Approximating Approximate Pattern Matching

Jan Studený
Department of Computer Science, ETH Zürich, Switzerland
studenyj@student.ethz.ch

Przemysław Uznański
Institute of Computer Science, University of Wrocław, Poland[1]
puznanski@cs.uni.wroc.pl

Abstract

Given a text T of length n and a pattern P of length m, the approximate pattern matching problem asks for computation of a particular *distance* function between P and every m-substring of T. We consider a $(1 \pm \varepsilon)$ multiplicative approximation variant of this problem, for ℓ_p distance function. In this paper, we describe two $(1+\varepsilon)$-approximate algorithms with a runtime of $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon})$ for all (constant) non-negative values of p. For constant $p \geq 1$ we show a deterministic $(1+\varepsilon)$-approximation algorithm. Previously, such run time was known only for the case of ℓ_1 distance, by Gawrychowski and Uznański [ICALP 2018] and only with a randomized algorithm. For constant $0 \leq p \leq 1$ we show a randomized algorithm for the ℓ_p, thereby providing a smooth tradeoff between algorithms of Kopelowitz and Porat [FOCS 2015, SOSA 2018] for Hamming distance (case of $p = 0$) and of Gawrychowski and Uznański for ℓ_1 distance.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Approximate Pattern Matching, ℓ_p Distance, ℓ_1 Distance, Hamming Distance, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.CPM.2019.15

Related Version A full version of the paper is available at [24], http://arxiv.org/abs/1810.01676.

1 Introduction

Pattern matching is one of the core problems in text processing algorithms. Given a text T of length n and a pattern P of length m, $m \leq n$, both over an alphabet Σ, one searches for occurrences of P in T as a substring. A generalization of a pattern matching is to find substrings of T that are similar to P, where we consider a particular string distance and ask for all m-substrings of T where the distance to P does not exceed a given threshold, or simply report the distance from P to every m-substring of T. Typical distance functions considered are Hamming distance, ℓ_1 distance, or in general ℓ_p distances for some constant p, assuming input is over a numerical, e.g. integer, alphabet.

For reporting all Hamming distances, Abrahamson [1] described an algorithm with the complexity of $\mathcal{O}(n\sqrt{m \log m})$. Using a similar approach, the same complexity was obtained in [18] and later in conference works [3, 5] for reporting all ℓ_1 distances. It is a major open problem whether near-linear time algorithm, or even $\mathcal{O}(n^{3/2-\varepsilon})$ time algorithm, is possible for such problems. A conditional lower bound [6] was shown, via a reduction from matrix multiplication. This means that existence of combinatorial algorithm with runtime $\mathcal{O}(n^{3/2-\varepsilon})$ solving the problem for Hamming distances implies combinatorial algorithms for boolean matrix multiplication with $\mathcal{O}(n^{3-\delta})$ runtime, which existence is unlikely. If one is uncomfortable with poorly defined notion of *combinatorial* algorithms, one can apply the reduction to obtain a lowerbound of $\Omega(n^{\omega/2})$ for Hamming distances pattern matching,

[1] Most of the work was done while the author was affiliated with ETH Zürich.

30th Annual Symposium on Combinatorial Pattern Matching (CPM 2019).
Editors: Nadia Pisanti and Solon P. Pissis; Article No. 15; pp. 15:1–15:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

where $2 \leq \omega < 2.373$ is a matrix multiplication exponent.[2] Later, the complexity of pattern matching under Hamming distance and under ℓ_1 distance was proven to be identical (up to polylogarithmic terms) [11, 19].

The mentioned hardness results serve as a motivation for considering relaxation of the problems, with $(1+\varepsilon)$ multiplicative approximation being the obvious candidate. For Hamming distance, Karloff [14] was the first to propose an efficient approximation algorithm with a run time of $\mathcal{O}(\frac{n}{\varepsilon^2}\log^3 m)$. The $\frac{1}{\varepsilon^2}$ dependency was believed to be inherent, as is the case for e.g. space complexity of sketching of Hamming distance, cf. [25, 12, 4]. However, for approximate pattern matching that was refuted by Kopelowitz and Porat [15, 16], by providing randomized algorithms with complexity $\mathcal{O}(\frac{n}{\varepsilon}\log n \log m \log \frac{1}{\varepsilon}\log|\Sigma|)$ and $\mathcal{O}(\frac{n}{\varepsilon}\log n \log m)$ respectively. Moving to ℓ_1 distance, Lipsky and Porat [20] gave a deterministic algorithm with a run time of $\mathcal{O}(\frac{n}{\varepsilon^2}\log m \log U)$, while later Gawrychowski and Uznański [10] have improved the complexity to a (randomized) $\mathcal{O}(\frac{n}{\varepsilon}\log^2 n \log m \log U)$, where U is the maximal integer value on the input. Additionally, we refer the reader to the line of work on other relaxations on exact the distance reporting [2, 7, 10, 3].

A folklore result (c.f. [20]) states that the randomized algorithm with a run time of $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon^2})$ is in fact possible for any ℓ_p distance, $0 < p \leq 2$, with use of p-stable distributions and convolution.[3] Such distributions exist only when $p \leq 2$, which puts a limit on this approach. See [22] for wider discussion on p-stable distributions. Porat and Efremenko [23] have shown how to approximate general distance functions between pattern and text in time $\mathcal{O}(\frac{n}{\varepsilon^2}\log^2 m \log^3|\Sigma| \log B_d)$, where B_d is upperbound on distance between two characters in Σ. Their solution does not immediately translates to ℓ_p distances, since it allows only for score functions of form $\sum_j d(t_{i+j}, p_j)$ where d is arbitrary metric over Σ. Authors state that their techniques generalize to computation of ℓ_2 distances, but the dependency ε^{-2} in their approach is unavoidable. [20] observe that ℓ_2 pattern matching can be in fact computed in $\mathcal{O}(n \log m)$ time, by reducing it to a single convolution computation. This case and analogously case of $p = 4, 6, \ldots$ are the only ones where fast and exact algorithm is known.

We want to point that for ℓ_∞ pattern matching there is an approximation algorithm of complexity $\mathcal{O}(\frac{n}{\varepsilon}\log m \log U)$ by Lipsky and Porat [20]. Moving past pattern matching, we want to point that in a closely related problem of computing $(\min,+)$-convolution there exists $\mathcal{O}(\frac{n}{\varepsilon}\log\frac{n}{\varepsilon}\log U)$ time algorithm computing $(1+\varepsilon)$ approximation, cf. Mucha et al. [21].

