On the Collapse of the q-Gram Filtration *

E. SUTINEN
University of Helsinki, Finland

W. SZPANKOWSKI
Purdue University, U.S.A.

Abstract

In the approximate pattern matching problem, the text area to be searched for an occurrence of a pattern can be pruned by applying a filtration condition. A q-gram based filtration condition defines potential text areas in terms of pattern q-grams, i.e., strings of length q. A text area will be checked by an accurate method only if the set of the q-grams in the text area satisfies a certain condition. One hopes that the filtration limits the number of checks to a minimum, thus making the algorithm quite efficient. However, computer experiments show that the filtration method works fine for cases when the allowed error level k is relatively small compared to the pattern length, but loses its efficiency quite sharply with an increasing k. This is a *phase transition phenomenon* that is quite often observed in nature. In this paper, we present a theoretical explanation for this phenomenon which will excuse us to introduce advanced mathematical analysis based on certain languages, correlation polynomials, generating functions and complex analysis. It is our view that nothing can be more exciting and rewarding than finding a theoretical justification for an abrupt manifestation of nature.

Keywords: Algorithm Analysis, Approximate Pattern Matching.

1 Introduction

The *approximate pattern matching problem* can be stated as follows: Given a *text* $T = T[1 \dots n]$ and a *pattern* $P = P[1 \dots m]$ built over an *alphabet* Σ, and an *error level* k, the task is to locate (the end positions of) all the k*-approximate matches* of P in T. A substring P' of T is a k-approximate match of P if P' can be transformed to P with at most k errors which might be interpreted as *edit operations* (insertions, deletions, or changes) or Hamming distance or square error distance (e.g., for image compression; cf. [11]).

*This work has been supported NSF Grants NCR-9206315 and NCR-9415491, NATO Collaborative Grant CRG.950060, and Grant 22586 from Academy of Finland.

Approximate pattern matching algorithms are typically used on long texts, like biosequences [1], hence it is crucial to find time-efficient solutions to the problem. The basic solutions, based on dynamic programming, work in $O(kn)$ time [4, 6, 10, 18]. Because of the text size n, this is usually unacceptable. More efficient algorithms work in two phases: the first phase, called *filtration*, finds potential matches, which are later verified by the second phase.

One way to recognize potential matches is to extract a set of q-*grams*, i.e., substrings of length q, from the pattern and compare this set to another set of q-grams, picked from a scanned text area (i.e., a window of size m). If the sets share a sufficient amount of common q-grams, the corresponding text area will be marked as potential. The diverse techniques of this approach [3, 9, 15, 16, 17] belong to the family of the q-gram *filtration algorithms*.

There is an interesting phenomenon in the behavior of the q-gram filtration methods: For relatively small error levels k, compared to the pattern length m, the filtration phase marks only very few text areas which in the second phase of the algorithm turn out not to contain matches; we call these *false matches*. However, for a certain error level k_d, the filtration phase loses its efficiency almost totally, that is, there is an abrupt change in the efficiency of the algorithm. In fact, for error levels of at least k_d, the filtration phase marks practically all the text as potential. In this respect, the q-gram filtration algorithm family shares the classical pattern of humoristic behavior: a phenomenon changes in an unexpected way, like Donald Duck running out of window and only outside figuring out his desperate situation.

This phase transition behavior of the q-gram filtration has not yet been theoretically explained; an indication of true humor. However, the reason for the collapse of the pruning accuracy can be justified in descriptive terms: a larger error level k means that a k-approximate match has less similarity with the original pattern, measured as common q-grams. This means that the filtration finds text areas which do have same q-grams as in the pattern. Thus, a potential match is a false one, with "broken" information (cf. Figures 1 and 2).

In this paper, we derive a formula which predicts accurately the behavior of a q-gram based filtration method. Our results holds in a general probabilistic framework: We could assume that the text T is generated according to a Markov model, however, for simplicity of presentation we restrict ourselves to the Bernoulli model (and often we illustrate our findings on the symmetric Bernoulli model in which every symbol is generated with equal probability). The analysis is based on three techniques: correlation sets, language approach, and generating functions. As a rule of thumb, our results say that the cut–off level k_d is approximately $\frac{m(1-P(\mathcal{Q}))}{q}$, where $P(\mathcal{Q})$ refers to the probability of the set $\mathcal{Q}$ of all q-grams of the pattern P (cf. Section 5). We are able to compute exact and asymptotic probability of having a given number of q-grams in a text-window of size m. This extends recent findings of Régnier and Szpankowski [13, 14], and is a contribution to the so called *scan statistics* [5, 12] (indeed, we measure a property of the text in a "sliding window").

2 The Formula for the Filtration Efficiency

The q-gram based approximate pattern matching algorithms have two phases. First, an algorithm filters the prominent text areas with potential occurrences. The filtration is based on a mathematical condition which is characteristic for every algorithm. In addition to true matches, the filtration phase gives also false matches, i.e., text areas where the filtration condition is fulfilled but no matches occur. Therefore, the second phase has to check the filtered text area by an accurate method.

Let us consider a pattern $P = P[1 \dots m]$ and a text $T = T[1 \dots n]$ in an alphabet Σ of size c. Let q be an integer and k an error level. The filtration condition of the Jokinen-Ukkonen q-gram method [9] is the following:

Lemma 1 (Jokinen-Ukkonen 1991) *Let an occurrence of P with at most k differences end at position j of text T. Then at least $m + 1 - (k+1)q$ q-grams of P occur in $T[j - m + 1 \dots j]$.*

A crucial factor, determining the applicability of an algorithm, is how efficiently it focuses the second phase on only those text areas which have true matches. The less false matches the first phase gives, the more efficient the filtration is.

