\mathbf{v} \in \mathbb{R}^{N-|P|}$. Let $\mathbf{x} = (\mathbf{x}_P^t, \mathbf{x}_T^t)^t \in \mathbb{R}^{|P|+k}$ be the least squares solution of $[A_P \ A_T]\mathbf{x} = \mathbf{b}$. Then equation (13) implies that

$$R\mathbf{x}_P + U\mathbf{x}_T = \mathbf{u},$$
$$V\mathbf{x}_T = \mathbf{v}.$$

In turn, the last equality above yields

$$\mathbf{x}_T = V^{\dagger}\mathbf{v} = (V^tV)^{-1}V^t\mathbf{v}, \tag{14}$$

where $V^{\dagger}$ is the Moore-Penrose inverse of matrix V. Since we have

$$0 = A_p^t\mathbf{b} = \begin{bmatrix} R^t & 0 \end{bmatrix} Q^t\mathbf{b} = \begin{bmatrix} R^t & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = R^t\mathbf{u},$$

where the first equality comes from (12), and R is nonsingular, then $\mathbf{u} = 0$. Therefore, the relation $\mathbf{w}_T = A_T^t\mathbf{b} = (Q^tA_T)^tQ^t\mathbf{b} = U^t\mathbf{u} + V^t\mathbf{v}$ reduces to

$$\mathbf{w}_T = V^t\mathbf{v}$$

and equation (14) rewrites as

$$\mathbf{x}_T = V^{\dagger}\mathbf{v} = (V^tV)^{-1}\mathbf{w}_T. \tag{15}$$

Notice that $W = V^tV$ is a symmetric matrix containing the cosine of the angles between each pair of columns of V, multiplied by the norm of such column vectors. Since $\mathbf{w}_T > 0$, equation (15) ensures $\mathbf{x}_T > 0$ if, for instance, one of the following conditions is satisfied:

1. the matrix V has pairwise orthogonal columns;
2. the matrix $W^{-1} = \left\{\overline{w}_{ij}\right\}_{i,j=1}^{k}$ is positive, i.e. $\overline{w}_{ij} > 0$, $i, j = 1, \ldots, k$, or briefly $W^{-1} > 0$;
3. the matrix $W = \left\{w_{ij}\right\}_{i,j=1}^{k}$ is diagonally dominant and irreducible, and moreover $w_{ii} > 0$, $i = 1, \ldots, k$, and $w_{ij} \leq 0$, $i \neq j$, $i, j = 1, \ldots, k$; then $W^{-1} > 0$ and we fall back into case 2 [1, Theorem 5.21];
4. the matrix $W = c(I - S)$, with c a real positive constant and $\|S\|_{\infty} < 1/2$ (or S irreducible and $\|S\|_{\infty} \leq 1/2$); then W^{-1} is strictly diagonally dominant with a positive diagonal [21]; we need moreover $\mathbf{w}_T$ to be little dispersed around its mean value.

Let $\mathbf{v}, \mathbf{w}$ be arbitrary elements of $\mathbb{R}^N$ and let Q be an orthogonal $N \times N$ matrix. We have

$$\cos(\alpha(\mathbf{v}, \mathbf{w})) = \frac{\mathbf{v}^t\mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|} = \frac{(Q\mathbf{v})^t(Q\mathbf{w})}{\|Q\mathbf{v}\|\|Q\mathbf{w}\|} = \cos(\alpha(Q\mathbf{v}, Q\mathbf{w})),$$

where $\alpha(\mathbf{v}, \mathbf{w})$ is the angle between vectors $\mathbf{v}$ and $\mathbf{w}$, meaning that any orthogonal transformation preserves the cosine of the angle between vectors. Thus, if we impose a threshold on the cosine of the angle between every pair of columns in A_T, then such a threshold holds also for every pair of columns in $[U^t\ V^t]^t$. Therefore, we are able to control the "gap" of diagonally dominance of the matrix

$$[U^t\ V^t] \begin{bmatrix} U \\ V \end{bmatrix} = U^tU + V^tV.$$

If the columns of A_T are pairwise orthogonal, then the same will be true for the columns of $[U^t\ V^t]^t$. Numerical tests highlight that this choice leads to poor performances and a slower convergence rate, since there are very few (or even no) pairwise orthogonal columns indexed in the candidate set C defined in (11). A similar issue is experienced when we restrict C to those indices i such that $w_i = \max \mathbf{w}$. We then discard condition 1.

