Some new results on assessment of $Q-$gram filter efficiency

Andrej Novak*, Krešimir Križanović†, Alen Lančić‡ and Mile Šikić§
*†§University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
Email: andrej.novak@fer.hr, kresimir.krizanovic@fer.hr, mile.sikic@fer.hr
‡University of Zagreb, Faculty of Science, Department of Mathematics, Croatia.
Email: alen@student.math.hr

***Abstract*—We present some new results on expectation and threshold problem for contiguous and gapped $Q-$grams. We also introduce a novel exhaustive search algorithm for determining the threshold, based on (sparse) matrix formulation of the problem. Computational results obtained with this algorithm show that the $Q-$gram filter has same threshold in the case of Hamming and Levenshtein errors given that the number of errors is sufficiently small.**

I. Introduction

One of the oldest problems in computer science is finding an approximate string matching of a relatively short string called pattern in a long string called text. It finds applications in various areas such as bioinformatics, text processing, pattern recognition and signal processing (see [9], [10], [11]). For these reasons, fast practical algorithms for approximate string matching are still in high demand. There are several variants of the approximate string matching problem but in this paper we are interested in the problem of finding the distance between two strings.

Let P and S be two strings such that $d(S, P) = k$. In this paper we are going to present some theoretical and computational results in the case when d is Hamming or Levenshtein distance. Levenshtein distance leads to the k difference problem, the minimum number of single character insertions, deletions and replacements required to transform one string into another. A simpler variant (Hamming distance) is the k mismatches problem, which does not allow insertions or deletions that makes it the number of non-matching characters between two strings of the same length. Hamming distance can be used in practice when Levenshtein distance is difficult to calculate and, as we will see in Section IV, filter performance will be the same when k is small enough. It is useful to imagine strings P and S in a manner that string S is a string derived from P by introducing k errors.

One way to speed up approximate string matching is use of filters. A filter is an algorithm that quickly discards large parts of the text based on some filter criterium. Filters usually have preprocessing phase in which they build a data structure on the text (an index) and use it afterward to speed up searches. Many filters are based on Q-grams, substrings of length Q. The $Q-$gram similarity of two strings is defined as the number of Q-grams shared by these strings. Paper by Ukkonen [1] presented the algorithm that finds the locally best approximate occurrences of pattern string in text using $Q-$gram similarity as a measure. Its time complexity is linear with respect to the text length. This algorithm takes into account only contigous $Q-$grams. A generalization of the contiguous $Q-$grams is given in the work by Burkhardt and Karkkainen [2] in a form of gapped $Q-$grams. Gapped $Q-$gram is a subsets of Q characters of a fixed noncontiguous shape. For example, the $Q-$grams of shape $xx - x$ in the string ACAGCT are ACG, CAC and AGT. Burkhard and Karkkainen showed in [2] that gapped $Q-$grams can provide orders of magnitude faster and more efficient filtering than contiguous $Q-$grams. The same authors also showed in [3] how to extend one-gapped $Q-$grams for Levenshtein errors.

Beside theoretical estimates on $Q-$gram filter efficiency, in this paper, we propose a novel algorithm for computing the threshold of $Q-$gram filter in the case when d is Hamming distance function. Threshold gives the minimum number of $Q-$grams that an approximate match must share with the pattern. Our algorithm is based on sparse matrix formulation of the threshold problem and that makes it suitable for parallel implementation. Also, the most expensive step of the algorithm is matrix-to-matrix multiplication what can be done, bellow quadratic time (for one entry in resulting matrix) because corresponding matrices are sparse.

The paper is organized as follows. After the introduction, we introduce basic notation on $Q-$grams and state some generalizations of existing results on expected $Q-$gram similarity of two strings. Than we introduce a novel algorithm for computing the threshold and estimate its time complexity. Finally, we conclude the paper with computational results obtained with the implementation of the proposed algorithm and few comments that illustrate situations when Hamming errors are a good approximation for Levenshtein errors in the context of $Q-$grams.

II. Mathematical formalism

Efficiency of a $Q-$gram filter can be determined by looking at the number of $Q-$grams without error. In our work we considered two basic measures, expectation and threshold. To precisely define our theoretical and experimental work, we first need to introduce some basic notation for $Q-$grams.

