Quantum Information and Computation, Vol. 8, No. 10 (2008) 0943–0950

A NOTE ON QUANTUM ALGORITHMS AND THE MINIMAL DEGREE OF ε-ERROR POLYNOMIALS FOR SYMMETRIC FUNCTIONS

RONALD de WOLF
CWI. Kruislaan 413
1098SJ Amsterdam, The Netherlands
rdewolf@cwi.nl

Received April 11, 2008
Revised May 31, 2008

The degrees of polynomials representing or approximating Boolean functions are a prominent tool in various branches of complexity theory. Sherstov [31] recently characterized the minimal degree $deg_\varepsilon(f)$ among all polynomials (over $\mathbb{R}$) that approximate a *symmetric* function $f : \{0,1\}^n \to \{0,1\}$ up to worst-case error ε: $deg_\varepsilon(f) = \widetilde{\Theta}\left(deg_{1/3}(f) + \sqrt{n \log(1/\varepsilon)}\right)$. In this note we show how a tighter version (without the log-factors hidden in the $\widetilde{\Theta}$-notation), can be derived quite easily using the close connection between polynomials and quantum algorithms.

Keywords: quantum algorithms, approximating polynomials, quantum proof, symmetric functions, probability theory
Communicated by: R Cleve & M Mosca

1 Introduction

1.1 Setting

Boolean functions are one of the primary objects of study in theoretical computer science. Such functions can be represented or approximated by *polynomials* in a number of ways, and the algebraic properties of such polynomials (such as their degree) often give information about the complexity of the function involved. Areas where this approach has been used include circuit complexity [28, 33, 6], complexity classes [8, 7, 34], decision trees [26, 12], communication complexity [11, 29, 32, 21], and learning theory [25, 23].

In this note we focus on polynomials over the field of real numbers. An *n-variate multilinear polynomial* p is a function $p : \mathbb{R}^n \to \mathbb{R}$ that can be written as

$$p(x_1, \ldots, x_n) = \sum_{S \subseteq [n]} a_S \prod_{i \in S} x_i,$$

for some real numbers a_S. The *degree* of p is $deg(p) = \min\{|S| \mid a_S \neq 0\}$. If is well known (and easy to show) that every function $f : \{0,1\}^n \to \mathbb{R}$ has a unique representation as such a polynomial; $deg(f)$ is defined as the degree of that polynomial.

In many applications it suffices if the polynomial is *close* to f instead of being equal to it:

Definition 1 *The* ε-approximate degree *of* $f : \{0,1\}^n \to \mathbb{R}$ *is*

$$deg_\varepsilon(f) = \min\{deg(p) \mid \forall x \in \{0,1\}^n : |p(x) - f(x)| \leq \varepsilon\}.$$

A function f is called *symmetric* if its value only depends on the Hamming weight $|x|$ of its input $x \in \{0,1\}^n$. Equivalently, $f(x) = f(\pi(x))$ for all $x \in \{0,1\}^n$ and all permutations $\pi \in S_n$. We will restrict attention here to symmetric functions f. Examples are OR, AND, PARITY, MAJORITY etc. Since the only thing that matters is the Hamming weight $|x|$ of the input, one can actually restrict attention to *univariate* polynomials. We say that a univariate polynomial p *ε-approximates* a symmetric function f if $|p(|x|) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$. By a technique called *symmetrization* [24], it turns out that for symmetric functions, the minimal degree of such univariate ε-approximating polynomials is the same degree $deg_\varepsilon(f)$ as for n-variate multilinear polynomials. Hence we can switch back and forth between these two kinds of polynomials at will.