Condition 2 is sufficient but not necessary and way too strong as shown in the experiments. We then discard condition 2 and hence condition 3, which is stronger and it is consequently discarded.

At the end of this analysis, supported by numerical experiments, we propose an algorithm that aims at satisfying condition 4, since it has practically demonstrated to be the most promising.

Now, the last step is to translate the properties of matrix $[U^t\ V^t]^t$, that we control by a suitable columns selection, to the matrix V, which is responsible for the result (15). Here the experiments give a sound result, at least for the class of problems here considered: as shown in section 3, with this choice of indices it is often the case that $\mathbf{x}_T > 0$ and we observe results like in Figure 3. However, it remains a future work to formally prove the degree to which we can ensure positivity using equation (15).

3 Numerical experiments

In this section, numerical experiments for the empirical validation of the LHDM strategy are carried out. We present below some tests on the NNLS problem (6)-(8), where $\mathbf{u}$ is a near G-optimal discrete design (G-efficiency=95%) with a very large support in 2d and 3d instances, computed following [5] (there the classic Lawson-Hanson algorithm is then adopted to extract a near-optimal Tchakaloff regression design). We are going to compare four strategies:

- the classic Lawson-Hanson algorithm, briefly LH;
- the classic Lawson-Hanson algorithm with initialization, briefly LH-init;
- the Lawson-Hanson Deviation Maximization algorithm with the addition of at most 3 columns, briefly LHDM(3);
- the Lawson-Hanson Deviation Maximization algorithm with the addition of at most $k = \lceil N_{2n}/n \rceil$ columns, with $N_{2n} = dim(\mathbb{P}_{2n}^d)$, briefly LHDM($k$).

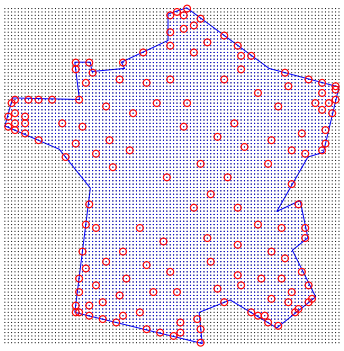
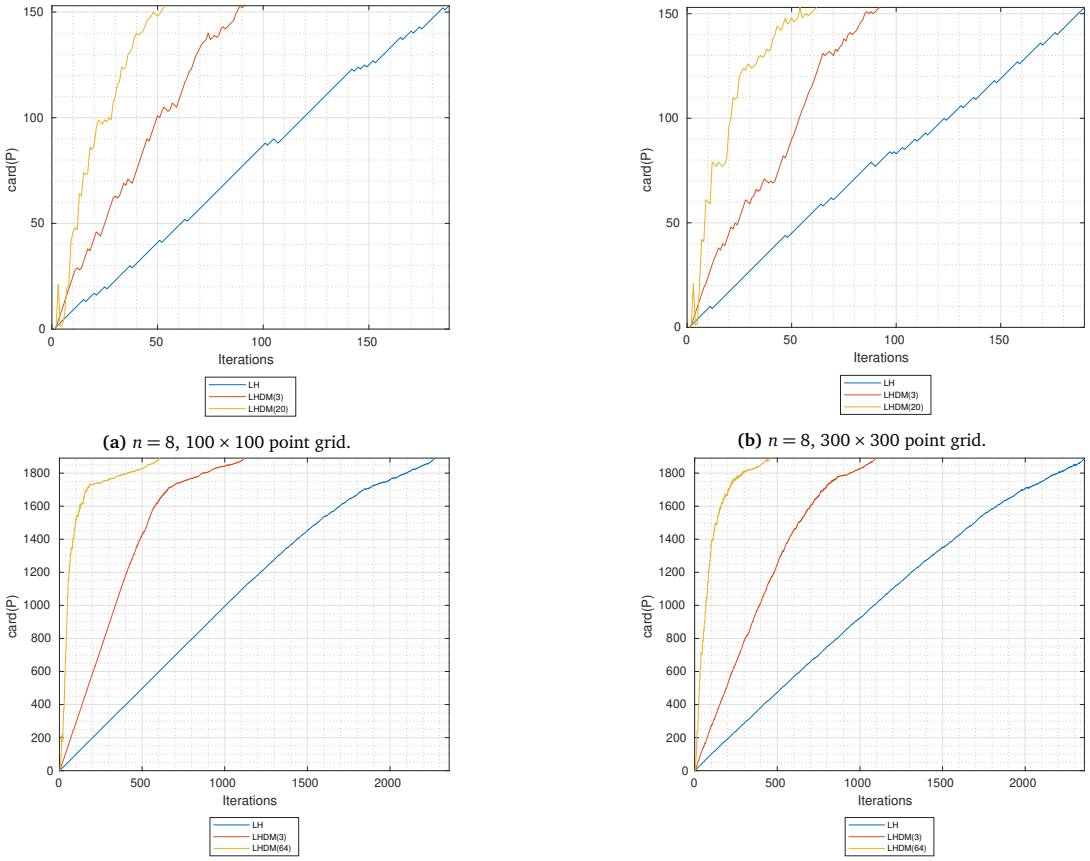