Let now pattern P of length m, error level k, and gram length q be *fixed*, and text T of length n be randomly generated according a probabilistic model (we allow Markov model of any order which represents quite well English texts). Let a random variable O_n denote the number of text positions j such that an approximate match of P ends at j. In order to measure the applicability of algorithm A, we define its filtration efficiency f_A as follows:

Definition 1 *Let $U_n(A)$ denote the number of potential ending positions of an approximate match given by the filtration condition of algorithm A. The* filtration efficiency *of algorithm A is*

$$f_A = \frac{\boldsymbol{E}[O_n]}{\boldsymbol{E}[U_n(A)]}. \tag{1}$$

We will next analyze the filtration condition of the Jokinen-Ukkonen algorithm to explain its steep decrease in the filtration efficiency (i.e., phase transition) as a function of the error level k. This behaviour is characteristic also of other members of the q-family.

3 Discussion of the Main Results

Experimentally, we can estimate the filtration efficiency f_A by running algorithm A for N randomly generated texts $T_1, \dots, T_N$ of size n for a fixed pattern P,

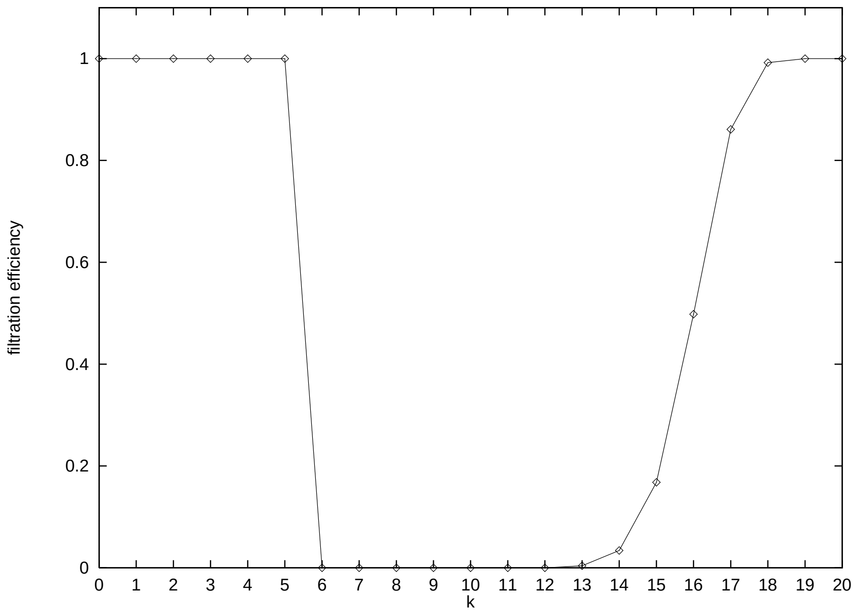

Figure 1: Collapse of the filtration

error level k, and gram length q. If O_{T_j} refers to the number of text positions j of T such that an approximate match of P ends at j and $U_T(A)$ refers to the number of potential ending positions marked in text T by algorithm A, it is natural to approximate f_A with the *experimental filtration efficiency* $\hat{f}_A(N)$, where

$$\hat{f}_A(N) = \frac{\Sigma_{i=1}^{N} O_{T_i}}{\Sigma_{i=1}^{N} U_{T_i}(A)}.$$

Experimental results show an interesting phenomenon in the behavior of q-gram based pattern matching algorithms. For relatively small error levels k, the filtration algorithm marks almost no positions of the text as potential, i.e., the filtration efficiency is pretty close to 100%. However, quite suddenly, at a certain error level $k_d = k_d(c, m)$, the method starts to lose its filtration efficiency very fast. Finally, after error level $k_0 = k_0(c, m)$, the method marks all the positions as potential.

Figure 1 gives the experimental filtration efficiency for a search of a word of length $m = 20$ ($n = 100000$, $q = 2$ and $c = 20$) as a function of the error level k, in a text generated according to the symmetric Bernoulli model ($\frac{0}{0}$ has been interpreted as 100% filtration, because of no false matches). Here, k_d can be anything between 6 and 9 but Figure 2 suggests $k_d = 8$ or 9, while $k_0 = 18$. For other alphabets, the curve is similar, but the values for k_d and k_0 vary. As

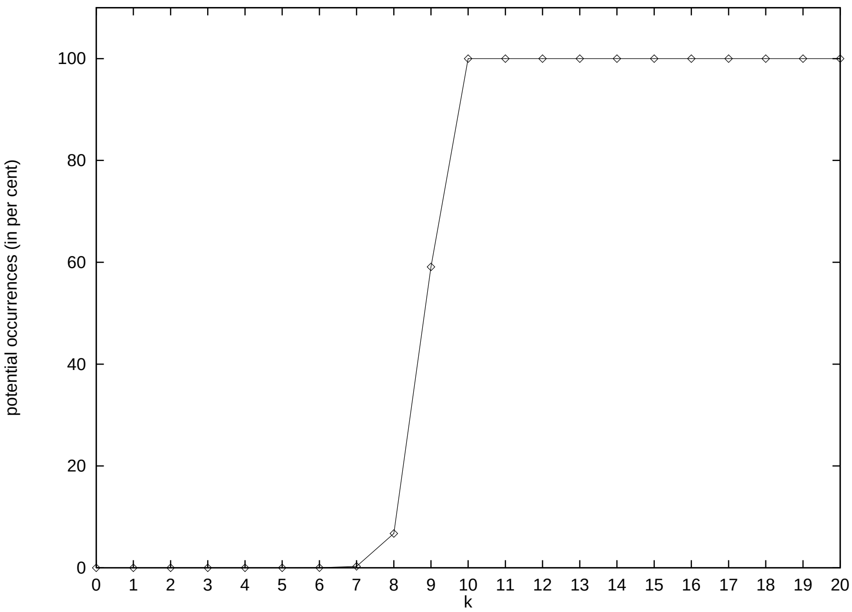

Figure 2: The percentage of the potential number of occurrences $\mathbf{E}[U_n]/n$ as a function of k.