Two questions follow naturally. First, is there a $\widetilde{\mathcal{O}}(\frac{n}{\mathrm{poly}(\varepsilon)})$ algorithm for ℓ_p norms pattern matching when $p > 2$? Second, is there anything special to $p = 0$ and $p = 1$ cases that allows for faster algorithms, or can we extend their complexities to other ℓ_p norms? To motivate further those questions, observe that in the regime of maintaining ℓ_p sketches in the *turnstile* streaming model (sequence of updates to vector coordinates), one needs small space of $\Theta(\log n)$ bits when $p \leq 2$ (cf. [13]), while when $p > 2$ one needs large space of $\Theta(n^{1-2/p}\log n)$ bits (cf. [9, 17]) meaning there is a sharp transition in problem complexity at $p = 2$. Similar phenomenon of transition at $p = 2$ is observed for p-stable distributions, and one could expect such transition to happen in the pattern matching regime as well.

In this work we show that for any *constant* $p \geq 0$ there is an algorithm of complexity $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon})$, replicating the phenomenon of linear dependency on ε^{-1} from Hamming distance and ℓ_1 distance to all ℓ_p norms. Additionally this provides evidence that no transition at $p = 2$ happens, and so far to our understanding cases of $p > 2$ and $p < 2$ are of similar hardness.

[2] Although the issue is that we do not even know whether $\omega > 2$ or not.

[3] We use $\widetilde{\mathcal{O}}$ notation to hide factors polylogarithmic in $n, m, |\Sigma|, U$ and ε^{-1}.

1.1 Definitions and preliminaries

Model. In most general setting, our inputs are strings taken from arbitrary alphabet Σ. We use this notation only when structure of alphabet is irrelevant for the problem (e.g. Hamming distances). However, when considering ℓ_p distances we focus our attention over an integer alphabet $[U] \stackrel{\text{def}}{=} \{0, 1, ..., U-1\}$ for some U. One can usually assume that $U = \text{poly}(n)$, and then $\log U$ term can be safely hidden in the $\widetilde{\mathcal{O}}$ notation, however we provide the dependency explicitly in Theorem statements. Even without such assumption, we can assume standard word RAM model, in which arithmetic operations on words of size $\log U$ take constant time. Otherwise the complexities have an additional $\log U$ factor. We also denote $u = \log U$. While we restrict input integer values, we allow intermediate computation and output to consist of floating point numbers having u bits of precision.

Distance between strings. Let $X = x_1 x_2 \dots x_n$ and $Y = y_1 y_2 \dots y_n$ be two strings. For any $p > 0$, we define their ℓ_p distance as

$$\ell_p(X, Y) = \left(\sum_i |x_i - y_i|^p\right)^{1/p}.$$

Particularly, ℓ_1 distance is known as *Manhattan distance*, and ℓ_2 distance is known as *Euclidean distance*. Observe that the p-th power of ℓ_p distance has particularly simpler form of $\ell_p(X,Y)^p = \sum_i |x_i - y_i|^p$.

The *Hamming distance* between two strings is defined as

$$\text{Ham}(X, Y) = |\{i : x_i \neq y_i\}|.$$

Adopting the convention that $0^0 = 0$ and $x^0 = 1$ for $x \neq 0$, we observe that $(\ell_p)^p$ approaches Hamming distance as $p \to 0$. Thus Hamming distance is usually denoted as ℓ_0 (although $(\ell_0)^0$ is more precise notation).

Text-to-pattern distance. For text $T = t_1 t_2 \dots t_n$ and pattern $P = p_1 p_2 \dots p_m$, the text-to-pattern distance is defined as an array S such that, for every i, $S[i] = d(T[i+1 \mathinner{..} i+m], P)$ for particular distance function d. Thus, for ℓ_p distance $S[i] = \left(\sum_{j=1}^m |t_{i+j} - p_j|^p\right)^{1/p}$, while for Hamming distance $S[i] = |\{j \in \{1, \dots, m\} : t_{i+j} \neq p_j\}|$. Then $(1+\varepsilon)$-approximate distance is defined as an array S_ε such that, for every i, $(1-\varepsilon) \cdot S[i] \leq S_\varepsilon[i] \leq (1+\varepsilon) \cdot S[i]$.

Rounding and arithmetic operations. For any value x, we denote by $x^{(i)} = \lfloor x/2^i \rfloor \cdot 2^i$ the value with i y bits rounded. However, with a little stretch of notation, we do not limit value of i to be positive. We denote by $\|r\|_c$ the norm *modulo* c, that is $\|r\|_c = \min(r \bmod c, c - (r \bmod c))$.

1.2 Our results

In this paper we answer favorably both questions by providing relevant algorithms. First, we show how to extend the deterministic ℓ_1 distances algorithm into ℓ_p distances, when $p \geq 1$.

▶ **Theorem 1.** *For any $p \geq 1$ there is a* deterministic *algorithm computing $(1+\varepsilon)$ approximation to pattern matching under ℓ_p distances in time $\mathcal{O}(\frac{n}{\varepsilon} \log m \log U)$ (assuming $\varepsilon \leq 1/p$).*

We then move to the case of ℓ_p distances when $p < 1$. We show that it is possible to construct a randomized algorithm with the desired complexity.

▶ **Theorem 2.** *For* $0 < p < 1$*, there is a* randomized *algorithm computing* $(1+\varepsilon)$ *approximation to pattern matching under* ℓ_p *distances in time* $\mathcal{O}(p^{-1}\varepsilon^{-1}n\log m\log^2 U\log n)$*. The algorithm is correct with high probability.*[4]

Finally, combining with existing ℓ_0 algorithm from [16] we obtain as a corollary that for *constant* $p \geq 0$ approximation of pattern matching under ℓ_p distances can be computed in $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon})$ time.

2 Approximation of ℓ_p distances

We start by showing how convolution finds its use in counting versions of pattern matching, either exact or approximation algorithms. Consider the case of pattern matching under ℓ_2 distances. Observe that we are looking for S such that $S[i]^2 = \sum_{j-k=i}(t_j - p_k)^2 = \sum_j t_j^2 + \sum_k p_k^2 - 2\sum_{j-k=i} t_j p_k$. The last term is just a convolution of vectors in disguise and is equivalent to computing convolution of T and reverse ordered P. Such approach can be applied to solving exact pattern matching via convolution (observing that ℓ_2 distance is 0 iff there is an exact match).