Figure 1: France shaped nonconvex polygon and nearly optimal Tchakaloff regression points at degree $n = 8$.



(a) $n = 8$, 100×100 point grid.

(b) $n = 8$, 300×300 point grid.

(c) $n = 30$, 100×100 point grid.

(d) $n = 30$, 300×300 point grid.

Figure 2: The evolution of the cardinality of the passive set P among the execution of Lawson-Hanson algorithm.

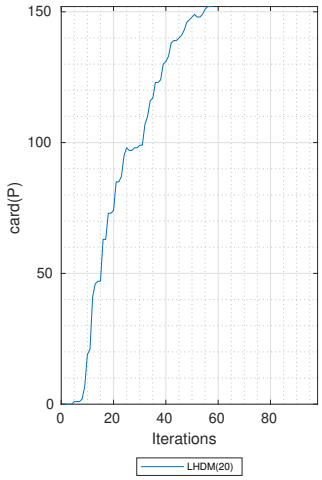
The initialization phase involves the solution of the corresponding Unconstrained Least Squares (ULS) problem. The passive set is then initialized according to the indices corresponding to positive entries in the solution vector. Similarly, the solution vector of ULS is chosen as initial guess of NNLS, provided that negative entries are set to zero in order to obtain a feasible vector. The solution of the ULS can be performed, e.g., by a QR decomposition. Considering that an Householder QR factorization of a matrix $A \in \mathbb{R}^{\alpha \times \beta}$ costs $2\beta^2(\alpha - \beta/3)$ flops [12], the product $Q^t\mathbf{b}$ using Householder vectors costs $O(\alpha\beta)$ and that the solution of a triangular system with matrix R costs $O(\beta^2)$, the initialization step in LH-init costs approximately $2\beta^2(\alpha - \beta/3)$, where here

$\alpha = N_{2n}$ and $\beta = M$. Then, at each iteration it is necessary to update the QR factorization by adding and removing columns [14, cap. 24]. This has a cost $O(\alpha\beta)$ and $O(\beta^2)$, respectively [12, sec. 12.5.2]. Therefore, also these operations are more expensive with the initialization phase, since each iteration has much more columns from the beginning.