shown in Figure 1, the filtration efficiency increases again when the error level approaches pattern length m. This part of the behavior is easy to explain: a higher allowed error level means more matches, and an error level of m accepts each string of length m as an approximate match. Therefore, the behavior is not typical of a specific filtration algorithm but depends on the definition of the distance. In summary, q–gram algorithms have a promising start, then sudden collapse, and finish with a happy end.

In Sections 4 and 5 we present our theoretical results that will predict quite well the filtration efficiency $f_A(k)$ around $k = k_d$. We point out that we can use a simple argument (see Section 5) to approximate the cut-off value k_d. Indeed, experimental results confirm that $k_d \approx k_d^{theor}$ where

$$k_d^{theor} = \frac{m(1 - P(\mathcal{Q}))}{q} - O(\sqrt{m}), \tag{2}$$

where $P(\mathcal{Q})$ is the probability of the set of all q-grams of the pattern P. For the above experiment $k_d^{theor} = 9$ (a more precisely formula derived in Section 5 gives $k_d^{theor} = 8$) which approximates well the true cut-off value $k_d = 8 - -9$.

We will now fix our approach to explain the behavior of the q-gram based filtration. To do this, we have to determine

- a representative A among the q-gram based algorithms such that its behavior follows the general scheme of the q-gram based algorithms, especially as far as filtration efficiency is concerned, and
- a quantity which explains the average-case behavior of algorithm A; for this, we use $U_n(A)$.

Representative algorithm. As a representative of q-gram based algorithms, we choose the one which is a straight-forward simplification of the original filtration lemma of Jokinen and Ukkonen. The representative algorithm gives the best filtration efficiency provided by the lemma, with one distinction: Jokinen and Ukkonen take into account the multiplicity of the q-grams of the searched pattern. Furthermore, their implementation applies a more time-efficient bookkeeping heuristics, at the cost of a slightly reduced filtration efficiency. However, an analysis of the representative algorithm determines the area of applicability of the original lemma.

The representative algorithm is based on counting the number of pattern q-grams for each text area $T[i-m+1\dots i], i = m,\dots,n$. The algorithm is as follows:

Algorithm ORIG
for $j = q$ **to** n **do** $COUNT[j] = 0$;
for $i = q$ **to** n
  if $T[i-q+1\dots i]$ occurs in P then
    for $j = i$ **to** $\min(n, i+m-q)$ **do**
      $COUNT[j] = COUNT[j] + 1$;

4 Analysis of Approximate Pattern Occurrences

In order to assess the efficiency f_{ORIG} we must analyze the number of approximate pattern occurrence O_n, and the number of potential matches U_n. In this section, following Régnier and Szpankowski [13, 14], we provide general approach to analyze the number of approximate pattern occurrence O_n (in particular, we estimate here the numerator $\mathbf{E}[O_n]$ appearing in the efficiency definition (1).) In the next section, we extend results from [13, 14] to assess U_n.

The method proposed in [13, 14] (se also [7]) is based on several techniques, such as

- employing an *autocorrelation set* of a pattern (or *correlation sets* for a collection of patterns) to determine how the pattern overlaps with itself;
- defining *languages* for strings which satisfy certain conditions; and

- deriving *generating functions* representing the languages, to give the probabilities of their occurrences.

Generating function defining a language. Let $\mathcal{S}$ denote a set of strings generated according to a given model (e.g., the symmetric Bernoulli model or a Markovian model) over the alphabet Σ. By a *language* $\mathcal{L}$, we refer to a *subset of strings* in $\mathcal{S}$. Consider a certain string ω in the language $\mathcal{L}$. Let $\mathsf{Pr}(\omega)$ denote the probability of ω among all the strings of length $|\omega|$ in $\mathcal{S}$, i.e., the probability that ω occurs at a given position in a text of the set $\mathcal{S}$. To get the probability of a string of a given length k in the language $\mathcal{L}$, we use a generating function $L(z)$, defined by

$$L(z) = \sum_{\omega \in \mathcal{L}} \mathsf{Pr}(\omega) z^{|w|},$$

where $\mathsf{Pr}(\varepsilon)$ conventionally equals 1. Let us consider an example.

Example 1 *Let Θ be a collection of texts which have been generated according to the symmetric Bernoulli model in a binary alphabet $\Sigma = \{\mathtt{A}, \mathtt{L}\}$, i.e.,*

$$\Theta = \{\omega \in \Sigma^\star \mid \mathsf{Pr}(\omega_i = \mathtt{A}) = \mathsf{Pr}(\omega_i = \mathtt{L}) = \frac{1}{2} \textit{ for each } i, 1 \le i \le |\omega|\},$$

where ω_i denotes the ith character in text ω. A language $\mathcal{L}_1 = \{\mathtt{A}, \mathtt{AA}, \mathtt{LA}, \mathtt{LL}\}$ denotes a subset of texts in Θ. Since in the symmetric Bernoulli model, $\mathsf{Pr}(\mathtt{A}) = \frac{1}{2}$ and $\mathsf{Pr}(\mathtt{AA}) = \mathsf{Pr}(\mathtt{LA}) = \mathsf{Pr}(\mathtt{LL}) = \frac{1}{4}$ hold, we get the corresponding generating function $L_1(z) = 1 + \frac{z}{2} + \frac{3z^2}{4}$.