Let $I \subset \mathbb{N} \cup \{0\}$ be a finite set, i.e. the cardinal number of I is finite. The span of I is defined as $span(I) = max(I) - min(I) + 1$.

The shape of I is the set $Q = \{i - \min(I) : i \in I\}$. For a shape Q and a natural number $i \in \mathbb{N}$ we define set $Q_i = \{i + j : j \in Q\}$.

Given a string $S = s_1 s_2 \dots s_m$ and shape Q with corresponding sets $Q_i = \{i_1, i_2, \dots, i_q\}$, $i \in \{1, 2, \dots, m - \text{span}(Q) + 1\}$, we can define $Q-$gram on string S, at position i, based on shape Q as $S[Q_i] = s_{i_1} s_{i_2} \dots s_{i_q}$.

Definition 2.1: If $|Q| = \text{span}(Q)$ than shape Q is contiguous, otherwise we say that Q is gapped shape.

We can observe that contiguous shapes generate contiguous $Q-$grams and gapped shapes generate gapped $Q-$grams.

$Q-$grams can be used to define a measure of similarity between two strings. $Q-$gram similarity $S_q(P, S)$ for strings P and S is the number of their common $Q-$grams, that is

$$s_Q(P, S) = |\{i \in [1, m - \text{span}(Q) + 1] : P(Q_i) = S(Q_i)\}|.$$

Example 2.2: Let $Q = \{0, 2, 3\}$ be a shape. Using previous notation we see that $|Q| = 3$, and $\text{span}(Q) = 4$. The string $S =$ `ACGACCGTA` has six $Q-$grams $S[Q_1] =$ `AGA`, $S[Q_2] =$ `CAC`, $S[Q_3] =$ `GCC`, $S[Q_4] =$ `ACG`, $S[Q_5] =$ `CGT`, and $S[Q_6] =$ `CTA`. If we define $P =$ `AGGATCGTA`, than Hamming distance $d(S, P) = 2$, but the $Q-$gram similarity of P and S is $s_Q = 3$ since P and Q share three common $Q-$grams $S[Q_1]$, $S[Q_4]$, and $S[Q_6]$.

Let us note that we can look at strings S and P, in the above example, little differently. We can consider string S as the original string, and we can say that string P is gained from string S by introducing k errors, or in this case mutations. This could correspond, for example, to genomes of two viruses where one of them has mutated. In this case, the expected similarity between S and P actually represents the number of "good" $Q-$grams in string P (i.e. $Q-$grams without errors).

A. Expectation

Expectation of the number of $Q-$grams without errors is one of two filter efficiency measures that we considered in our work. In the sequel we present some new results.

Proposition 2.3: Let S and P be given strings such that $d(S, P) = k$, where d is Hamming distance. Then the expected $Q-$gram similarity between S and P is

$$E = (|S| - \text{span}(Q) + 1)(1 - k/|S|)^{|Q|}.$$

*P*roof. As we mentioned before, we can compare characters $P[i]$ and $S[i]$, $i \in \{1, 2, \dots, |S|\}$ and mark k places where $P[i] \neq S[i]$ by a special character in one of the strings, say S. We can than work only with one string keeping in mind that the special character codes for error. Define an indicator random variable X_i, $i \in \{1, \dots, |S| - \text{span}(Q) + 1\}$ for $i-$th $Q-$gram. That is, $X_i = 1$ if $S[Q_i]$ does not contain special character, else $X_i = 0$. Let $z = |S| - \text{span}(Q) + 1$, now $Y = \sum_{i=1}^{z} X_i$ is wanted random variable. By using the linearity of expectation we obtain

$$\begin{aligned} E[Y] = E[\sum_{i=1}^{z} X_i] &= \sum_{i=1}^{z} E[X_i] = \sum_{i=1}^{z} (1 - k/|S|)^{|Q|} \\ &= (S| - \text{span}(Q) + 1)(1 - k/|S|)^{|Q|}. \end{aligned}$$

■

We can go one step further towards a potential $Q-$gram filter application and determine a similar formula in the case when, besides mismatch, deletions and insertions are also allowed. In the statement of the following proposition, instead of a fixed number of errors (of each type), we use probabilities for mismatch, deletion and insertion. Note that the same could have been done in Proposition 2.3. by defining the probability $p = k/|S|$ that a randomly chosen character in P is different from the character on the same position in S.