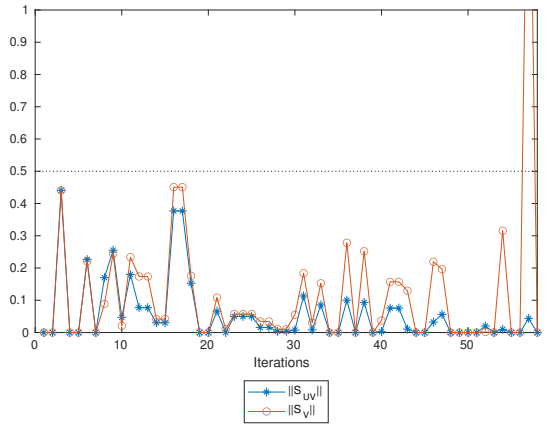
We first explore the algorithm's performances on a two dimensional France shaped highly nonconvex polygon, shown in Figure 1. The point cloud X, that is the initial support of the finite measure, is given by the blue points in Fig. 1, i.e. the points of an evenly spaced grid that fall within the polygon, while the compressed Tchakaloff points are highlighted in red.

Figures 2a, 2b show clearly that our LHDM procedure largely accelerates the standard LH algorithm even for a low regression degree, here $n = 8$: the number of iterations required by LH algorithm is almost four times larger than our LHDM algorithm. Moreover, the results do not depend on the cardinality of the initial grid. In the cases shown in the figure, the Vandermonde-like matrix (6) has dimension $N_{2n} \times M = 153 \times 5746$ on a 100×100 grid, and $N_{2n} \times M = 153 \times 52361$ on a 300×300 grid.

Figures 2c, 2d show results for a higher regression degree, namely $n = 30$, highlighting a remarkable improvement in the number of iterations. Here the Vandermonde-like matrix (6) has dimension $N_{2n} \times M = 1891 \times 5746$ on a 100×100 grid, and $N_{2n} \times M = 1891 \times 52361$ on a 300×300 grid.



(a) The evolution of the passive set.



(b) Norm of the splitting factors.

Figure 3: Numerical validation of condition 4; regression at degree $n = 8$ on a 100×100 point grid.

Figure 3 shows by numerical evidence that our procedure makes condition 4 (see section 2.1.1) hold in most iterations. For a fixed iteration, let U, V be the matrices defined in equation (13). Recall that in order to make (15) hold, it is sufficient to show that $\|S_V\|_\infty < 1/2$, where S_V is the splitting factor of the decomposition $V^tV = c_V(I - S_V)$, and c_V is, for instance, the maximum diagonal value of V^tV. Let us denote by S_{UV} the slitting factor of the following decomposition

$$[U^tV^t]^t \begin{bmatrix} V \\ U \end{bmatrix} = c_{UV}(I - S_{UV}),$$

where c_{UV} is, as before, the maximum diagonal value of the initial matrix. Here, we tested LHDM procedure on the France test case on an initial 100×100 evenly spaced point grid, with regression at degree $n = 8$. Figure 3a on the left shows the evolution of the cardinality of the passive set within the execution of LHDM algorithm. Figure 3b on the right shows at each iteration the infinity norm of the splitting factors S_V and S_{UV}. As we can observe, the deviation maximization technique allows us to control the value of $\|S_{UV}\|_\infty$ (in blue), which is always below the desired $1/2$ bound; the same is true for the values of $\|S_V\|_\infty$ (in red) in most iterations.

Next, we compare LHDM with the initialization strategy by means of Unconstrained Least Squares, first introduced in [24]. By inspecting Figure 4, it is evident that the LHDM procedure is indeed an improvement compared to the mere initialization, both in terms of total number of iterations and in operation count, since our procedure involves, especially in the early iterations, much smaller submatrices. On the right we observe that even the initialization procedure for LHDM can take an advantage over LH-init, in terms of number of iterations, but this is often eroded from the extra computational cost required by the initialization and QR update of bigger matrices on average, so it remains an option.