We follow with a technique for computing exact text-to-pattern distance, for arbitrary distance functions, introduced by [20], which is a generalization of a technique used in [8]. We provide a short proof for completeness.

▶ **Theorem 3** ([20])**.** *Text-to-pattern distance where strings are over* arbitrary *alphabet* Σ *can be computed* exactly *in time* $\mathcal{O}(|\Sigma| \cdot n\log m)$*.*

Proof. For every letter $c \in \Sigma$, construct a new text T^c by setting $T^c[i] = 1$ if $t_i = c$ and $T^c[i] = 0$ otherwise. A new pattern P^c is constructed by setting $P^c[i] = d(c, p_i)$. Since $d(t_{i+j}, p_j) = \sum_{c\in\Sigma} T^c[i+j] \cdot P^c[j]$, it is enough to invoke $|\Sigma|$ times convolution. ◀

Theorem 3 allows us to compute text-to-pattern distance exactly, but the time complexity $\mathcal{O}(|\Sigma|n\log m)$ is prohibitive for large alphabets (when $|\Sigma| = \mathit{poly}(n)$). However, it is enough to reduce the size of alphabet used in the problem (at the cost of reduced precision) to reach desired time complexity. While this might be hard, we proceed as follows: we decompose our weight function into a sum of components, each of which is approximated by a corresponding function on a reduced alphabet.

We say that a function d is *effectively over smaller alphabet* Σ' if it is represented as $d(x,y) = d'(\iota_1(x), \iota_2(y))$ for some $\iota_1, \iota_2 : \Sigma \to \Sigma'$ and d'. It follows from Theorem 3 that text-to-pattern under distance d can be computed in time $\widetilde{\mathcal{O}}(|\Sigma'|n)$ (ignoring the cost of computing ι_1 and ι_2).

Decomposition. Let $D(x,y) = |x-y|^p$ be a function corresponding to $(\ell_p)^p$ distance, that is $\ell_p(X,Y)^p = \sum_i D(x_i, y_i)$. Our goal is to decompose $D(x,y) = \sum_i \alpha_i(x,y)$ into small (polylogarithmic) number of functions, such that each $\alpha_i(x,y)$ is approximated by $\beta_i(x,y)$ that is effectively over alphabet of $\mathcal{O}(\frac{1}{\varepsilon})$ size (up to polylogarithmic factors). Now we can use Theorem 3 to compute contribution of each β_i. We then have that $G(x,y) = \sum_i \beta_i(x,y)$ approximates F, and text-to-pattern distance under G can be computed in the desired $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon})$ time. We present such decomposition, useful immediately in case of $p \geq 1$ and as we see in section 2.2 with a little bit of effort as well in case when $0 < p \leq 1$.

[4] Probability at least $1 - 1/n^c$ for arbitrarily large constant c.

Useful estimations. We use following estimations in our proofs. For $p \geq 1$

$$(1-\varepsilon)^p \geq 1 - p\varepsilon, \qquad \text{for } 0 \leq \varepsilon \leq 1, \tag{1}$$

$$(1+\varepsilon)^p \geq 1 + p\varepsilon, \qquad \text{for } 0 \leq \varepsilon, \tag{2}$$

$$(1-\varepsilon)^p \leq 1 - p\varepsilon(1-1/e), \qquad \text{for } 0 \leq \varepsilon \leq 1/p, \tag{3}$$

$$a^p - (a-b)^p \leq pa^{p-1}b, \qquad \text{for } a \geq b \geq 0. \qquad \text{(follows from 2)} \tag{4}$$

For $0 \leq p \leq 1$

$$(1-\varepsilon)^p \leq 1 - p\varepsilon, \qquad \text{for } 0 \leq \varepsilon \leq 1, \tag{5}$$

$$(1-\varepsilon)^p \geq 1 - 2p\varepsilon \ln 2, \qquad \text{for } 0 \leq \varepsilon \leq 1/2, \tag{6}$$

$$(1+\varepsilon)^p \geq 1 + p\varepsilon \ln 2, \qquad \text{for } 0 \leq \varepsilon \leq 1, \tag{7}$$

$$a^p - (a-b)^p \leq 2pa^{p-1}b \ln 2, \qquad \text{for } a \geq 2b \geq 0. \qquad \text{(follows from 6)} \tag{8}$$

2.1 Algorithm for $p \geq 1$

In this section we prove Theorem 1. We start by constructing a family of functions F_i, which are better refinements of F as i decreases.

First step. Let us denote

$$F_i(x,y) = \Big(\max(0, |x-y| - 2^i)\Big)^p \qquad \text{and} \qquad f_i = F_i - F_{i+1}.$$

Observe that $F_u = 0$ (for $0 \leq x, y \leq U$). Moreover, there is a telescoping sum $F_i = \sum\limits_{j=i}^{u} f_j$. To better see the the telescopic sum, consider case $p = 1$. We then represent $F_{-u}(x,y) = \sum_{i=-u}^{u} f_i(x,y) = (-2^{-u} + 2^{-u+1}) + (-2^{-u+1} + 2^{-u+2}) + \ldots + (-2^{t-1} + 2^t) + (|x-y| - 2^t) + 0 + \ldots + 0$. Such decomposition (for $p = 1$) was first considered, to our knowledge, in [20].

Second step. Instead of using x and y for evaluation of F_i, we evaluate F_i using x and y with all bits younger than i-th one set to zero. Formally, define $x^{(i)} = \lfloor x/2^i \rfloor \cdot 2^i$, $y^{(i)} = \lfloor y/2^i \rfloor \cdot 2^i$. Now we denote

$$G_i(x,y) = F_i(x^{(i)}, y^{(i)})$$

Similarly as for f_i, define $g_i = G_i - G_{i+1}$. Using the same reasoning, we have $G_u = 0$. For integers $i \leq 0$ the functions F_i and G_i are the same (as we are not rounding) and therefore $F_{-u} = G_{-u} = \sum\limits_{i=-u}^{u} g_i$. Intuitively, g_i captures contribution of i-th bit of input to the output value (assuming all older bits are set and known, and all younger bits are unknown).