Lemma 2 *Let us consider the concatenation of two languages in a Bernoulli model. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages, with their corresponding generating functions $L_1(z)$ and $L_2(z)$, respectively. The concatenated language $\mathcal{L} = \mathcal{L}_1 \cdot \mathcal{L}_2$ corresponds to the product $L(z)$ of the generating functions, i.e., $L(z) = L_1(z)L_2(z)$.*

Correlation set, correlation polynomial, and correlation matrix. An overlap of a pattern with itself, like that of the pattern LALLA in the text portion LALLALLAA, is possible to describe using the language formalism. Since the generating function, giving the probabilities for a match, is a transformation of the language representation, we need to find language formulae, describing texts with a certain amount of occurrences, including the ones with self-overlaps. In the following definitions, autocorrelation and correlation sets correspond to a language and autocorrelation and correlation polynomials to the generating function of the language.

Definition 2 *Let $P = P[1 \ldots m]$ be a pattern. The set $\mathcal{A}$,*

$$\mathcal{A} = \mathcal{A}_{P,P} = \{P[k+1 \ldots m] \mid P[m-k+1 \ldots m] = P[1 \ldots k]\},$$

is called the autocorrelation set *of pattern* P. *It defines a set* PP, *given by the following formula:*

$$PP = \{k \mid P[k+1 \ldots m] \in \mathcal{A}\}.$$

For the pattern $P = \texttt{LALLA}$, the autocorrelation set $\mathcal{A}$ is $\{\texttt{LLA}, \varepsilon\}$, corresponding to $PP = \{2, 5\}$. This is because the overlaps of the pattern P with itself, i.e. the prefixes that coincide with the suffixes of the pattern, are of length 2 and 5. Note that for any pattern ω, the empty word ε is an element of the autocorrelation set $\mathcal{A}_{\omega,\omega}$, and the length $|\omega|$ always belongs to the set $\omega\omega$.

Since an autocorrelation set is a language as a set of words, there is a corresponding generating function:

Definition 3 *Let* P *be a pattern. The generating function corresponding to the autocorrelation set* $\mathcal{A}_{P,P}$ *is called the* autocorrelation polynomial $A_P(z)$ *of pattern* P.

Example 2 *The autocorrelation polynomial for pattern* $P = \texttt{LALLA}$ *is* $A_P(z) = 1 + \frac{z^3}{8}$ *for the symmetric Bernoulli model. Intuitively, the autocorrelation polynomial gives the probabilities of those substrings which, appended to an occurrence of* P, *produce another occurrence of* P *within the interval of at most* $|P|-1$ *positions, i.e., the occurrences overlap with each other. In our example, an occurrence of* LALLA *is followed with probability* $\frac{1}{8}$ *by an overlapping occurrence, with an interval of 3:* ...LALLALLA....

However, when working with a set of patterns, like that of q-grams for a given pattern, we need a mechanism to deal with the overlaps between all the words in the set. To do this, we extend the definitions of the autocorrelation set and polynomial:

Definition 4 *Let* $P = P[1 \ldots m]$ *and* $R = R[1 \ldots m]$ *be patterns. The set* $\mathcal{A}_{P,R}$,

$$\mathcal{A}_{P,R} = \{R[k+1 \ldots m] \mid P[m-k+1 \ldots m] = R[1 \ldots k]\},$$

is called the correlation set *of patterns* P *and* R. *The correlation set defines a set* PR, *given by the following formula:*

$$PR = \{k \mid R[k+1 \ldots m] \in \mathcal{A}_{P,R}\}.$$

The generating function $A_{P,R}(z)$ *corresponding to the correlation set* $\mathcal{A}_{P,R}$ *is called the* correlation polynomial.

The overlappings between members of a set of patterns is given by the correlation matrix:

Definition 5 *Let $\mathcal{P} = \{P_1, \ldots, P_r\}$ be a set of patterns of length m. The matrix $A(z)$ of correlation polynomials,*

$$A(z) = \{A_{P_i,P_j}(z)\}_{i,j=1,\ldots,r},$$

is called the correlation matrix *of the set $\mathcal{P}$.*

Note that the patterns are of equal length. For our purposes, this is no restriction, since the members of a set of q-grams are all of length q.

Note also that the correlation polynomial $A_{P,R}(z) = 0$ if the pattern R does not overlap with the pattern P at P's right end, i.e., no prefix of R is a suffix of P. In addition, the constant term of the correlation polynomial is 1 if and only if P equals R.