Paturi [27] tightly characterized the 1/3-approximate degree $deg_{1/3}(f)$ of all symmetric f (see the start of Section 2 for the precise statement). Recently Sherstov [31] studied the dependence on the error ε. He proved the surprisingly clean result that for all $\varepsilon \in [2^{-n}, 1/3]$,

$$deg_\varepsilon(f) = \widetilde{\Theta}\left(deg_{1/3}(f) + \sqrt{n \log(1/\varepsilon)}\right),$$

where the $\widetilde{\Theta}$ notation hides some logarithmic factors. Note that the statement is false if $\varepsilon \ll 2^{-n}$, since clearly $deg(f) \leq n$ for all f.

Sherstov gave an interesting application of his result in the context of the inclusion-exclusion principle of probability theory. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Suppose one has events $A_1, \ldots, A_n$ in some probability space, and one knows the exact values of $\Pr[\cap_{i \in S} A_i]$ for all sets $S \subseteq [n]$ of size at most k. How well can we now estimate $\Pr[f(A_1, \ldots, A_n)]$? In other words, what is the maximal difference $|\Pr[f(A_1, \ldots, A_n)] - \Pr[f(B_1, \ldots, B_n)]|$, maximized over all pairs of sequences of events $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$ satisfying $\Pr[\cap_{i \in S} A_i] = \Pr[\cap_{i \in S} B_i]$ for all sets S of size up to k? Sherstov [31, Theorem 1.1] proved that this maximal distance equals exactly twice the minimal error ε for which $\deg_\varepsilon(f) \leq k$. Hence tight bounds on $\deg_\varepsilon(f)$ give tight bounds for this question of probability theory. This generalizes earlier results for the case where f is the OR function, i.e. where one is estimating $\Pr[\cup_{i \in [n]} A_i]$ [22, 17].

1.2 Our result

In this note we give a different proof, for a slightly tighter version of Sherstov's degree-result:

Theorem *For every non-constant symmetric function $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon \in [2^{-n}, 1/3]$:*

$$deg_\varepsilon(f) = \Theta\left(deg_{1/3}(f) + \sqrt{n \log(1/\varepsilon)}\right).$$

Note that there are no hidden logarithmic factors anymore. As a consequence, the result on approximate inclusion-exclusion is sharpened as well, but we won't elaborate on that here.

The lower bound on $deg_\varepsilon(f)$ follows immediately from combining Paturi's tight bound for $deg_{1/3}(f)$ with the tight bound on the ε-approximate degree of the OR-function proved in [10]. More interestingly, our upper bound is obtained by exhibiting an efficient ε-error *quantum algorithm* for computing a symmetric function. It is well known (at least in quantum circles) that the acceptance probability of a quantum algorithm that makes T queries to its input can be written as an n-variate multilinear polynomial of degree at most $2T$ [5] (see also [14]). The upper bound of Theorem **??** actually applies to a larger class of functions, namely all

functions $f : \{0,1\}^n \to [0,1]$ that are constant when $|x| \in \{t, \ldots, n-t\}$. These functions may take arbitrary real values in $[0,1]$ and may be non-symmetric for inputs with smaller or larger Hamming weights. For every such function we have $deg_\varepsilon(f) = O(\sqrt{tn} + \sqrt{n \log(1/\varepsilon)})$.

1.3 Discussion

The main message of this note is that one can obtain essentially optimal polynomial approximations of symmetric Boolean functions by arguing about quantum algorithms. This fits in a line of papers in recent years that prove or reprove theorems about various topics in classical computer science or mathematics with the help of quantum computational techniques. This includes results about locally decodable codes [19, 35], classical proof systems for lattice problems inspired by earlier quantum proof systems [3, 4], limitations on classical algorithms for local search [1] inspired by an earlier quantum proof, a proof that the complexity class PP is closed under intersection [2], lower bounds on the rigidity of Hadamard matrices [36], classical formula size lower bounds from quantum query lower bounds [20], and an approach to proving lower bounds for classical circuit depth using quantum communication complexity [18].

