Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2021) 45: 2103 – 2107
© TÜBİTAK
doi:10.3906/mat-2012-65

Sparse polynomial interpolation with Bernstein polynomials

Erdal İMAMOĞLU*
Department of Mathematics, Faculty of Arts and Sciences, Kırklareli University, Kırklareli, Turkey

Received: 17.12.2021 • **Accepted/Published Online:** 25.07.2021 • **Final Version:** 16.09.2021

Abstract: We present an algorithm for interpolating an unknown univariate polynomial f that has a t sparse representation ($t << \deg(f)$) using Bernstein polynomials as term basis from $2t$ evaluations. Our method is based on manipulating given black box polynomial for f so that we can make use of Prony's algorithm.

Key words: Symbolic computation, sparse polynomial interpolation, Bernstein polynomials, Bernstein polynomial basis

1. Introduction

Univariate polynomial interpolation is the process of reconstructing an unknown polynomial f from some set of its evaluations. Let $\{P_i(x)\}_{i=1,2,3,\ldots}$ be a vector space basis for the univariate polynomials $K[x]$ where K is a field. Any polynomial $f \in K[x]$ can be represented as a linear combination of t basis elements,

$$f(x) = \sum_{j=1}^{t} c_j P_{\delta_j}(x) \text{ where } c_j \neq 0, 0 \leq \delta_1 < \delta_2 < \cdots < \delta_t.$$

Here $t << \deg(f)$ is the sparsity of f with respect to the basis $\{P_i(x)\}_{i=1,2,3,\ldots}$. Sparse interpolation algorithms reconstruct the term coefficients c_j and the term degrees δ_j from values $a_i = f(\omega_i)$, at $x = \omega_i \in K$. Current algorithms use $i = 1, \ldots, 2t$ evaluations with a known sparsity t (if t is unknown, current algorithms use $i = 1, \ldots, t + B$ evaluations where B is an upper bound for t, i.e. $B \geq t$).

A sparse interpolation algorithm is given by Prony [1]. Ben-Or and Tiwari [2] adapted Prony's algorithm to computer algebra and they gave an interpolation algorithm using standard power basis $\{P_i(x) = x^i\}_{i=1,2,\ldots}$. That algorithm interpolates a polynomial with coefficients in $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$, and can be adapted to finite fields. More details about Prony's algorithm can be found at [2–5] and references therein. Within the last years, different algorithms for interpolating a sparse polynomial using term basis different than power basis are designed: [6] uses Pochhammer polynomials and Chebyshev polynomials of the first kind; [7] uses Legendre polynomials; [8] and [9] use Chebyshev polynomials of the second first kind and second kind, respectively, as term basis. Algorithms in [10] perform sparse interpolation using all four kinds of Chebyshev polynomials as term basis.

Bernstein polynomials of degree n form a basis, which is also called Bernstein-Bézier basis, for the vector space of polynomials of degree $\leq n$. Bernstein–Bézier basis is the standard way in computer-aided geometric

*Correspondence: eimamoglu@klu.edu.tr, ei11b@my.fsu.edu

2010 *AMS Mathematics Subject Classification:* 68W30.

design for representing a polynomial curve. In the present paper, we examine the problem of interpolating a degree n sparse polynomial f using Bernstein polynomials as term basis, i.e. $\{P_i(x) = B_{i,n}(x)\}_{i=1,2,\ldots,n}$. We want to compute c_j and δ_j such that

$$f(x) = \sum_{j=1}^{t} c_j B_{\delta_j,n}(x) \text{ where } c_j \neq 0, 0 \leq \delta_1 < \delta_2 < \cdots < \delta_t \leq n.$$

from given a black box for f, sparsity t and $n = \deg(f)$ by using $2t$ evaluations of the black box.

We start with defining a black box for an unknown polynomial and Bernstein polynomials. We state the problem at the end of this section. We give our result and state our algorithm in the next section.

Definition 1.1 *A black box for an unknown polynomial $f \in K[x]$ is an object which takes ω for x and produces $a = f(\omega)$:*

$$\omega \to \blacksquare \to a = f(\omega).$$

See [11] for more details.

We assume a black box for f always returns correct evaluation without any error. If a black box for f is given, we can compute evaluations of f using the given its black box.

Definition 1.2 *We define i-th Bernstein polynomial of degree d as*

$$B_{i,d}(x) = \binom{d}{i} x^i (1-x)^{d-i}$$

where $\binom{d}{i}$ is a binomial coefficient. More properties and details of Bernstein polynomials can be found in [12, 13].