Third step. Let η be a value to be fixed later, depending on ε and p. Assume w.l.o.g. that η is such that $1/\eta$ is an integer. We now define $\widehat{g}_i$ as a refinement of g_i, by replacing $|x^{(i)} - y^{(i)}|$ with $\|x^{(i)} - y^{(i)}\|_{B_i}$ and $|x^{(i+1)} - y^{(i+1)}|$ with $\|x^{(i+1)} - y^{(i+1)}\|_{B_i}$, where $B_i = 2^i/\eta$, that is doing all the computation *modulo* B_i. To be precise, define

$$\overrightarrow{G}_i(x,y) = \Big(\max(0, \|x^{(i)} - y^{(i)}\|_{B_i} - 2^i)\Big)^p$$

$$\overleftarrow{G}_{i+1}(x,y) = \Big(\max(0, \|x^{(i+1)} - y^{(i+1)}\|_{B_i} - (2^{i+1})\Big)^p$$

and then $\widehat{g}_i = \overrightarrow{G}_i - \overleftarrow{G}_{i+1}$. Additionally, we denote for short $\widehat{G}_i = \sum\limits_{j=i}^{u} \widehat{g}_j$.

Intuitively, $\widehat{g}_i$ approximates g_i in the scenario of limited knowledge – it estimates contribution of i-th bit of input to the output, assuming knowledge of bits $i+1$ to $i+\log\eta^{-1}$ of input. We are now ready to provide an approximation algorithm to $(\ell_p)^p$ text-to-pattern distances.

▶ **Algorithm 4.**

Input:

- *T is the text,*
- *P is the pattern,*
- *η controls the precision of the approximation.*

Steps:

1. *For each $i \in \{-u,\dots,u\}$ compute array S_i being the text-to-pattern distance between T and P using $\widehat{g}_i$ distance function (parametrized by η) using Theorem 3.*
2. *Output array $S_\varepsilon[i] = \left(\sum\limits_{j=-u}^{u} S_j[i]\right)^{1/p}$.*

To get the $(1+\varepsilon)$ approximation we run the Algorithm 4 with $\eta = \frac{\varepsilon}{128}$.

Now, we need to show the running time and correctness of the result. Firstly, to prove the correctness, we divide summands $\widehat{g}_i$ into three groups and reason about them separately. As computing F_{-u}, G_{-u}(by summing f_i's and g_i's respectively) yields $(1+\varepsilon)$ multiplicative error, we will show that the difference between computing g_i and $\widehat{g}_i$ brings only an additional $(1+\varepsilon)$ multiplicative error.

▶ **Lemma 5.** *For i such that $|x-y| \le 2^i$ both $g_i(x,y)=0$ and $\widehat{g}_i(x,y)=0$.*

Proof. As both $g_i, \widehat{g}_i$ are symmetric functions, we can w.l.o.g. assume $x \ge y$. $\forall j \ge i$:

$$\left|x^{(j)} - y^{(j)}\right| = 2^j\left(\left\lfloor\frac{x}{2^j}\right\rfloor - \left\lfloor\frac{y}{2^j}\right\rfloor\right) \le 2^j\left(\left\lfloor\frac{x}{2^j}\right\rfloor - \left\lfloor\frac{x-2^i}{2^j}\right\rfloor\right) \le 2^j.$$

Therefore $G_j = 0$ from which $g_i(x,y)=0$ follows. And because $\|x^{(j)} - y^{(j)}\|_{B_j} \le |x^{(j)} - y^{(j)}|$ we have $\widehat{g}_i(x,y)=0$ as well. ◀

▶ **Lemma 6.** *For i such that $|x-y| > 2^i \ge 4\eta|x-y|$ we have $g_i(x,y) = \widehat{g}_i(x,y)$.*

Proof. For $g_i(x,y) = \widehat{g}_i(x,y)$ to hold, it is enough to show that both *norms* $|\cdot|$ and $\|\cdot\|_{B_i}$ are the same for $x^{(i)} - y^{(i)}$ and $x^{(i+1)} - y^{(i+1)}$. This happens if the absolute values of the respective inputs are smaller than $B_i/2$. Let us bound both $|x^{(i)} - y^{(i)}|$ and $|x^{(i+1)} - y^{(i+1)}|$:

$$\max(|x^{(i)} - y^{(i)}|, |x^{(i+1)} - y^{(i+1)}|) \le |x-y| + 2^{i+1} \le 2^{i+1}(1 + \frac{1}{8\eta}).$$

We can w.l.o.g. assume $\eta \le 1/8$ in order to make $\frac{1}{8\eta}$ a dominant term in the parentheses and reach:

$$\max(|x^{(i)} - y^{(i)}|, |x^{(i+1)} - y^{(i+1)}|) \le 2^{i+1}(1 + \frac{1}{8\eta}) \le \frac{2^i}{2\eta} = \frac{B_i}{2}.$$

Therefore $\|x^{(i)} - y^{(i)}\|_{B_i} = |x^{(i)} - y^{(i)}|$ as well as $\|x^{(i+1)} - y^{(i+1)}\|_{B_i} = |x^{(i+1)} - y^{(i+1)}|$ which completes the proof. ◀

▶ **Lemma 7.** *If $p \ge 1$ then for i such that $4\eta|x-y| > 2^i$ we have $|g_i(x,y)| \le 2p2^i \cdot |x-y|^{p-1}$.*

Proof. For the sake of the proof, we will w.l.o.g. assume $\eta \leq 1/8$. Denote $A = |x^{(i)} - y^{(i)}|$, $B = |x^{(i+1)} - y^{(i+1)}|$, $A' = \max(0, A - 2^i)$ and $B' = \max(0, B - 2^{i+1})$. Observe that $|x-y|-2^i \leq A \leq |x-y|+2^i$ thus $|x-y|-2\cdot 2^i \leq A' \leq |x-y|$, and similarly $|x-y|-2\cdot 2^{i+1} \leq B' \leq |x-y|$ so $|A'-B'| \leq 2\cdot 2^i$. Assume w.l.o.g. that $A' \geq B'$. We bound

$$\begin{aligned} |g_i(x,y)| &= (A')^p - (A' - (A'-B'))^p \\ &\leq p(A'-B')(A')^{p-1} && \text{(by (4))} \\ &\leq 2p2^i \cdot |x-y|^{p-1} \end{aligned}$$ ◀

▶ **Lemma 8.** *If $p \geq 1$ then for i such that $4\eta|x-y| > 2^i$ we have $|\widehat{g}_i(x,y)| \leq 2p2^i \cdot |x-y|^{p-1}$.*