There are advantages as well as disadvantages to our approach in this note. We feel that for someone familiar with quantum algorithms and their connection to polynomials, our proof should be quite simple and straightforward. Also, our bound is tight up to constant instead of logarithmic factors, and applies to a larger class of functions than Sherstov's. On the other hand, for those unfamiliar with quantum computation our proof is probably somewhat less accessible. Also, we do not construct the ε-approximating polynomials explicitly (though one may derive them from our quantum algorithm), in contrast to Sherstov's construction based on Chebyshev polynomials.

2 Proof

Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant symmetric function that is constant if the Hamming weight $|x|$ of the input is in the interval $\{t, \ldots, n-t\}$ (where $0 < t \leq n/2$ is the smallest t for which this holds). We know $deg_{1/3}(f) = \Theta(\sqrt{tn})$ from Paturi [27]. In the next two subsections we provide matching upper and lower bounds on $deg_\varepsilon(f)$, thus proving Theorem **??**.

2.1 Upper bound on $deg_\varepsilon(f)$

Beals et al. [5] showed that the acceptance probability of a T-query quantum algorithm on n-bit input is a multilinear n-variate polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree at most $2T$. Hence it suffices to give an ε-error quantum algorithm for f that uses $O(deg_{1/3}(f) + \sqrt{n \log(1/\varepsilon)})$ queries. The acceptance probability of the algorithm will be our ε-error polynomial.

Here is the algorithm. It uses various quantum algorithms based on Grover's search algorithm, which are explained in Appendix 1. Let $x \in \{0,1\}^n$ be the input string. The algorithms have access to this string via *queries*. In the quantum case, one query is one application of the unitary that maps $|i\rangle \mapsto (-1)^{x_i}|i\rangle$. A *solution* is an index $i \in [n]$ such that $x_i = 1$.

1. Use t repeated applications of exact Grover to try to find up to t solutions (initially assuming $|x| = t$, and "crossing out" in subsequent applications the solutions already found). If $|x| \leq t$, then *with probability 1* these repeated applications find all solutions. This costs $O(\sqrt{tn})$ queries.

2. Use $\varepsilon/2$-error Grover to try to find one more solution. This costs $O(\sqrt{n\log(1/\varepsilon)})$ queries.

3. The same as step 1, but now looking for positions of 0s instead of 1s.

4. The same as step 2, but now looking for a 0 instead of a 1.

The total number of queries is indeed $O(\sqrt{tn}+\sqrt{n\log(1/\varepsilon)})$. We need to show that this gives error probability at most $\leq \varepsilon$ for every input $x \in \{0,1\}^n$. Observe the following:

- if step 1 found t solutions, then we know $|x| \geq t$ with probability 1 (note that you can verify whether a given position is a solution with only 1 extra query).
- if step 1 found fewer than t solutions, but step 2 found another solution, then we know $|x| > t$ (for if $|x| \leq t$ then step 1 would certainly have found all solutions and there would be none left to be found in step 2).
- if step 1 found fewer than t solutions, but step 2 did *not* find another solution, then the probability that there are more solutions than those found by step 1, is at most $\varepsilon/2$ (because step 2 ran an $\varepsilon/2$-error search algorithm which didn't find any solution); hence in this case we assume step 1 has found all solutions.
- similar observations for steps 3 and 4 (with 0s and 1s switching roles).

These observations imply that at the end of the 4 steps we have enough information to compute f. Note that with probability at least $1-\varepsilon$ we can distinguish between the three cases $|x| < t$, $|x| \in \{t,\ldots,n-t\}$, and $|x| > n-t$. If $|x| \in \{t,\ldots,n-t\}$ then we are done because f is constant on this interval. If $|x| < t$ then step 1 found all solutions, so we know x completely and can compute $f(x)$. If $|x| > n-t$ then step 3 found all non-solutions of x, and again we know x completely. In all cases we compute $f(x)$ with error probability at most ε.

This algorithm even works for many other functions $f : \{0,1\}^n \to [0,1]$: it suffices if f is constant on all inputs with Hamming weight in $\{t,\ldots,n-t\}$; f may be arbitrary if $|x| < t$ or $|x| > n-t$ since in these cases the algorithm actually determines x completely, rather than just its Hamming weight. Once it knows x, it can just output a random bit whose probability of being 1 equals $f(x)$.