If Π_d is the vector space of polynomials of degree $\leq d$ with real coefficients, then the set of all Bernstein polynomials of degree d, $\{B_{i,d}(x)\}_{i=0,1,\ldots,d}$, form a basis (Bernstein-Bézier basis) for the vector space Π_d. That means a polynomial f of degree $n \leq d$ can be represented by a linear combination of t Bernstein polynomials of degree n, i.e. $f(x) = \sum_{j=1}^{t} c_j B_{\delta_j,n}(x)$ where $c_j \neq 0$, $0 \leq \delta_1 < \delta_2 < \cdots < \delta_t \leq n = \deg(f)$. This representation is useful when $t << \deg(f)$.

Example 1.3 *The degree* 37 *polynomial*

$$\begin{aligned}
f(x) = &-24937107930x^{37} + 598488958620x^{36} - 6882595285230x^{35}\\
&+ 50472060297120x^{34} - 264975875536680x^{33} + 1059888367802880x^{32}\\
&- 3356237492989920x^{31} + 8630011484851680x^{30} - 18337677165381420x^{29}\\
&+ 32597023240892880x^{28} - 48886976389897800x^{27} + 62200337061139200x^{26}\\
&- 67344796339984800x^{25} + 62095600522519200x^{24} - 48681243903322800x^{23}\\
&+ 32302743492096000x^{22} - 17981019326655000x^{21} + 8250114749877000x^{20}\\
&- 2996836554442500x^{19} + 757095550596000x^{18} - 37854777529800x^{17}\\
&- 100946073412800x^{16} + 82592241883200x^{15} - 46084076992800x^{14}\\
&+ 21371241332700x^{13} - 8558471441520x^{12} + 2962547806680x^{11}\\
&- 877791942720x^{10} + 219447985680x^{9} - 45403031520x^{8} + 7567171920x^{7}\\
&- 976409280x^{6} + 91538370x^{5} - 5547780x^{4} + 163170x^{3}
\end{aligned}$$

can be written as a sum of only $t = 2$ *Bernstein polynomials of degree* 37 :

$$f(x) = 21B_{3,37}(x) - 7B_{13,37}(x).$$

Here the polynomial f *has sparsity* $t = 2$ *in terms of Bernstein polynomials. Note that* $2 = t << \deg(f) = 37$. *We can interpolate* f *in Bernstein polynomials from* $2t$ *evaluations of its black box.*

Problem 1.4 *From given a black box for a polynomial* $f \in \Pi_d$ *(*f *is unknown),* $n = \deg(f)$, *and sparsity* t, *using* $2t$ *evaluations of the black box, compute the* c_j *and the* δ_j *such that*

$$f(x) = \sum_{j=1}^{t} c_j B_{\delta_j,n}(x) \text{ where } c_j \neq 0, 0 \leq \delta_1 < \delta_2 < \cdots < \delta_t \leq n.$$

2. Results and algorithm

Bernstein polynomial $B_{i,d}(x)$ satisfies the reduction formula

$$(1+x)^d B_{i,d}\Big(\frac{x}{1+x}\Big) = \binom{d}{i} x^i.$$

We reduce the sparse polynomial interpolation problem in Bernstein polynomials to interpolation problem in the standard basis, so we can make use of Prony's algorithm [1–5]. We perform change of variables $x \to \frac{z}{1+z}$

on f and use the multiplier $(1+z)^n$ to define g :

$$
\begin{aligned}
g(z) := (1+z)^n f\Big(\frac{z}{1+z}\Big) &= (1+z)^n \sum_{j=1}^{t} c_j B_{\delta_j,n}\Big(\frac{z}{1+z}\Big) \\
&= \sum_{j=1}^{t} c_j (1+z)^n B_{\delta_j,n}\Big(\frac{z}{1+z}\Big) \\
&= \sum_{j=1}^{t} c_j \binom{n}{\delta_j} z^{\delta_j} \\
&= \sum_{j=1}^{t} C_j z^{\delta_j} \text{ where } C_j = c_j \binom{n}{\delta_j}.
\end{aligned}
$$

The resulting polynomial $g(z) = \sum_{j=1}^{t} C_j z^{\delta_j}$ has the sparsity t in the standard power basis $\{z^i\}_{i=0,1,2,\ldots}$ (if f is a sum of t Bernstein polynomials of degree n, then g has t terms). The polynomial g can be interpolated by using Prony's algorithm [1–5] which uses $2t$ evaluations of the black box for g. Since g and f are related, we can use the black box for f to get evaluations of g.