Proof. Follows by the same proof strategy as in proof of Lemma 7, replacing $|\cdot|$ with $\|\cdot\|_{B_i}$. ◀

▶ **Theorem 9.** *$\widehat{G}_{-u} = \sum_{i \geq -u} \widehat{g}_i$ approximates F_{-u} up to an additive $32 \cdot p \cdot \eta \cdot |x-y|^p$ term.*

Proof. We bound the difference between two terms:

$$\begin{aligned} |F_{-u}(x,y) - \sum_{i=-u}^{u} \widehat{g}_i(x,y)| &\leq \sum_{i=-u}^{\log_2(4\eta|x-y|)} (|\widehat{g}_i(x,y)| + |g_i(x,y)|) \\ &\leq 2 \cdot \left(\sum_{i=-\infty}^{\log_2(4\eta|x-y|)} 2^i \right) \cdot 2 \cdot p \cdot |x-y|^{p-1} \\ &\leq 32 \cdot \eta|x-y| \cdot p \cdot |x-y|^{p-1} \end{aligned}$$

where the bound follows from Lemma 5, 6, 7 and 8. ◀

We now show that F_{-u} is a close approximation of D (recall $D(x,y) = |x-y|^p$).

▶ **Lemma 10.** *For integers x, y there is $D(x,y) \cdot (1 - (2\ln 2)p/U) \leq F_{-u}(x,y) \leq D(x,y)$.*

Proof. For $x = y$ the lemma trivially holds, so for the rest of the proof we will assume $x \neq y$. As x, y are integers only, their smallest non-zero distance is 1. As $-u < 0$ the $|x-y| - 2^{-u} > 0$ and we bound $|x-y| \cdot (1 - 1/U) \leq \max(0, |x-y| - 2^{-u}) \leq |x-y|$. By (1) (when $p \geq 1$) or (6) (when $p \leq 1$) the claim follows. ◀

By combining Theorem 9 with the Lemma 10 above we conclude that additive error of Algorithm 4 at each position is $(32p \cdot \eta + \frac{p}{U}) \cdot |x-y|^p = p(\varepsilon/4 + 1/U) \cdot |x-y|^p \leq p\varepsilon|x-y|^p$ (since w.l.o.g. $\varepsilon \geq 4/U$), thus the relative error is $(1 + p\varepsilon/2)$.

Observe that each $\widehat{g}_i$ is effectively a function over the alphabet of size $B_i/2^i = 1/\eta$. Thus, the complexity of computing text-to-pattern distance using $\widehat{g}_i$ distance is $\mathcal{O}(\eta^{-1} n \log m)$, and iterating over at most $2u$ summands makes the total time $\mathcal{O}(\varepsilon^{-1} n \log m \log U)$.

Finally, since $p \geq 1$ and w.l.o.g. $\varepsilon \leq 1/p$, by (2) and (3) $(1 + p\varepsilon/2)$ approximation of ℓ_p^p distances is enough to guarantee $(1+\varepsilon)$ approximation of ℓ_p distances.

2.2 Algorithm for $0 < p \leq 1$

In this section we prove Theorem 2. We note that the algorithm presented in the previous section does not work, since in the proof of Lemma 7 and 8 we used the convexity of function $|t|^p$, which is no longer the case when $p < 1$.

However, we observe that Lemma 5 and 6 hold even when $0 < p \leq 1$. To combat the situation where adversarial input makes the estimates in Lemma 7 and 8 to grow too large, we use a very weak version of hashing. Specifically, we pick at random a linear function

$\sigma(t) = r \cdot t$, where $r \in [1, 9)$ is a random independent variable. Such function applied to the input makes its bit sequences appear more "random" while preserving the inner structure of the problem.

Consider a following approach:

▶ **Algorithm 11.**

1. *Fix* $\eta = \frac{\varepsilon \cdot p}{15555 \log U \ln 2}$.
2. *Pick* $r \in [1, 9)$ *uniformly at random.*
3. *Compute* $T' = r \cdot T$ *and* $P' = r \cdot P$.
4. *Use Algorithm 4 to compute* S', *text-to-pattern distance between* T' *and* P' *using* $\widehat{G}_{-u}$ *distance function.*
5. *Output* $S'' = S' \cdot r^{-1}$.

Now we analyze the expected error made by estimation from Algorithm 11. We denote the expected additive error of estimation of $(\ell_p)^p$ distances as

$$\mathrm{err}(x,y) \overset{\mathrm{def}}{=} \mathbb{E}_{r\in[1,9)}\Big[[\Big(\frac{1}{r}\Big)^p \Big|\widehat{G}_{-u}(rx,ry) - |rx-ry|^p\Big|\Big].$$

▶ **Theorem 12.** *The procedure of Algorithm 11 has the* expected *additive error* $err(x, y) \le \frac{\varepsilon p}{3 \ln 2}|x-y|^p$.

Proof. Assume that $x \neq y$, as otherwise the bound trivially follows. We bound the absolute error as follow, denoting $k = \log(8\eta|x - y|)$).

$$\begin{aligned}
\mathrm{err}(x,y) &\le \mathbb{E}_{r\in[1,9)}\Big[\Big(\frac{1}{r}\Big)^p \Big|\widehat{G}_{-u}(rx,ry) - F_{-u}(rx,ry)\Big|\Big] \\
&\quad + \mathbb{E}_{r\in[1,9)}\Big[|F_{-u}(rx,ry) - D(rx,ry)|\Big] \\
&\le \mathbb{E}_{r\in[1,9)}\Big[\Big|\sum_{i=-u}^{u}(\widehat{g}_i(rx,ry) - g_i(rx,ry))\Big|\Big] \\
&\quad + \mathbb{E}_{r\in[1,9)}\Big[\Big(\frac{1}{r}\Big)^p 2(\ln 2)\frac{p}{U}D(rx,ry)\Big] && ((1/r)^p \le 1) \\
&\le \sum_{i=-u}^{k}\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] + \mathbb{E}_{r\in[1,9)}\Big[\Big|\sum_{i=-u}^{k} g_i(rx,ry)\Big|\Big] \\
&\quad + 2(\ln 2)\frac{p}{U}|x-y|^p && \text{(Lemma 5, 6)}
\end{aligned}$$

Now, we bound the first two summands separately in following lemmas.

▶ **Lemma 13.** $|\sum_{i=-u}^{k} g_i(rx,ry)|$ *is upper bounded by* $32(\ln 2)\eta|x-y|^p$.