2.2 Lower bound on $deg_\varepsilon(f)$

We can assume $t < n/4$, because if $t \geq n/4$ then we already have a tight bound from Paturi:

$$n \geq deg(f) \geq deg_\varepsilon(f) \geq deg_{1/3}(f) = \Theta(n).$$

Buhrman et al. [10] showed for the n-bit OR function that $deg_\varepsilon(\mathrm{OR}_n) = \Theta(\sqrt{n\log(1/\varepsilon)})$. For completeness we include a proof of this in Appendix B.[a] Since $t < n/4$, we can embed an OR on at least $n-2t \geq n/2$ bits into f by fixing at most $2t$ of the n input bits to specific values. Hence

$$deg_\varepsilon(f) \geq \max\left(deg_{1/3}(f), \Omega(\sqrt{n\log(1/\varepsilon)})\right) = \Omega\left(deg_{1/3}(f) + \sqrt{n\log(1/\varepsilon)}\right).$$

[a] The earlier paper by Kahn et al. [17] showed a $\widetilde{\Theta}$-version of this bound.

Acknowledgments

Thanks to Sasha Sherstov for his paper [31] (which prompted this note) and some comments. This work was partially supported by a Veni grant from the Netherlands Organization for Scientific Research (NWO), and by the European Commission under the Integrated Project Qubit Applications (QAP) funded by the IST directorate as Contract Number 015848.

References

1. S. Aaronson. Lower bounds for local search by quantum arguments. In *Proceedings of 35th ACM STOC*, pages 465–474, 2003. quant-ph/0307149.
2. S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. In *Proceedings of the Royal Society*, volume A461(2063), pages 3473–3482, 2005. quant-ph/0412187.
3. D. Aharonov and O. Regev. A lattice problem in quantum NP. In *Proceedings of 44th IEEE FOCS*, pages 210–219, 2003. quant-ph/0307220.
4. D. Aharonov and O. Regev. Lattice problems in NP∩coNP. In *Proceedings of 45th IEEE FOCS*, pages 362–371, 2004.
5. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049.
6. R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference*, pages 82–95, 1993.
7. R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Computational Complexity*, 4:339–349, 1994.
8. R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191–202, 1995.
9. G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.
10. H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of 40th IEEE FOCS*, pages 358–368, 1999. cs.CC/9904019.
11. H. Buhrman and R. de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of 16th IEEE Conference on Computational Complexity*, pages 120–130, 2001. cs.CC/9910010.
12. H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.
13. D. Coppersmith and T. J. Rivlin. The growth of polynomials bounded at equally spaced points. *SIAM Journal on Mathematical Analysis*, 23(4):970–983, 1992.
14. L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999. Earlier version in Complexity'98. Also cs.CC/9811023.
15. M. de Graaf and R. de Wolf. On quantum versions of the Yao principle. In *Proceedings of 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS'2002)*, volume 2285 of *Lecture Notes in Computer Science*, pages 347–358. Springer, 2002. quant-ph/0109070.
16. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.
17. J. Kahn, N. Linial, and A. Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16(4):465–477, 1996.
18. I. Kerenidis. Quantum multiparty communication complexity and circuit lower bounds. In *Proceedings of 4th Annual Conference on Theory and Applications of Models of Computation (TAMC)*, volume 4484 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 2007. quant-ph/0504087.
19. I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69(3):395–420, 2004. Special issue on STOC'03. quant-ph/0208062.
20. S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and classical formula