Finally, we move to a simple three dimensional test case: the domain is the unit cube $[0, 1]^3$, where the point cloud X, that is the initial support of the finite measure, is given by an evenly spaced grid, corresponding to the green points in Figure (5b); we highlighted in red the Tchakaloff points at a low regression degree $n = 5$ in order to keep the figure legible. Here, the advantage of LHDM strategy is actually confirmed, as it requires almost ten times less iterations then the standard LH algorithm, see Figure 5a on the left. In this last test case the regression degree is $n = 10$ and corresponding the Vandermonde-like matrix (6) has dimension $N_{2n} \times M = 1771 \times 64000$ on a $40 \times 40 \times 40$ evenly spaced point grid.

We show in Table 1 an example of the execution times obtained with Matlab on a workstation with an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz. The times here reported agree with the gain in the total number of iterations. However, these are to be

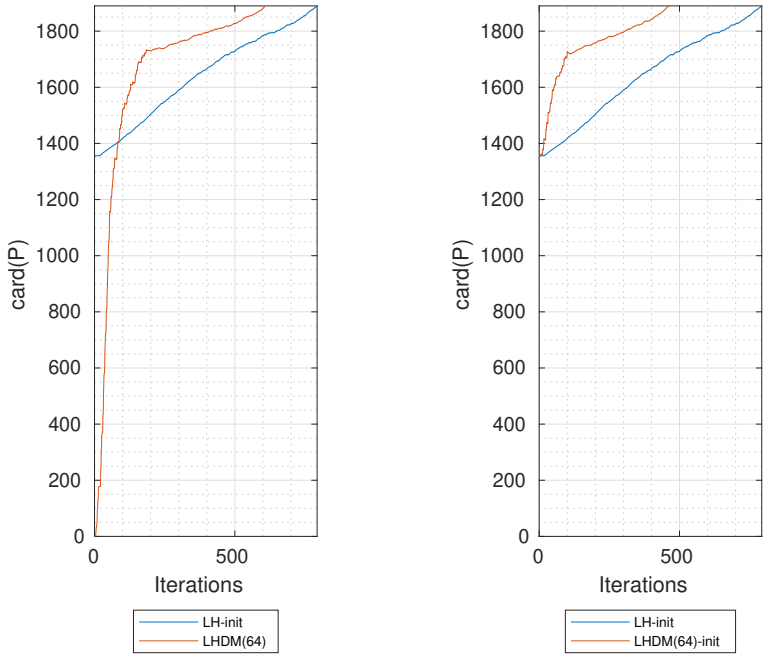
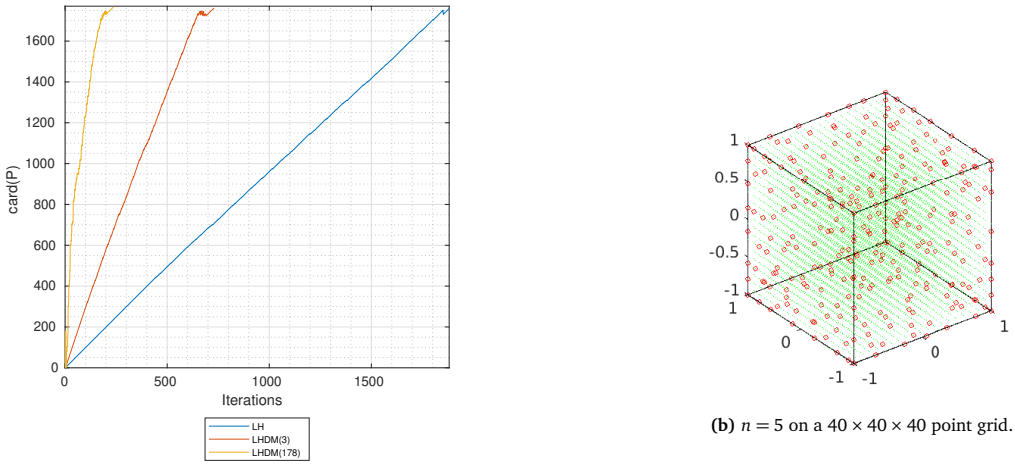


Figure 4: Comparison of LHDM versus initialization strategy on the France test case; regression at degree $n = 30$ on a 100×100 point grid.

considered as indicative only, due to the interpreted execution of Matlab scripts. An high-performance implementation of this algorithm is premature at this moment, but it is expected as a future work.