Proof. Since w.l.o.g. $\eta \le 1/32$ thus $2^{k+1} \le 1/2 \cdot r|x-y|$):

$$\begin{aligned}
\Big|\sum_{i=-u}^{k} g_i(rx,ry)\Big| &\le \Big|\sum_{i=-\infty}^{k} g_i(rx,ry)\Big| \\
&\le |G_{k+1}(rx,ry) - D(rx,ry)| \\
&\le ((r|x-y|)^p - (r|x-y| - 2^{k+1})^p) \\
&\le r^p|x-y|^p \cdot 2p(\ln 2)\frac{2^{k+1}}{r|x-y|} && \text{(by (8))} \\
&\le 32(\ln 2)\eta|x-y|^p. && (\ r^{p-1} \le 1)
\end{aligned}$$

◀

▶ **Lemma 14.** *For $i \leq k = \log(8\eta|x-y|)$ we have $\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] \leq (1152 + 192(\ln 2)))\eta|x-y|^p$.*

(Due to the space constraints, the proof of this Lemma is deferred to the appendix.)

By combining bounds from Lemma 13, and Lemma 14 we get:

$$\begin{aligned}
\mathrm{err}(x,y) &\leq \sum_{i=-u}^{k} \mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] + \mathbb{E}_{r\in[1,9)}\Bigg[\Bigg|\sum_{i=-u}^{k} g_i(rx,ry)\Bigg|\Bigg] + 2(\ln 2)\frac{p}{U}|x-y|^p \\
&\leq 2(\ln 2)\frac{p}{U}|x-y|^p + (32(\ln 2)\eta|x-y|^p + \sum_{i=-u}^{u}(1152+192(\ln 2)))\eta|x-y|^p \\
&\leq 2(\ln 2)\frac{p}{U}|x-y|^p + (32(\ln 2)\eta|x-y|^p + 2\log U(1152+192(\ln 2)))\eta|x-y|^p \\
&\leq 2(\ln 2)\frac{p}{U}|x-y|^p + (32(\ln 2) + 2\log U(1152+192(\ln 2))))\eta|x-y|^p \\
&\leq 2(\ln 2)\frac{p}{U}|x-y|^p + 2593\log U\eta|x-y|^p \\
&\leq \frac{\varepsilon p}{6\ln 2}|x-y|^p + \frac{\varepsilon p}{6\ln 2}|x-y|^p \qquad \text{w.l.o.g. } \varepsilon \geq \frac{12(\ln 2)^2}{U} \\
&\leq \frac{\varepsilon p}{3\ln 2}|x-y|^p
\end{aligned}$$
◀

To finish the proof of Theorem 2 we observe, that for any position i of output, Algorithm 11 outputs $S''[i]$ such that $\mathbb{E}_r[|(S''[i])^p - (S[i])^p|] \leq \frac{p\varepsilon}{3\ln 2}\cdot(S[i])^p$. By Markov's inequality it means that with probability 2/3 the relative error of $(\ell_p)^p$ approximation is at most $\frac{p}{\ln 2}\cdot\varepsilon$. Thus, by (5) and (7) relative error of ℓ_p approximation is ε with probability at least 2/3. Now a standard amplification procedure follows: invoke Algorithm 11 independently t times and take the median value from $S''_{(1)}[i], \ldots S''_{(t)}[i]$ as the final estimate $S_\varepsilon[i]$. Taking $t = \Theta(\log n)$ to be large enough makes the final estimate good with high probability, and by the union bound whole S_ε is a good estimate of S. The complexity of the whole procedure is thus $\mathcal{O}(\log n \cdot \log U \cdot \eta^{-1} \cdot n\log m) = \mathcal{O}(p^{-1}\varepsilon^{-1}n\log m\log^2 U\log n)$.

3 Hamming distances

As a final note we comment on a particularly simple form that Algorithm 11 takes for Hamming distances (limit case of $p = 0$).

$$\widehat{g}_i(x,y) = \begin{cases} 1 & \text{if } \|x^{(i)} - y^{(i)}\|_{B_i} = 1 \\ 0 & \text{otherwise,} \end{cases}$$

with Algorithm being simply: pick at random $r \in [1,9]$, apply it multiplicatively to the input, compute text-to-pattern distance using $\sum_i \widehat{g}_i$ function.

Taking a limit of $p \to 0$ in proof of Theorem 2, we reach that bound from Lemma 14 becomes

$$\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] \leq 24\eta$$

and since all other terms in error estimate have multiplicative term p in front, we reach

$$\mathrm{err}(x,y) \leq 2\log U\cdot\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] \leq 48\eta\log U.$$

We thus observe that expected relative error in estimation of Hamming distance is: $\mathbb{E}[S''[i] - S[i]] \leq 48\eta \log U \cdot S[i]$. With probability at least 2/3 the relative error is at most $144\eta \log U$. Setting $\eta = \frac{\varepsilon}{144 \log U}$ and repeating the randomized procedure $\Theta(\log n)$ with taking median for concentration completes the algorithm. The total runtime is, by a standard trick of reducing alphabet size to $2m$, $\mathcal{O}(\frac{n}{\varepsilon} \log^2 m \log n)$, and while it compares unfavorably to algorithm from [16] (in terms of runtime), it gives another insight on why $\widetilde{\mathcal{O}}(n/\varepsilon)$ time algorithm is possible for Hamming distance version of pattern matching.

References

1 Karl R. Abrahamson. Generalized String Matching. *SIAM J. Comput.*, 16(6):1039–1051, 1987.

2 Amihood Amir, Moshe Lewenstein, and Ely Porat. Faster algorithms for string matching with k mismatches. *J. Algorithms*, 50(2):257–275, 2004. doi:10.1016/S0196-6774(03)00097-X.

3 Amihood Amir, Ohad Lipsky, Ely Porat, and Julia Umanski. Approximate Matching in the L_1 Metric. In *CPM*, pages 91–103, 2005. doi:10.1007/11496656_9.

4 Amit Chakrabarti and Oded Regev. An Optimal Lower Bound on the Communication Complexity of Gap-Hamming-Distance. *SIAM J. Comput.*, 41(5):1299–1317, 2012. doi:10.1137/120861072.

5 Peter Clifford, Raphaël Clifford, and Costas S. Iliopoulos. Faster Algorithms for δ,γ-Matching and Related Problems. In *CPM*, pages 68–78, 2005. doi:10.1007/11496656_7.