size lower bounds. In *Proceedings of 20th IEEE Conference on Computational Complexity*, pages 76–90, 2005. quant-ph/0501057.
21. T. Lee and A. Shraibman. Disjointness is hard in the multi-party number-on-the-forehead model. In *Proceedings of 23rd IEEE Conference on Computational Complexity*, 2008. arXiv:0712.4279.
22. N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990. Earlier version in STOC'90.
23. R. Lipton, E. Markakis, A. Mehta, and N. Vishnoi. On the Fourier spectrum of symmetric Boolean functions with applications to learning symmetric juntas. In *Proceedings of 20th IEEE Conference on Computational Complexity*, pages 112–119, 2005.
24. M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1968. Second, expanded edition 1988.
25. E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. Earlier version in STOC'03.
26. N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. Earlier version in STOC'92.
27. R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proceedings of 24th ACM STOC*, pages 468–474, 1992.
28. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. *Mathematical notes of the Academy of Science of the USSR*, 41(4):333–338, 1987.
29. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Sciences, mathematics*, 67(1):159–176, 2003. quant-ph/0204025.
30. T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Wiley-Interscience, second edition, 1990.
31. A. Sherstov. Approximate inclusion-exclusion for arbitrary symmetric functions. In *Proceedings of 23rd IEEE Conference on Computational Complexity*, 2008.
32. A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *Proceedings of 40th ACM STOC*, 2008.
33. R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of 19th ACM STOC*, pages 77–82, 1987.
34. S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.
35. S. Wehner and R. de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *Proceedings of 32nd ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1424–1436, 2005. quant-ph/0403140.
36. R. de Wolf. Lower bounds on matrix rigidity via a quantum argument. In *Proceedings of 33rd ICALP*, volume 4051 of *Lecture Notes in Computer Science*, pages 62–71, 2006. quant-ph/0505188.

Appendix A: Grover's Algorithm and Applications

A.1 Finding one solution

Grover's quantum algorithm [16] for finding a solution (i.e. an $i \in [n]$ such that $x_i = 1$) consists of T applications of a certain unitary G, starting from the uniform superposition $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle$. We won't explain the details of G here. Suffice it to say that each G makes one quantum query, so the total number of queries is T. The intuition is that G changes the state by moving amplitude from non-solutions to solutions. One can show [9] that the probability that a measurement of the state after T steps gives a solution, is exactly

$$(\sin((2T+1)\theta))^2, \text{ where } \theta = \arcsin(\sqrt{|x|/n}).$$

If $|x| > 0$ and $T = \lceil (\pi/4)\sqrt{n/|x|} \rceil$, then this probability is close to 1. Hence if we know (at least approximately) the number of solutions $|x|$, then we can find one with good probability

using $O(\sqrt{n/|x|})$ queries. If we know $|x|$ exactly, a small modification of the algorithm finds a solution *with probability 1* [9]. This uses exactly $\lceil(\pi/4)\sqrt{n/|x|}\rceil$ queries; we will refer to it as "exact Grover".

What if we don't know how many solutions there are in the input? We can first apply Grover assuming the number of solutions is $n/2$, then assuming it is $n/4$ etc. This finds one solution with probability at least some constant, even if we don't know the number of solutions. The complexity is $\sum_{i=1}^{\log n} O(\sqrt{n/2^i}) = O(\sqrt{n})$ queries. If we know there are *at least* t solutions, this can be improved to $O(\sqrt{n/t})$. We will refer to this as "usual Grover".

And what if we want to have probability at least $1-\varepsilon$ of finding a solution? Buhrman et al. [10] designed an algorithm that achieves this using $O(\sqrt{n\log(1/\varepsilon)})$ queries, and showed (by proving the lower bound on $deg_\varepsilon(\mathrm{OR})$ mentioned in Section 2.2) that this complexity is optimal up to a constant factor. Their algorithm is quite simple. Apply exact Grover $\log(1/\varepsilon)$ times, first assuming there is 1 solution, then assuming there are 2 solutions, then assuming there are 3 solutions, etc. If the actual number of solutions is between 1 and $\log(1/\varepsilon)$, at least one solution will have been found with probability 1 by now. If no solution has been found yet, then apply usual Grover $O(\log(1/\varepsilon))$ many times assuming there are at least $t = \log(1/\varepsilon)$ solutions. It is easy to verify that this has overall query complexity $O(\sqrt{n\log(1/\varepsilon)})$ and error probability at most ε. We will refer to this as "ε-error Grover".