(a) $n = 10$ on a $40 \times 40 \times 40$ point grid.

(b) $n = 5$ on a $40 \times 40 \times 40$ point grid.

Figure 5: On the left, the evolution of the cardinality of the passive set P among the execution of Lawson-Hanson algorithm. On the right, nearly optimal Tchakaloff regression points.

Test	LH	LH-init	LHDM(3)	LHDM(k)
France $n = 8$, $m = 100$	0.225	0.141	0.125	0.086
France $n = 8$, $m = 300$	0.927	0.785	0.591	0.394
France $n = 30$, $m = 100$	524.900	301.792	327.897	230.984
France $n = 30$, $m = 300$	605.121	533.820	323.205	173.401
Cube $n = 10$, $m = 40$	389.037	320.422	163.333	64.012

Table 1: Execution times (in seconds) of the different methods for each test carried out.

4 Conclusions and future perspectives

In this paper we have dealt with the efficient computation of Tchakaloff points for optimizing (possibly high-degree) weighted polynomial regression, often referred as optimal design in statistics. Regression optimization is here considered from the point of view of finding probability weights that sparsify the support and at the same time nearly minimize the estimate of the regression uniform operator norm. This problem boils down to finding a sparse solution of a nonnegative least squares problem, where the matrix is strongly underdetermined.

Having in mind such an application, we derived an accelerated Lawson-Hanson algorithm, which we named "Lawson-Hanson Deviation Maximization" (LHDM for short), here presented. We have tested this algorithm on different instances of the Tchakaloff problem, obtaining a good speed-up for a serial computer, with respect to the state-of-the-art. In particular, the contribution of this paper is mainly due to a new strategy to choose simultaneously, at each iteration, the biggest bunch of columns that can be added to build incrementally a non-negative solution with the smallest support. The efficiency of this strategy has been not entirely demonstrated, and we complemented this with an experimental evidence. Indeed, it is an open problem to determine if a proof is possible using specific properties of the matrices arising in the problem here considered, or it can be demonstrated for general matrices.

Moreover, as has been pointed out in the present work, the linear algebra problem that arises in the computation of Tchakaloff points is affected by the so called *curse of dimensionality*. For higher-dimensional problems, the underlying linear algebra becomes even more large-scale and parallel computing is a possible answer for efficiency, as the authors experienced recently with GPU architectures in other kind of matrix problems [9]. Therefore, an extension of this work, and of [15], would be to formulate this algorithm e.g. in a BLAS3-based parallel implementation on GPUs, thus exploiting the massive parallelism attainable in BLAS3 operations on big matrices. The challenge is the intrinsic sequential structure of Lawson-Hanson algorithm, that creates a serial bottleneck. Another possible solution may consist in a clever exploitation of the Vandermonde-like structure of the matrices here involved. In this way, this algorithm could work without explicitly forming these matrices, when they are too large. However, this step is far from being immediate, since it requires the study and the design of dedicated linear algebra algorithms, e.g. QR decomposition and related update/downdate procedures, which are the main ingredients of any Lawson-Hanson based process.

Acknowledgements

Work partially supported by the DOR funds and the Project BIRD192932 of the University of Padova, and by the GNCS-INdAM 2019 project "Tecniche innovative e parallele per sistemi lineari e nonlineari di grandi dimensioni, funzioni ed equazioni matriciali ed applicazioni". The authors gratefully acknowledge the doctoral grant funded by BeanTech s.r.l. "GPU computing for modeling, nonlinear optimization and machine learning". This research has been accomplished within the RITA "Research ITalian network on Approximation".

References

[1] Bini, D., Capovani, M., Menchi, O. Metodi numerici per l'algebra lineare. *Collana di matematica. Testi e manuali. Zanichelli*, 1988.

[2] Bloom, T., Bos, L., Levenberg, N. The Asymptotics of Optimal Designs for Polynomial Regression. arXiv preprint: 1112.3735.