6 Raphaël Clifford. Matrix multiplication and pattern matching under Hamming norm. http://www.cs.bris.ac.uk/Research/Algorithms/events/BAD09/BAD09/Talks/BAD09-Hammingnotes.pdf, 2009. Retrieved March 2017.

7 Raphaël Clifford, Allyx Fontaine, Ely Porat, Benjamin Sach, and Tatiana A. Starikovskaya. The k-mismatch problem revisited. In *SODA*, pages 2039–2052. SIAM, 2016.

8 M. J. Fischer and M. S. Paterson. STRING-MATCHING AND OTHER PRODUCTS. Technical report, Massachusetts Institute of Technology, 1974.

9 Sumit Ganguly. Taylor Polynomial Estimator for Estimating Frequency Moments. In *ICALP*, pages 542–553, 2015. doi:10.1007/978-3-662-47672-7_44.

10 Paweł Gawrychowski and Przemysław Uznański. Towards Unified Approximate Pattern Matching for Hamming and L_1 Distance. In *ICALP*, pages 62:1–62:13, 2018. doi:10.4230/LIPIcs.ICALP.2018.62.

11 Daniel Graf, Karim Labib, and Przemysław Uznański. Brief Announcement: Hamming Distance Completeness and Sparse Matrix Multiplication. In *ICALP*, pages 109:1–109:4, 2018. doi:10.4230/LIPIcs.ICALP.2018.109.

12 T. S. Jayram, Ravi Kumar, and D. Sivakumar. The One-Way Communication Complexity of Hamming Distance. *Theory of Computing*, 4(1):129–135, 2008. doi:10.4086/toc.2008.v004a006.

13 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the Exact Space Complexity of Sketching and Streaming Small Norms. In *SODA*, pages 1161–1178, 2010. doi:10.1137/1.9781611973075.93.

14 Howard J. Karloff. Fast Algorithms for Approximately Counting Mismatches. *Inf. Process. Lett.*, 48(2):53–60, 1993. doi:10.1016/0020-0190(93)90177-B.

15 Tsvi Kopelowitz and Ely Porat. Breaking the Variance: Approximating the Hamming Distance in $1/\epsilon$ Time Per Alignment. In *FOCS*, pages 601–613, 2015. doi:10.1109/FOCS.2015.43.

16 Tsvi Kopelowitz and Ely Porat. A Simple Algorithm for Approximating the Text-To-Pattern Hamming Distance. In *SOSA*, pages 10:1–10:5, 2018. doi:10.4230/OASIcs.SOSA.2018.10.

17 Yi Li and David P. Woodruff. A Tight Lower Bound for High Frequency Moment Estimation with Small Error. In *APPROX-RANDOM*, pages 623–638, 2013. doi:10.1007/978-3-642-40328-6_43.

18 Ohad Lipsky. Eficient Distance Computations. Master's thesis, Bar-Ilan University. Department of Mathematics and Computer Science., 2003.

19 Ohad Lipsky and Ely Porat. L_1 pattern matching lower bound. *Inf. Process. Lett.*, 105(4):141–143, 2008. doi:10.1016/j.ipl.2007.08.011.

20 Ohad Lipsky and Ely Porat. Approximate Pattern Matching with the L_1, L_2 and L_∞ Metrics. *Algorithmica*, 60(2):335–348, 2011. doi:10.1007/s00453-009-9345-9.

21 Marcin Mucha, Karol Węgrzycki, and Michał Włodarczyk. A Subquadratic Approximation Scheme for Partition. In *SODA*, pages 70–88, 2019. doi:10.1137/1.9781611975482.5.

22 John Nolan. *Stable distributions: models for heavy-tailed data*. Birkhauser New York, 2003.

23 Ely Porat and Klim Efremenko. Approximating general metric distances between a pattern and a text. In *SODA*, pages 419–427, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347128.

24 Jan Studený and Przemysław Uznański. Approximating Approximate Pattern Matching. *CoRR*, abs/1810.01676, 2018. arXiv:1810.01676.

25 David P. Woodruff. Optimal space lower bounds for all frequency moments. In *SODA*, pages 167–175, 2004. URL: http://dl.acm.org/citation.cfm?id=982792.982817.

A Omitted proofs

▶ **Lemma 14.** *For $i \le k = \log(8\eta|x-y|)$ we have $\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] \le (1152 + 192(\ln 2)))\eta|x-y|^p$.*

Proof. First, we define symbols A, B, A', B' to be parts of the $\widehat{g}_i$.

$$\begin{aligned}
A &= \|(rx)^{(i)} - (ry)^{(i)}\|_{B_i}\\
B &= \|(rx)^{(i+1)} - (ry)^{(i+1)}\|_{B_i}\\
A' &= \max(0, A - 2^i)\\
B' &= \max(0, B - 2^{i+1})
\end{aligned}$$

Repeating reasoning from proof of Lemma 7, we get

$$|A' - B'| \le 2 \cdot 2^i \tag{9}$$

$$\|rx - ry\|_{B_i} - 2 \cdot 2^i \le A' \le \|rx - ry\|_{B_i} \tag{10}$$

$$\|rx - ry\|_{B_i} - 2 \cdot 2^{i+1} \le B' \le \|rx - ry\|_{B_i} \tag{11}$$

We also bound $B_i = 2^i/\eta \le 2^k/\eta = 8|x-y|$. Now let's bound the $|\widehat{g}_i(rx,ry)|$. A simple bound that comes from the definition of $\widehat{g}_i$ gives us:

$$|\widehat{g}_i(rx,ry)| = |A'^p - B'^p| \le \max(A'^p, B'^p) \le \|rx - ry\|_{B_i}^p. \qquad \text{(Use of 10,11)} \tag{12}$$

Unfortunately, this bound is not tight enough for larger values of $\|rx - ry\|_{B_i}$, so for $\|rx - ry\|_{B_i} \ge 6 \cdot 2^i$, we prove stronger bound:

$$\begin{aligned}
|\widehat{g}_i(rx,ry)| &= |(A')^p - (B')^p|\\
&= \max(A', B')^p - \min(A', B')^p\\
&= \max(A', B')^p - (\max(A', B') - |A' - B'|)^p\\
&= \max(A', B')^p \left(1 - \left(1 - \frac{|A' - B'|}{\max(A', B')}\right)^p\right)\\
&\le \|rx - ry\|_{B_i}^p \cdot (1 - (1 - \frac{2 \cdot 2^i}{\|rx - ry\|_{B_i} - 2 \cdot 2^i})^p)\\
&\le \|rx - ry\|_{B_i}^p \cdot (1 - (1 - \frac{3 \cdot 2^i}{\|rx - ry\|_{B_i}})^p)\\
&\le 6p(\ln 2)\|rx - ry\|_{B_i}^{p-1} \cdot 2^i \qquad (\|rx - ry\|_{B_i} \ge 6 \cdot 2^i, \text{ by } (6))
\end{aligned}$$