A.2 Finding all solutions

De Graaf and de Wolf [15, Lemma 2] observed that exact Grover can be used to *find all solutions with probability 1*, as long as we know an *upper bound* t on the number of solutions. Suppose we run exact Grover t times: the first time assuming we have exactly t solutions, the second time assuming we have exactly $t-1$ solutions, etc. Each time we find a solution i, we "cross it out" in the sense of modifying the input by setting x_i to 0 (this can easily be achieved by some unitary pre- and post-processing around the query). This prevents the algorithm from finding the same solution twice. The total number of queries used is

$$\sum_{i=1}^{t} \lceil(\pi/4)\sqrt{n/i}\rceil \approx \frac{\pi}{2}\sqrt{tn}.$$

To see that this finds all solutions with probability 1, observe that the assumed number of solutions $t-i+1$ of the ith run always upper bounds the actual number of remaining solutions (this "loop invariant" is easily proved with downward induction). Hence if we start with at most t remaining solutions, then after t runs we end with 0 solutions—meaning all solutions have been found.

Appendix B: The BCWZ Lower Bound for $deg_\varepsilon(\mathrm{OR}_n)$

Here we give a brief proof of the $\Omega(\sqrt{n\log(1/\varepsilon)})$ degree-bound for ε-approximations of the n-bit OR function from [10]. By symmetrization, it suffices to lower bound the degree d of a single-variate polynomial p with the following properties:

$$p(0) \in [-\varepsilon,\varepsilon] \text{ and } p(i) \in [1-\varepsilon, 1+\varepsilon] \text{ for all } i \in \{1,\ldots,n\}.$$

Defining $q(x) = (1-p(n-x))/\varepsilon$, we get a degree-$d$ polynomial satisfying

$$q(n) \geq 1/\varepsilon - 1 \text{ and } |q(i)| \leq 1 \text{ for all } i \in \{0,\ldots,n-1\}.$$

By a result of Coppersmith and Rivlin [13, p. 980], we can extend the latter bound to a bound on the real interval: there exist absolute constants $a, b > 0$ such that

$$|q(x)| \leq ae^{bd^2/(n-1)} \text{ for all } x \in [0, n-1].$$

We now rescale the domain $[0, n-1]$ to $[-1, 1]$. Define

$$r(x) = (q((x+1)(n-1)/2) - 1)/(ae^{bd^2/(n-1)}).$$

This polynomial has degree d and satisfies

$$|r(x)| \leq 1 \text{ for all } x \in [-1, 1], \text{ and } r(1+\mu) = q(n)/(ae^{bd^2/(n-1)}) \geq (1/\varepsilon - 1)/(ae^{bd^2/(n-1)})$$

for $\mu = 2/(n+1)$. In other words, r is bounded by 1 on the interval $[-1, 1]$ and grows fast to the right of that interval. It is known that among all degree-d polynomials bounded by 1 on the interval $[-1, 1]$, the fastest-growing is the Chebyshev polynomial of the first kind [30], defined by:

$$T_d(x) = \frac{1}{2}\left(\left(x + \sqrt{x^2-1}\right)^d + \left(x - \sqrt{x^2-1}\right)^d\right).$$

Hence we have

$$r(1+\mu) \leq T_d(1+\mu) \leq \left((1+\mu) + \sqrt{(1+\mu)^2 - 1}\right)^d \leq \left(1 + 2\sqrt{2\mu + \mu^2}\right)^d \leq e^{2d\sqrt{2\mu+\mu^2}}.$$

Combining upper and lower bounds on $r(1+\mu)$ and rearranging gives $d = \Omega(\sqrt{n \log(1/\varepsilon)})$.