Proposition 2.4: Let S be a string and let p, r and s be appropriate probabilities of mismatch, deletion and insertion, respectively. The expected number of contiguous $Q-$grams without error is

$$E = (|S| - |Q| + 1)(1 - p - s - r)^{|Q|}.$$

*P*roof. Define indicator random variable X_i $i \in \{1, \dots, |S| - |Q| + 1\}$ for $i-$th $Q-$gram. That is, $X_i = 1$ if $S[Q_i]$ does not contain either a mismatch or inserted character and none of its Q characters has been deleted, else $X_i = 0$. Then, $P(X_i = 1) = (1 - p - s - r)^{|Q|}$ is the probability that given character is not a mismatch or that it has not been inserted. Let $z = |S| - |Q| + 1$, now $Y = \sum_{i=1}^{z} X_i$ is wanted random variable. Using the linearity of expectation as in the proof of the Proposition 2.3 we obtain the result. ■

Until now, we have considered only $Q-$grams that do not contain any error to be "good". If we weaken that condition and consider a $Q-$gram to be good if it contains at most $\varepsilon \geq 0$ errors of Hamming type, we can deduce the following result.

Proposition 2.5: Let S be a string and let p be the probability of a mismatch. Expected number of $Q-$grams with up to ε errors is given by

$$E = (|S| - \text{span}(Q) + 1) \sum_{l=0}^{\varepsilon} \binom{\text{span}(Q)}{l} (1 - p)^{\text{span}(Q) - l} p^l.$$

*P*roof. Repeat the proof of Proposition 2.4. with random variable X_i such that $X_i = 1$ if $S[Q_i]$ contains at most ε errors and $X_i = 0$ otherwise.
This gives $P(X_i = 1) = \sum_{l=0}^{\varepsilon} \binom{\text{span}(Q)}{l} (1 - p)^{\text{span}(Q) - l} p^l$. Proceeding as before and using the linearity of expectation we obtain the result. ■
Although theoretically interesting, expectation is a rough method of measuring $Q-$gram efficiency.

B. Threshold

In this section we will consider another measure of the $Q-$gram filter efficiency. If we keep the original string size and the number of errors constant, the threshold is defined as the minimum number of $Q-$grams without errors over all possible error distributions. In the case of contiguous shape Q, the threshold can be determined from the well known $Q-$gram lemma.

Lemma 2.6: Let P and S be given strings such that $d(P, S) = k$, where d can be either Hamming or Levenshtein distance. Let Q be a contiguous shape. The $Q-$gram similarity of P and S is at least

$$t = max\{|S| - |Q| + 1 - k|Q|, 0\}.$$

■

If the number of errors is small enough we can generalize the previous formula to include gapped shapes. More precisely, the following holds:

Lemma 2.7: Let P and S be strings such that $d(P,S) = k$ in Hamming or Levenshtein distance where $k < \min(|P|,|S|)/\text{span}(Q)$. Then, $Q-$gram similarity of P and S is at least:

$$t = \max\{|S| - \text{span}(Q) + 1 - k \cdot \text{span}(Q), 0\}. \quad (1)$$

■

In general, last formula can be considered as a lower bound for the threshold. With a low error rate, the formula actually gives an exact threshold value. However, as the error rate increases, it becomes overly pessimistic, giving values that are significantly lower then the actual threshold.

Example 2.8: Let $Q = \{0,2,3\}$ and $S =$ `AxCTGx` be given. Formula (1) yields $t = \max\{6-4+1-3\cdot 2, 0\} = 0$, but we have $S[Q_1] =$ `ACT` unaffected by errors, and therefore $t = 1$.

We were unable to provide a general closed formula for calculating the threshold. Instead of a closed formula, we will propose an algorithm for computing it.

III. MatMat threshold algorithm

Since [1] was published, several authors proposed algorithms for computation of threshold [2], [8], [7]. In a way, majority of them is based on dynamic programming. The essence of our algorithm is exhaustive search based on the matrix formulation of the problem and sparse structure of the corresponding matrices. This approach is useful if one wants to compute the threshold of multiple $Q-$grams while keeping the length of the string and the number of errors fixed.