The *norm* function $\|x\|_{B_i} = \min(x \bmod B_i, B_i - (x \bmod B_i))$ is in fact a triangle wave function varying between 0 and $B_i/2$ with periodicity of B_i. So if the input is a random variable that follows uniform distribution at interval that is larger than its period (in our case B_i), the output has piece-wise uniform distribution, and its probability density function can be bounded by two times the probability density function of the uniform distribution for the whole domain. Formally, if $X = U(a,b)$ with $b - a \geq B_i$ then for $Y = \|X\|_{B_i}$ its probability density function $f_Y(y)$ is:

$$f_Y(y) \leq \frac{2}{B_i/2} \text{ for } 0 \leq y \leq B_i/2 \tag{13}$$

As the input in the expression $\|rx - ry\|_{B_i}$ to the *norm* function is uniformly distributed between $a = |x-y|$ and $b = 9|x-y|$ and $B_i \leq 8|x-y|$, we can use 13 to bound the probability density function of the $Z = \|rx - ry\|_{B_i}$ by $f_Z(y) \leq \frac{2}{B_i/2}$.

Now when we have the approximate probability density function (namely its upper bound) we can condition on the value of $\|rx - ry\|_{B_i}$ to be able to use the bounds for small and large values of $\|rx - ry\|_{B_i}$.

$$\begin{aligned}
&\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)|\Big] = \\
&\quad = \mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)| \;\Big|\; \|rx-ry\|_{B_i} \leq 6\eta B_i\Big] \Pr_{r\in[1,9)}\Big[\|rx-ry\|_{B_i} \leq 6\eta B_i\Big] + \\
&\quad\quad + \mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)| \;\Big|\; \|rx-ry\|_{B_i} > 6\eta B_i\Big] \Pr_{r\in[1,9)}\Big[\|rx-ry\|_{B_i} > 6\eta B_i\Big]
\end{aligned}$$

We bound those two summands separately. Now, bound on the first part:

$$\begin{aligned}
&\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)| \;\Big|\; \|rx-ry\|_{B_i} \leq 6\eta B_i\Big] \Pr_{r\in[1,9)}\Big[\|rx-ry\|_{B_i} \leq 6\eta B_i\Big] \leq \\
&\quad \leq \mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)| \;\Big|\; \|rx-ry\|_{B_i} \leq 6\eta B_i\Big] 24\eta \\
&\quad \leq \mathbb{E}_{r\in[1,9)}\Big[\|rx-ry\|^p_{B_i} \;\Big|\; \|rx-ry\|_{B_i} \leq 6\eta B_i\Big] 24\eta && \text{(by 12)} \\
&\quad \leq (6\eta B_i)^p 24\eta \\
&\quad \leq 24\eta(6\cdot 2^i)^p \\
&\quad \leq 24\cdot 6\eta(8\eta|x-y|)^p \\
&\quad \leq 1152\eta|x-y|^p
\end{aligned}$$

And on the second part:

$$\begin{aligned}
&\mathbb{E}_{r\in[1,9)}\Big[|\widehat{g}_i(rx,ry)| \Big| \|rx-ry\|_{B_i} > 6\eta B_i\Big] \Pr_{r\in[1,9)}\Big[\|rx-ry\|_{B_i} > 6\eta B_i\Big] \leq \\
&\quad \leq \mathbb{E}_{r\in[1,9)}\Big[6p(\ln 2)\|rx-ry\|^{p-1}_{B_i}\cdot 2^i \Big| \|rx-ry\|_{B_i} > 6\eta B_i\Big] \cdot \\
&\quad\quad \cdot \Pr_{r\in[1,9)}\Big[\|rx-ry\|_{B_i} > 6\eta B_i\Big]
\end{aligned}$$

$$\ldots \le \int_1^9 6p(\ln 2)\|rx - ry\|_{B_i}^{p-1} \cdot 2^i \frac{1}{8} \cdot \mathbf{1}[\|rx - ry\|_{B_i} > 6\eta B_i]\, dr$$

(1/8 is the density of r.v. r, $\mathbf{1}[\cdot]$ is the indicator function)

$$\le 6p(\ln 2) \cdot 2^i \int_0^{B_i/2} z^{p-1} \frac{2}{B_i/2} \mathbf{1}[z > 6\eta B_i]\, dz$$

(changed to r.v. $z = \|rx - ry\|_{B_i}$)

$$\le p(\ln 2)\frac{24}{B_i} \cdot 2^i \int_0^{B_i/2} z^{p-1} dz$$
$$\le (\ln 2)\frac{24}{B_i} \cdot 2^i \left(\frac{B_i}{2}\right)^p$$
$$\le 24(\ln 2)B_i^{p-1} \cdot 2^i$$
$$\le 24(\ln 2)2^{ip}\eta^{-p+1}$$
$$\le 24(\ln 2)(8\eta|x-y|)^p\eta^{-p+1}$$
$$\le 192(\ln 2)\eta|x-y|^p$$

So finally, we reach:

$$\mathbb{E}_{r\in[1,9)}\left[|\widehat{g}_i(rx, ry)|\right] =$$
$$= \mathbb{E}_{r\in[1,9)}\left[|\widehat{g}_i(rx, ry)| \,\middle|\, \|rx - ry\|_{B_i} \le 6\eta B_i\right] \Pr_{r\in[1,9)}\left[\|rx - ry\|_{B_i} \le 6\eta B_i\right] +$$
$$+ \mathbb{E}_{r\in[1,9)}\left[|\widehat{g}_i(rx, ry)| \,\middle|\, \|rx - ry\|_{B_i} > 6\eta B_i\right] \Pr_{r\in[1,9)}\left[\|rx - ry\|_{B_i} > 6\eta B_i\right]$$
$$\le 1152\eta|x-y|^p + 192(\ln 2)\eta|x-y|^p$$
$$\le (1152 + 192(\ln 2))\eta|x-y|^p$$

◀