[3] Bloom, T., Bos, L., Levenberg, N., Waldron, S. On the Convergence of Optimal Measures. *Constr. Approx.* 32: 159–179, 2010.

[4] Bos, L., Piazzon, F., Vianello, M. Near G-optimal Tchakaloff designs. *Comput. Statistics*, published online 25 October 2019.

[5] Bos, L., Vianello, M. CaTchDes: Matlab codes for Caratheodory-Tchakaloff Near-Optimal Regression Designs. *SoftwareX*, 10: 100349, 2019.

[6] Bro, R., de Jong, S. A fast nonnegativity constrained least squares algorithm. *Journal of Chemometrics*, 11.5: 393-401, 1997.

[7] Calvi, J.-P., Levenberg, N. Uniform approximation by discrete least squares polynomials. *J. Approx. Theory*, 152: 82–100, 2008.

[8] De Castro, Y., Gamboa, F., Henrion, D., Hess, R., Lasserre, J.-B. Approximate Optimal Designs for Multivariate Polynomial Regression. *Ann. Statist.*, 47: 127–155, 2019.

[9] Dessole, M., Marcuzzi, F. Fully iterative ILU preconditioning of the unsteady Navier-Stokes equations for GPGPU. *Comput. Math. Appl.*, 77: 907–927, 2019.

[10] Foucart, S., Koslicki, D. Sparse recovery by means of nonnegative least squares. *IEEE Signal Proc. Lett.*, 21: 498–502, 2014.

[11] Foucart, S., Rahut, H. A Mathematical Introduction to Compressive Sensing. *Birkhäuser*, 2013.

[12] Golub, Gene H., Van Loan, Charles F. Matrix Computations (3rd Ed.). *Johns Hopkins University Press*, 1996.

[13] Kiefer, J., Wolfowitz, J. The equivalence of two extremum problems. *Canad. J. Math.* 12: 363–366, 1960.

[14] Lawson, C.L., Hanson, R.J. Solving Least Squares Problems. *Classics in Applied Mathematics 15, SIAM, Philadelphia*, 1995.

[15] Luo, Y., Duraiswami, R. Efficient parallel non-negative least-squares on multi-core architectures. *SIAM Journal on Scientific Computing*, 33: 2848–2863, 2011.

[16] Mandal, A., Wong, W.K., Yu, Y. Algorithmic Searches for Optimal Designs. *Handbook of Design and Analysis of Experiments (A. Dean, M. Morris, J. Stufken, D. Bingham Eds.)*. *Chapman & Hall/CRC, New York*, 2015.

[17] Piazzon, F., Sommariva, A., Vianello, M. Caratheodory-Tchakaloff Subsampling. *Dolomites Res. Notes Approx. DRNA*, 10: 5–15, 2017.

[18] Piazzon, F., Sommariva, A., Vianello, M. Caratheodory-Tchakaloff Least Squares. *International Conference on Sampling Theory and Applications (SampTA)*, 672–676. IEEE Xplore Digital Library, 2017.

[19] Portugal, Luìs F., Júdice, Joaquim J., Vicente, Luìs N. A Comparison of Block Pivoting and Interior-Point Algorithms for Linear Least Squares Problems with Nonnegative Variables. *Mathematics of Computation*, 63: 625–643 1994.

[20] Putinar, M. A note on Tchakaloff's theorem. *Proc. Amer. Math. Soc.*, 125: 2409–2414, 1997.

[21] Radons, M. Direct solution of piecewise linear systems. *Theor. Comput. Sci.*, 626: 97-109, 2016.

[22] Slawski, M. Nonnegative least squares: comparison of algorithms. Paper and code available online at: https://sites.google.com/site/slawskimartin/code

[23] Titterington, D.M. Algorithms for computing d-optimal designs on a finite design space. *Proc. 1976 Conference on Information Sciences and Systems*, Baltimora, 1976.

[24] Van Benthem, M.H., Keenan, R. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *J. Chemometrics*, 18: 441–450, 2004.