Example 3 *Let us consider patterns $P = \texttt{LALLA}$ and $R = \texttt{ALLAL}$. The correlation sets and correlation polynomials are as follows:*

$$\begin{array}{llll}
A_{P,P} & = \{2,5\}, & A_{P,P}(z) & = 1 + \frac{z^3}{8}, \\
A_{P,R} & = \{1,4\}, & A_{P,R}(z) & = \frac{z}{2} + \frac{z^4}{16}, \\
A_{R,P} & = \{1,3\}, & A_{R,P}(z) & = \frac{z^2}{4} + \frac{z^4}{16}, \\
A_{R,R} & = \{2,5\}, & A_{R,R}(z) & = 1 + \frac{z^3}{8}.
\end{array}$$

Consequently, the correlation matrix $A(z)$ is

$$A(z) = \begin{pmatrix} 1 + \frac{z^3}{8} & \frac{z}{2} + \frac{z^4}{16} \\ \frac{z^2}{4} + \frac{z^4}{16} & 1 + \frac{z^3}{8} \end{pmatrix}.$$

The average filtration efficiency of any q-gram algorithm depends on the expected number of q-grams in a given text area, belonging to a given set. For example, the filtration efficiency of algorithm ORIG depends on the number of pattern q-grams in a text area of length m, and the number of approximate occurrences of the pattern in the text. Therefore, we will now consider how to determine this quantity.

In general, let $\mathcal{H} = \{P_i\}_{i=1,\ldots,\mu}$ be a set of patterns of length m and $O_n(\mathcal{H})$ a random variable which gives the number of occurrences of any member of the set $\mathcal{H}$ in a random text T of length n. Let us denote by $\mathcal{T}_i$, $i = 1, 2, \ldots,$ the language of words which have exactly i occurrence(s) from the set $\mathcal{H}$. The associated generating function $T^{(i)}(z)$ becomes

$$T^{(i)}(z) = \sum_{\omega \in \mathcal{T}_i} \mathsf{Pr}(\omega) z^{|\omega|} = \sum_{n \geq 0} \mathsf{Pr}(O_n(\mathcal{H}) = i) z^n. \tag{3}$$

Before defining the generating functions giving the distribution of the random variable $O_n(\mathcal{H})$, let us consider an example.

Example 4 *Consider a set of patterns* $\mathcal{H} = \{\mathtt{LALLA}, \mathtt{ALLAL}\}$*. Hence,* $\mathcal{T}_1 = \{\mathtt{LALLA}, \mathtt{ALLAL}, \mathtt{AALLAL}, \mathtt{ALALLA}, \mathtt{ALLALA}, \mathtt{ALLALL}, \mathtt{LALLAA}, \mathtt{LLALLA}, \ldots\}$ *and* $\mathcal{T}_2 = \{\mathtt{LALLAL}, \ldots\}$ *where all the elements of at most length six have been enumerated. It is straightforward to derive the first terms of the corresponding generating functions. Indeed,* $T^{(1)}(z) = \frac{2z^5}{32} + \frac{6z^6}{64} + \ldots$ *and* $T^{(2)}(z) = \frac{z^6}{64} + \ldots$*. The generating functions can be used as follows: the probability of exactly one occurrence from the set* $\mathcal{H}$ *in a random text of length six is given by the coefficient* $[z^6]$ *in* $T^{(1)}(z)$*, i.e.,* $[z^6]T^{(1)}(z) = \mathsf{Pr}(O_n(\mathcal{H}) = 1) = \frac{6}{64}$.

We are usually interested in the distribution and the moments of $O_n(\mathcal{H})$ for an arbitrary text of length n. It turns out that it is easier to first derive the generating functions, using the correlation sets, and then evaluate the needed coefficients of the generating functions. In addition, the approach gives a simple and intuitive formula for the expectation $\mathbf{E}O_n(\mathcal{H})$, the variance, limiting distribution and large deviations. Let us introduce the needed definitions:

Definition 6 *Let* $\mathcal{H} = \{P_i\}_{i=1,\ldots,\mu}$ *be a set of patterns of length* m*. Let* $O_n(\mathcal{H})$ *be a random variable, giving the number of occurrences from set* $\mathcal{H}$ *in a random text of length* n*. The distribution of* $O_n(\mathcal{H})$ *is given as a bivariate generating function* $T_{\mathcal{H}}(z, u)$*,*

$$T_{\mathcal{H}}(z,u) = \sum_{i=1}^{\infty} T_{\mathcal{H}}^{(i)}(z)u^i = \sum_{i=1}^{\infty}\sum_{n=0}^{\infty} \mathsf{Pr}(O_n(\mathcal{H}) = i)z^n u^i,$$

where the generating function $T_{\mathcal{H}}^{(i)}(z)$*, given in (3), corresponds to the language* $\mathcal{T}_i$ *of words including exactly* i *occurrences from the set* $\mathcal{H}$*.*

Note that the coefficient $[z^n u^i]$ of the generating function $T_{\mathcal{H}}(z, u)$ gives the probability of i occurrences from the set $\mathcal{H}$ in a random text of length n. The corresponding results for a single pattern P are obtained by choosing $\mathcal{H} = \{P\}$.

We are now ready to state the central result of Régnier and Szpankowski [14]. The results are based on applying the correlation matrix.

Theorem 1 *Let* $\mathcal{H}$ *be a given set of patterns of length* m*, and let* T *be a random text of length* n *generated according to the Bernoulli model (see [13] for an extension to the Markov model). The generating functions* $T_{\mathcal{H}}^{(i)}(z)$ *and* $T_{\mathcal{H}}(z, u)$ *can be computed as follows:*

$$T_{\mathcal{H}}^{(i)}(z) = R^t(z)M(z)^{i-1}U(z)$$

and

$$T_{\mathcal{H}}(z,u) = R^t(z)u(I - uM(z))^{-1}U(z),$$

where vectors $R(z), U(z)$ and matrix $M(z)$ are given by (in the above the upper index t denotes "transpose")

$$M(z) = (D(z) + (z-1)I)[D(z)]^{-1},$$

$$(I - M(z))^{-1} = A(z) + \frac{z^m}{1-z} 1 \cdot H^t,$$

$$U(z) = \frac{1}{1-z}(I - M(z)) \cdot 1,$$

$$R^t(z) = \frac{z^m}{1-z} H^t \cdot (I - M(z)),$$

and

$$D(z) = (1-z)A(z) + z^m 1 \cdot H^t.$$

Above, $H = (\mathsf{Pr}(P_1), \ldots, \mathsf{Pr}(P_{|\mathcal{H}|}))^t$ is a vector giving the probabilities of patterns P_i, $|\mathcal{H}|$ is the cardinality of $\mathcal{H}$, I is the identity matrix, $1 = (1, \ldots, 1)$ vector, and $A(z)$ is the correlation matrix for the set $\mathcal{H}$.