Definition 3.1: Let $\mathbf{X}$ and $\mathbf{Y}$ be two matrices such that their matrix product $\mathbf{Z} = \mathbf{XY}$ is well defined. A defect product $\mathbf{Z} = \mathbf{X} \boxtimes \mathbf{Y}$ is a matrix of zeros and ones whose elements are defined by the following formula

$$z_{ij} = \begin{cases} 1, & if \quad \sum_{k=1}^{n} x_{ik} y_{kj} > 0, \\ 0, & otherwise. \end{cases}$$

A. Problem definition

To describe the matrix formulation of the threshold problem first we need to define two matrices: $Q-$gram matrix $\mathbf{Q}$ and error matrix $\mathbf{E}$. $Q-$gram matrix $\mathbf{Q}$ encodes the movement of the given shape Q along the string being analysed, while error matrix $\mathbf{E}$ encodes all possible error distributions (k errors in a string of length n).

First, we will define a vector $q = [q_i]$, of length n. For a given shape Q, vector q is defined by

$$q_i = \begin{cases} 1, & i-1 \in Q, \\ 0, & otherwise, \end{cases}$$

where $i = 1,2,\ldots,|S| - \text{span}(Q) + 1$.

$Q-$gram matrix $\mathbf{Q}$ has $(n - \text{span}(Q) + 1)$ rows and n columns. The i^{th} row of matrix $\mathbf{Q}$ corresponds to the vector q such that $\mathbf{Q}(i, i : i + \text{span}(Q) - 1) = q$.

Error matrix $\mathbf{E}$ has n rows and $\binom{n}{k}$ columns. Its columns represent all possible combinations of $n-k$ zeroes and k ones. Each column represents one error distribution, where zeroes represent "good" characters, while ones represent errors.

It is significant to note that, due to the fact that both k and $|Q|$ are a lot smaller than n, in general both matrices $\mathbf{Q}$ and $\mathbf{E}$ are sparse.

B. Computing the threshold

The threshold for a given $Q-$gram can be computed from the defect product of matrices $\mathbf{Q}$ and $\mathbf{E}$.

Proposition 3.2: The threshold for $Q-$gram shape Q can be calculated as

$$t = n - \text{span}(Q) + 1 - \|\mathbf{C}\|_1,$$

where $\mathbf{C} = \mathbf{Q} \boxtimes \mathbf{E}$. ■

One row of matrix $\mathbf{Q}$ represents one $Q-$gram of original string S and one column of matrix $\mathbf{E}$ represents positions of the errors. Within the defect product of two matrices, the element gained from a single row of matrix $\mathbf{Q}$ and single column of matrix $\mathbf{E}$ represents whether corresponding $Q-$gram will contain errors applied to a corresponding error distribution. The matrix $\mathbf{C}$, obtained as the defect product of $\mathbf{Q}$ and $\mathbf{E}$, will have $(n-span(Q)+1)$ rows (one for each $Q-$gram in original string) and $\binom{n}{k}$ columns (one for each error distribution). Each column of matrix $\mathbf{C}$ will have zero on positions where $Q-$grams contain no errors, and one on positions where $Q-$grams contain one or more errors. The column with the most ones will represent the worst case error distribution, having the least number of $Q-$grams without errors. The threshold can be calculated by finding the column with the most ones, summing up all ones to get the number of erroneous $Q-$grams and subtracting that number from the total number of $Q-$grams. This is exactly what the formula presented above does.

C. Algorithm complexity

In order to compute the time complexity let us remark that defect product of one row of matrix $\mathbf{Q}$ by one column of matrix $\mathbf{E}$ can be computed in $\mathcal{O}(|Q|))$ operations rather than in $\mathcal{O}(2|Q|^2 - |Q|)$ as a standard matrix to vector multiplication. That gives $\mathcal{O}(|Q|(n - \text{span}(Q) + 1))$ for computation of one column of matrix $\mathbf{C}$, and in general $\mathcal{O}(|Q|(n - \text{span}(Q) + 1)\binom{n}{k})$ for the whole matrix $\mathbf{C}$. Constant $|Q|$ can be further decreased with the reduction of the dimension of matrix $\mathbf{E}$ if symmetric error distributions are omitted.