Example 2.1 *Consider the degree $n = 37$ polynomial f in Example 1.3:*

$$
f(x) = 21B_{3,37}(x) - 7B_{13,37}(x) = \sum_{j=1}^{2} c_j B_{\delta_j,37}(x).
$$

The polynomial f corresponds to

$$
g(z) := (1+z)^{37} f\Big(\frac{z}{1+z}\Big) = 163170z^3 - 24937271100z^{13} = \sum_{j=1}^{2} C_j z^{\delta_j}.
$$

Here $\delta_1 = 3, \delta_2 = 13$ and $c_1 = 21, c_2 = -7, C_1 = 161370, C_2 = -24937271100$. Here we can see that f is a sum of $t = 2$ Bernstein polynomials and g has only $t = 2$ terms. Prony's algorithm [1–5] interpolates g in standard basis from $2t$ evaluations. Hence, we can interpolate f in Bernstein polynomials from $2t$ evaluations.

We state our algorithm as follows:

2.1. Algorithm: interpolation with Bernstein polynomials as term basis

- Input: A black box for f, degree n of f and the sparsity t.
- Output: The δ_j and the c_j such that $f(x) = \sum_{j=1}^{t} c_j B_{\delta_j,n}(x)$.

1. Define $g(z) := (1+z)^n f(\frac{z}{1+z}) = \sum_{j=1}^{t} C_j z^{\delta_j}$ where $C_j = c_j \binom{n}{\delta_j}$.
2. Use Prony's algorithm [1–5] to compute the δ_j and the C_j such that $g(z) = \sum_{j=1}^{t} C_j z^{\delta_j}$ (Prony's algorithm reconstructs c_j, δ_j from $2t$ evaluations of the black box).

3. Compute c_j from the equality $C_j = c_j\binom{n}{\delta_j}$.

4. Return the δ_j and the c_j.

Remark 2.2 *If t is unknown and a bound $B \geq t$ is given on input, one can compute t from $t+B$ evaluations of the black box. One can also compute t from $2t+2$ evaluations by using early termination algorithm introduced in* [14].

Acknowledgment

The authors are supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project 119F426.

References

[1] Prony R. Essai expérimental et analytique sur les lois de la Dilatabilité de fluides élastiques et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. J. de l'École Polytechnique, 1795, (1):24–76 (in French).

[2] Ben-Or M, Tiwari P. A deterministic algorithm for sparse multivariate polynomial interpolation. ACM STOC '88, 1988, 301–309.

[3] Brezinski B. History of continued fractions and Padé approximations. Heidelberg, Germany: Springer Verlag, 1991.

[4] Roche DS. Efficient computation with sparse and dense polynomials. PhD, University of Waterloo, Waterloo, ON, Canada, 2011.

[5] Arnold A. Sparse polynomial interpolation and testing. PhD, University of Waterloo, Waterloo, ON, Canada, 2016.

[6] Lakshman YN, Saunders BD. Sparse polynomial interpolation in non-standard bases. SIAM J. Comput., 1995, 24(2):387-397.

[7] Peter T, Plonka G, Rosca D. Representation of sparse Legendre expansions. Journal of Symbolic Computation, 2013, (50):159-169.

[8] Arnold A, Kaltofen EL. Error-correcting sparse interpolation in the Chebyshev basis. In: Proceedings of the 2015 ACM International Symposium on Symbolic and Algebraic Computation, ISSAC'15, 2015, 21-28.

[9] Potts D, Taschee M. Sparse polynomial interpolation in Chebyshev bases. Linear Algebra and its Applications, 2014, (441):61-87.

[10] Imamoglu E, Kaltofen EL, Yang Z. Sparse polynomial interpolation with arbitrary orthogonal polynomial bases. In: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, ISSAC'18, 2018, 223-230.

[11] Kaltofen, EL, Trager B. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. In: Proceedings of the 29th Annual Symposium on Foundations of Computer Science, pages 296-305. IEEE, 1988.

[12] Farouki, RT. The Bernstein polynomial basis: A centennial retrospective. Computer Aided Geometric Design, 2012, 29(6):379-419.

[13] Farin G. Curves and surfaces for computer aided geometric design. Boston, MA, USA: Academic Press, 1993.

[14] Kaltofen EL, Lee W. Early termination in sparse interpolation algorithms. Journal of Symbolic Computation, 2003, 36(3):365-400.