The derivation of the theorem above is based on representing the language $\mathcal{T}_i$ as a decomposition of other, more simple languages. The decomposition leads to a formula where one can make use of the language giving the correlation set. The crucial strength of the presented approach is that it gives accurate estimates for the number of occurrences of a *given* set of patterns in a random text of a given length. The problem is that deriving the generating functions yields time consuming calculations even with a computer. However, Régnier and Szpankowski are able to derive a simple formula for the expected number of occurrences: $\mathbf{E}[O_n(\mathcal{H})] = (n - m + 1)\mathsf{Pr}(\mathcal{H})$, where $\mathsf{Pr}(\mathcal{H}) = \sum_{\omega \in \mathcal{H}} \mathsf{Pr}(\omega)$ and the variance $\mathbf{Var}[O_n(\mathcal{H})] = (n - m + 1)c_1 + c_2$ where c_1 and c_2 are explicit constants that depend on the correlation matrix $A(z)$. In addition, they derive the *asymptotic* distribution of the number of occurrences by giving the probabilities for r occurrences for three cases: (1) r is a constant, or $r = O(1)$, (2) r is close to the expected number of occurrences, or $r = \mathbf{E}[O_n] + x\sqrt{\mathbf{Var}[O_n]}$, where $x = O(1)$, and (3) r is differs from its expectation by constant factor: $r = (1 + \delta)\mathbf{E}[O_n]$, where $\delta \neq 0$.

In the following, we will apply the presented method to estimate the number of potential matches in algorithm ORIG.

5 Analysis of q-Grams and Filtration Efficiency

In the previous section, we analyzed the number of approximate occurrences of a given pattern P that led to the evaluation of $\mathbf{E}[O_n(\mathcal{H})]$ appearing in the filtration coefficient f_{ORIG}. In this section, we derive a formula for the denominator of the filtration coefficient, namely $\mathbf{E}[U_n]$ which represents the number of potential matches. Actually, our ultimate goal is to derive the exact distribution of U_n, and even more. Observe that we can view the evaluation of U_n as a problem of *scan statistics* (cf. [5, 12]): More precisely, consider windows of size m that move along the text of length n. At any position of such a window, we assess whether a given property occurs or not. In our case, this property is simply the number of q-grams occurrences and we must determine whether there are more than the "magic number" $m + 1 - (k + 1)q$ appearances or not. The random variable U_n counts the number of times the given property occurs in $(n - m + 1)$ possible window positions. An analysis of U_n is not trivial due to strong statistical dependency between m consecutive positions of the sliding window.

To analyze the above posed problem we need to extend the approach of Régnier and Szpankowski [13, 14] to evaluate the number of *text windows* (not text positions) which include a certain *amount* (not just one) occurrences from a given set of q-grams. Hereafter we restrict ourselves to some preliminary discussion and the evaluation of the average number of potential matches, $\mathbf{E}[U_n]$, in order to theoretically justify the phase transition of the filtration algorithm $ORIG$. In the nutshell, our idea is to analyze the number of potential matches in a random text T by applying Theorem 1 *twice*. Indeed, we first define a *set of q-grams* $\mathcal{Q}_P^q$ and analyze the probability distribution of $O_m(\mathcal{Q}_P^q)$ through Theorem 1. In fact, we need the probability of the event $\{O_m(\mathcal{Q}_P^q) \geq m + 1 - (k + 1)q\}$. This can be also viewed as a *set of patterns* $\mathcal{H}_P$ such that each member of the set has at least $m+1-(k+1)q$ q-grams of the pattern P. Then, the number of potential matches U_n in the text T is equal to the number of occurrences from the set $\mathcal{H}_P$ in the text T, a quantity given again by Theorem 1.

Before going into the details, let us define some notations:

Definition 7 *Given the pattern $P = P[1 \dots m]$ and an integer q, the set $\mathcal{Q}_P^q$ gives the* set of pattern q-grams *of the pattern P, i.e.*

$$\mathcal{Q}_P^q = \{P[i \dots i + q - 1] \mid 1 \leq i \leq m - q + 1\}.$$

For any pattern R, the quantity $O_R(\mathcal{Q}_P^q)$ gives the number of q-grams of the pattern P in the pattern R.
If there is no confusion, we shall write below $\mathcal{Q}$ for $\mathcal{Q}_P^q$ and $O_m(\mathcal{Q})$ for $O_R(\mathcal{Q}_P^q)$ when $|R| = m$.

Note that generally $O_R(\mathcal{Q}_P^q) = O_P(\mathcal{Q}_R^q)$ does not hold, i.e., the number of R q-grams in the pattern P does not equal the number of P q-grams in the pattern R, even if the patterns P and R are of the same length. As a counterexample, choose P = `LALALA`, R = `LLLALL`, and $q = 2$. Then, $O_R(\mathcal{Q}_P^q)$ is 2, while $O_P(\mathcal{Q}_R^q)$ equals 5.