Definition 3.3: We say that vectors $v = (v_1, v_2, \ldots, v_n)$ and $u = (u_1, u_2, \ldots, u_n)$ are symmetric if $v_i = u_{n-i}$, $i = 1,2,\ldots,n$.

We can now easily conclude the following.

Lemma 3.4: Let u and v be symmetric vectors. Than $\|C \boxtimes u\|_1 = \|C \boxtimes v\|_1$. ■

We see that symmetric (distribution of errors) columns of matrix $\mathbf{E}$ yield the same number of $Q-$grams with out error and therefore the number of columns can be reduced by half resulting in the equal reduction of constant $|Q|$ in the

complexity formula. Finally, time complexity of MatMat is given by $\mathcal{O}\left(\frac{|Q|}{2}(n-\text{span}(Q)+1)\binom{n}{k}\right)$.

IV. Computational results

In this section we present results obtained by implementing the MatMat algorithm for determining the threshold described in the previous section. Of our particular interest were one and two gapped $Q-$grams. Results are presented in Table 1 and Table 2 bellow. In the case of Hamming errors (mismatch only), thresholds were computed with MatMat and in the case of Levenshtein errors (mismatch, insertions and deletions) exhaustive search over all possible scenarios was applied. It is interesting to note that when k is small enough, there is no difference in threshold between Hamming and Levenshtein errors. That supports our claim that Hamming errors are a good model for simulating real world situations (Levenshtein errors) if the probability of error is small enough.

TABLE I: Shapes with one gap applied to the string with $n = 50$. Only Hamming errors are considered.

Shape	k	Threshold
{0,2,3,4,5,6,7,8,9}	3	14
	4	5
	5	1
	6	0
{0,1,3,4,5,6,7,8,9}	3	14
	4	5
	5	2
	6	0
{0,1,2,4,5,6,7,8,9}	3	14
	4	5
	5	3
	6	1
	7	0
{0,2,3,4,5,6,7,8,9}	3	14
	4	5
	5	3
	6	0

TABLE II: Gapped shapes applied to a string with $n = 50$. Threshold H is computed on a string with Hamming errors and Threshold L is computed on a string with Levenshtein errors.

Shape	k	Threshold H	Threshold L
{0,2,3,4,6,7,8,9}	3	17	17
	4	9	9
	5	6	2
	6	4	0
	7	2	0
	8	0	0
{0,1,3,5,6,7,8,9}	0	41	41
	1	33	33
	2	25	25
	3	17	17
	4	9	9
	5	6	2
	6	3	0
	7	2	0
	8	0	0

Furthermore, we have performed tests to find out the probability mass function of a number of $Q-$grams without errors with respect to a fixed numbers of errors in a string. For each number of errors k we have performed $5 \cdot 10^6$ experiments in which random positions of errors were chosen and the number of $Q-$grams without errors was determined. By interpolating the computed data with the function $f(x) = x^a e^{-bx+c}$, $a, b, c \in \mathbb{R}$ we have managed to obtain coefficient of determination R-square greater than 0.998 and $RMSE < 10^{-3}$. Results for shape $Q = \{0, 1, 3, 5, 6, 7, 8, 9\}$ are presented on Figure 1 and Table 3.