On the basis of the previous definitions, we can define the formula for the probability of a potential match at an arbitrary position i of a random text T:

Table 1: *The number of potential occurrences of Algorithm ORIG, in a binary text of length 100,000, generated according to the symmetric Bernoulli model. The* Observed *column gives the experimental value for the given pattern, while the* Predicted *column refers to the value given by the generating function $T_{\mathcal{Q}}(z,u)$.*

Pattern	*q*	*k*	*Observed*	*Predicted*
10110	3	0	9432	9375
11110	3	0	6341	6250
11111	3	0	3176	3125
11101	2	0	40671	40625
11101	2	1	90741	90625

Definition 8 *Let $I_i = 1$ when there is a potential match at the position i and zero otherwise. We denote the* probability of a potential match *at the position i by the quantity* $\mathsf{Pr}(I_i = 1)$*:*

$$\mathsf{Pr}(I_i = 1) = \mathsf{Pr}(O_{T[i-m+1\ldots i]}(\mathcal{Q}_P^q) \geq m + 1 - (k+1)q).$$

Observe that

$$\begin{aligned} \mathbf{E}[U_n] &= (n-m+1)\mathsf{Pr}(O_{T[1\ldots m]}(\mathcal{Q}_P^q) \geq m+1-(k+1)q) \\ &= (n-m+1)\sum_{i=m+1-(k+1)q}^{m-q+1} \mathsf{Pr}(O_m(\mathcal{Q}) = i) \end{aligned}$$

where we write $O_m(\mathcal{Q})$ for the longer $O_{T[1\ldots m]}(\mathcal{Q}_P^q)$. Thus, at least for the average $\mathbf{E}[U_n]$, we reduce the problem to the evaluation of the probability $\mathsf{Pr}(O_m(\mathcal{Q}) = i)$ where it can be computed from Theorem 1. Indeed, for the set $\mathcal{Q}$ define the correlation matrix $A_{\mathcal{Q}}(z)$, and then compute the bivariate generating function $T_{\mathcal{Q}}(z,u) = \sum_{i=1}^{\infty}\sum_{m=1}^{\infty} \mathsf{Pr}(O_m(\mathcal{Q}) = i) z^m u^i$ as given explicitly by Theorem 1 with vector H replaced by $Q = (\mathsf{Pr}(Q_1), \ldots, \mathsf{Pr}(Q_{|\mathcal{Q}|})$.

To evaluate the theoretically predicted number of potential occurrences, given by the generating function $T_{\mathcal{Q}}(z,u)$, we will consider a simple example in the binary alphabet $\Sigma = \{0,1\}$. Using Algorithm ORIG, we searched for various patterns of length 5, with the error level $k = 0$, in a text of length 100,000. The results, presented in Table 1, indicate that the theoretically predicted values coincide well with the observed ones.

The results illustrate also that the number of potential occurrences depends on the form of the pattern, not only its length.

In summary, we can formulate our final finding.

Theorem 2 *The filtration coefficient f_{ORIG} can be evaluated as*

$$\begin{aligned} f_{ORIG} &= \frac{(n-m+1)\mathsf{Pr}(\mathcal{H})}{(n-m+1)\mathsf{Pr}(O_m(\mathcal{Q}) \geq m+1-(k+1)q)} \\ &= \frac{\sum_{P_i \in \mathcal{H}} \mathsf{Pr}(P_i)}{\sum_{i=m+1-(k+1)q}^{m-q+1} [z^m u^i] T_{\mathcal{Q}}(z,u)} \end{aligned}$$

where P_i is a substring within distance k from the given pattern P, and $T_Q(z,u)$ is evaluated according to Theorem 1 with the correlation matrix $A_{\mathcal{Q}}(z)$ of the set $\mathcal{Q}$ of all q-grams.

Estimating the cut-off value of the collapse. Finally, we provide here an *heuristic* argument justifying our approximate formula (2) for the cut-off value k_d. We need to assess $\mathsf{Pr}(O_m(\mathcal{Q}) \geq m+1-(k+1)q)$ as a function of k. Observe that this probability increases (hence, f_{ORIG} decreases) with the increase of k. We know that $O_m(\mathcal{Q})$ can be well approximated by a normal distribution with mean $\mathbf{E}[O_m(\mathcal{Q})] = (m-q+1)\mathsf{Pr}(\mathcal{Q})$ and variance $\mathbf{Var}[O_m(\mathcal{Q})] = c_1(m-q+1)+c_2$ where c_1 and c_2 are explicitly computable constants (that depend on the correlation matrix $A_{\mathcal{Q}}(z)$). Therefore, one should expect a sudden increase in the probability $\mathsf{Pr}(O_m(\mathcal{Q}) \geq m+1-(k+1)q)$ and a sudden decrease in the filtration efficiency f_{ORIG} around the mean plus the standard deviation. In other words we expect that k_d should be such that

$$(m-q+1)P(\mathcal{Q}) + xc_1\sqrt{m} \approx m+1-(k+1)q$$

where $x \in (1,3)$ is a constant. Thus,

$$\begin{aligned} k &\approx \frac{m(1-P(\mathcal{Q}))}{q} - \frac{q(1+P(\mathcal{Q})) + xc_2\sqrt{m} - 1 - P(\mathcal{Q})}{q} \\ &\approx \frac{m(1-P(\mathcal{Q}))}{q} - O(\sqrt{m}), \end{aligned}$$

as proposed in (2) and confirmed by our experiments.