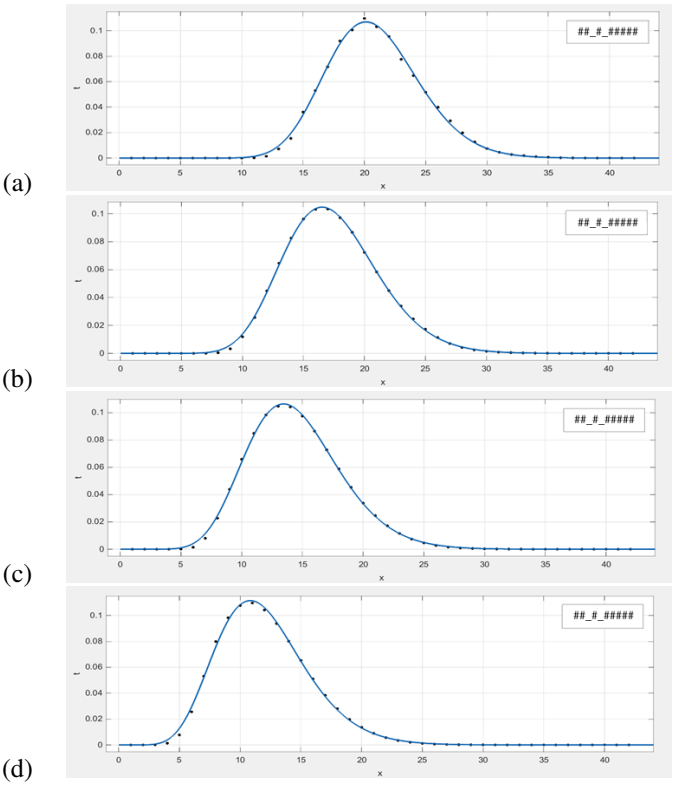

Fig. 1: The probability mass function for shape $Q = \{0, 1, 3, 5, 6, 7, 8, 9\}$ and its interpolation are represented by black dots and a continuous blue line, respectively. Length of string is $n = 50$ and number of errors are
(a) k = 4, (b) k = 5, (c) k = 6, (d) k = 7.

TABLE III: Parameters obtained by fitting the experimental data for shape $Q = \{0, 1, 3, 5, 6, 7, 8, 9\}$

k	a	b	c	RMSE	R-squared
4	29.08	1.444	-60.46	0.001449	0.9983
5	18.81	1.137	-36.21	0.0007293	0.9996
6	12.84	0.9549	-22-77	0.000936	0.9993
7	9.209	0.8498	-14.93	0.001437	0.9985

V. Conclusion and future work

In this paper we have extended some known analytical results on contiguous and gapped $Q-$grams. Motivated by the problem of assessing $Q-$grams, we also proposed a novel algorithm for computing the threshold. Advantages of the proposed algorithm are in its simple implementation, possible parallelization, and fast sparse matrix to matrix multiplications. Disadvantages, on the other hand, are large dimensions of the matrices involved in computation. In our future work we plan to adopt presented algorithm for Levenshtein errors.

VI. Funding

This work has been supported in full by Croatian Science Foundation under the project 7353 Algorithms for Genome Sequence Analysis.

REFERENCES

[1] Ukkonen, Esko, *Approximate string-matching with q-grams and maximal matches.* Theoretical computer science 92.1 (1992): 191-211.

[2] Burkhardt, Stefan, and Juha Karkkainen, *Better filtering with gapped q-grams.* Fundamenta informaticae 56.1 (2003): 51-70.

[3] Burkhardt, Stefan, and Juha Karkkainen, *One-gapped q-gram filters for Levenshtein distance.* Combinatorial pattern matching. Springer Berlin Heidelberg, 2002.

[4] Le Gall, Franois, *Faster algorithms for rectangular matrix multiplication.* Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on. IEEE, 2012.

[5] Gravano, Luis, et al., *Using q-grams in a DBMS for approximate string processing.* IEEE Data Eng. Bull. 24.4 (2001): 28-34.

[6] Sutinen, Erkki, and Jorma Tarhio, *On using q-gram locations in approximate string matching.* Lecture Notes in Computer Science, Vol. 979, 1995, pp 327-340 . Springer.

[7] Krkkinen, Juha, *Computing the threshold for q-gram filters*, Algorithm TheorySWAT 2002. Springer Berlin Heidelberg, 2002. 348-357.

[8] Rasmussen, Kim et. al., *Efficient q-Gram Filters for Finding All ε-Matches over a Given Length.* Lecture Notes in Computer Science, Vol. 3500, Springer, (2005)

[9] Ma, Bin, Tromp, John., and Li, Ming, *PatternHunter: faster and more sensitive homology search.* Bioinformatics 18.3 (2002): 440-445.

[10] Buchfink, Benjamin, Chao Xie, and Daniel H. Huson. *Fast and sensitive protein alignment using DIAMOND.* Nature methods 12.1 (2015): 59-60.

[11] Weese, David, et al. *RazerSfast read mapping with sensitivity control.* Genome research 19.9 (2009): 1646-1654.