6 Concluding Remarks

We conclude this paper with a brief sketch of our general approach that will lead to the distribution of the number of potential matches U_n. As we mentioned before, to analyze U_n we will apply Theorem 1 twice. The basic idea of our derivation is that we will define a *set of patterns* $\mathcal{H}_P$ such that each member of the set has at least $m+1-(k+1)q$ q-grams of the pattern P. The size of the set $\mathcal{H}_P$ is given by Theorem 1. Now, the number of potential matches in the text T is equal to the number of occurrences from the set $\mathcal{H}_P$ in the text T, a quantity given again by Theorem 1.

Let us now present our approach in a formal way. Let $\tilde{\mathcal{H}}_i$, $i = 0, \ldots, m-q+1$ denote the set of strings of length m, with exactly i q-grams of the pattern P:

$$\tilde{\mathcal{H}}_i = \{\omega \in \Sigma^m \mid O_\omega(\mathcal{Q}_P^q) = i\}.$$

Then

$$\tilde{\mathcal{H}} = \biguplus_{i=m+1-(k+1)q}^{m-q+1} \tilde{\mathcal{H}}_i, \tag{4}$$

where the operation $A \uplus B$ is defined for all pairs of disjoint sets such that $A \uplus B = A \cup B$. The characteristics of set $\tilde{H}$ are obtained through Theorem 1.

To evaluate U_n, we consider the number of occurrences of $\tilde{\mathcal{H}}$ in text T, thus one must study $U_n(\tilde{\mathcal{H}})$. Let its bivariate generating function be $T_{\tilde{\mathcal{H}}}(z, u)$. It can be evaluate again by Theorem 1. Then

$$\mathsf{Pr}(U_n = k) = [z^n u^k] T_{\tilde{\mathcal{H}}}(z, u).$$

Erkki Sutinen teaches in the Department of Computer Science, FIN-00014 University of Helsinki, Finland. E-mail: sutinen@cs.helsinki.fi

Wojciech Szpankowski teaches in the Department of Computer Science at Purdue University, W. Lafayette, IN 47907, U.S.A. E-mail: spa@cs.purdue.edu

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman: Basic local alignment search tool. *Journal of Molecular Biology*, 215, 403–410, 1990.

[2] M. Atallah, Y. Genin, and W. Szpankowski: A pattern matching approach to image compression. In: *Proc. International Conference on Image Processing*, vol. II. 349–352, Lausanne, 1996.

[3] W. Chang and T. Marr: Approximate string matching and local similarity. In: *Combinatorial Pattern Matching, Proceedings of 5th Annual Symposium* (ed. M. Crochemore and D. Gusfield), *Lecture Notes in Computer Science* **807**, Springer-Verlag, Berlin, 1994, 259–273.

[4] M. Crochemore and W. Rytter, *Text Algorithms*, Oxford University Press, New York, 1995.

[5] A. Dembo and S. Karlin: Poisson Approximations for r-Scan Processes. *The Annals of Applied Probability*, 2, 329-357, 1992.

[6] Z. Galil, and R. Giancarlo: Data structures and algorithms for approximate string matching, *Journal of Complexity*, 4, 33-72 (1988).

[7] L. Guibas and A. W. Odlyzko: String Overlaps, Pattern Matching, and Non-transitive Games. *J. Combin.Theory Ser. A*, 30, 183-208, 1981.

[8] P. Jacquet and W. Szpankowski: Asymptotic behavior of the Lempel-Ziv parsing scheme and digital search trees, *Theoretical Computer Science*, 144, 161–197, 1995.

[9] P. Jokinen and E. Ukkonen: Two algorithms for approximate string matching in static texts. In: *Proc. Mathematical Foundations of Computer Science 1991* (ed. A. Tarlecki), *Lecture Notes in Computer Science 520*, Springer-Verlag, Berlin, 1991, 240–248.

[10] G. Landau and U. Vishkin: Fast string matching with k differences. *Journal of Computer and System Sciences* **37** (1988), 63–78.

[11] T. Łuczak and W. Szpankowski: A suboptimal lossy data compression based on approximate pattern matching. *IEEE Trans. Information Theory*, 43, 1439–1451, 1997.

[12] J. Naus: Approximates of distributions of scan statistics. *J. Amer. Statist. Assoc.*, 77, 177-183, 1982.

[13] M. Régnier and W. Szpankowski: On the approximate pattern occurrences in a text. In: *Proc. Compression and Complexity of SEQUENCE'97*, Positano 1997.

[14] M. Régnier and W. Szpankowski: On pattern frequency occurrences in a Markovian sequence. *Algorithmica*, to appear; also Purdue University CSD-TR-96-043, 1996 available at `http://www.cs.purdue.edu/people/spa`.

[15] E. Sutinen and J. Tarhio: Filtration with q-samples in approximate string matching. In: *Proc. 7th Symposium on Combinatorial Pattern Matching CPM '96* (ed. D. Hirschberg, G. Myers), *Lecture Notes in Computer Science* **1075**, Springer, Berlin, 1996, 50–63.

[16] E. Sutinen and J. Tarhio: On using q-gram locations in approximate string matching. In: *Proc. 3rd Annual European Symposium on Algorithms ESA '95* (ed. P. Spirakis), *Lecture Notes in Computer Science* **979**, Springer, Berlin, 1995, 327–340.

[17] T. Takaoka: Approximate pattern matching with samples. *Proceedings of ISAAC '94, Lecture Notes in Computer Science* **834**, Springer-Verlag, Berlin, 1994, 234–242.

[18] E. Ukkonen: Finding approximate patterns in strings. *Journal of Algorithms* **6** (1985), 132–137.