Thesis: Chains with a Variety of Distances

Tianyi Ma

May 23, 2021

Abstract

The chains has been well-studied when the underlying function is either the distance function or the dot product function. In this paper, we survey current results of such chains, as well as the Erdos problem, especially the special case. We will present some computational results that shows preliminary evidences that support the current conjecture for the Erdos problem, and then consider the chains with a mixture of both functions.

1 Introduction

Let $\mathbb{F}_q$ denote a finite field with q elements, where q is a power of an odd primes. In a special case when $q = p$ is a prime, we use the notation $\mathbb{Z}_p$. How large are the chains when the underlying functions are the distance function and the dot product function respectively? Previous studies have proved the following results:

Theorem 1.1. Let $E \subset \mathbb{F}_q^d$ and define the following function

$$w(t) = |\{(x, y) \in E \times E \colon ||x - y|| = t\}|.$$

Then

$$w(t) = \frac{|E|^2}{q} + R(t)$$

where

$$|R(t)| \leq 2|E|q^{\frac{d-1}{2}}$$

Theorem 1.2. Let $E \subset \mathbb{F}_q^d$ and define the incidence function

$$v(t) = |\{(x, y) \in E \times E \colon x \cdot y = t\}|.$$

Then

$$v(t) = \frac{|E|^2}{q} + R'(t)$$

where

$$\begin{cases} |R'(t)| \leq |E|q^{\frac{d-1}{2}} & \text{for } t \neq 0 \\ |R'(0)| \leq |E|q^{\frac{d}{2}} \end{cases}$$

Generalizing these two thoerems, previous studies have obtained the following lemmas: **Lemma 1.3.** Let f, g two positive function in $\mathbb{F}_q^d$. Then

$$\sum_{||x-y||=t} f(x)g(y) = ||f||_1||g||_1 q^{-1} + R(t)$$

and

$$|R(t)| \leq 2||f||_2||g||_2 q^{\frac{d-1}{2}}$$

where $||f||_1 = \sum_{\mathbb{F}_q^d} f(x)$, $||f||_2 = (\sum_{\mathbb{F}_q^d} f^2(x))^{\frac{1}{2}}$.
Lemma 1.3. Let f, g two positive function in $\mathbb{F}_q^d$. Then

$$\sum_{x \cdot y=t} f(x)g(y) = ||f||_1||g||_1 q^{-1} + R(t)$$

and

$$|R(t)| \leq ||f||_2||g||_2 q^{\frac{d-1}{2}}$$

Another related problem is the Erdos problem. Specifically, let $\mathbb{F}_q^*$ denote the multiplicative group of $\mathbb{F}_q$. How large does $A \subset \mathbb{F}_q$ need to be to make sure that

$$dA^2 = \underbrace{A^2 + \cdots + A^2}_{\text{d times}} \supseteq \mathbb{F}_q^*?$$

Define

$$A^2 = A \cdot A = \{a \cdot a' \colon a, a' \in A\} and A + A = \{a + a' \colon a, a' \in A\}.$$

Current results have proved that in the specific case that $d = 2$, $A^2 + A^2$ covers $\mathbb{F}_q^*$ if $|A| > q^{\frac{3}{4}}$. We also have the following conjectures:
Conjecture 1.5. Let $A \subset \mathbb{F}_q$, where $\mathbb{F}_q$ is a finite field with q a prime and A a subgroup of $\mathbb{F}_q$, such that $|A| > q^{\frac{1}{2}}$,

$$\mathbb{F}_q^* \subset A^2 + A^2$$

Conjecture 1.6. Let $A \subset \mathbb{F}_q[i]$, where $\mathbb{F}_q[i]$ is the Gaussian Integer of the finite field $\mathbb{F}_q$, q a prime, and A a subgroup of $\mathbb{F}_q[i]$, such that $|A| > q$,

$$\mathbb{F}_q^*[i] \subset A^2 + A^2$$

In the following sections, we are going to provide code that tests these conjectures computationally and present the results of such experiments, which offer preliminary evidences that confirm the conjectures.
In addition to these results, we also derived a bound for the chains with underlying function as a mixture of the distance function and the dot product function.

Theorem 1.7. Let $E \subset \mathbb{F}_q^d$ and define the following function

$$g(a) = |\{(x, y, z) \in E \times E \times E \colon x \cdot y = a, x \cdot z = a\}|.$$

Then

$$g(a) = \frac{|E|^3}{q^2} + R_1(a, b) + R_2(a, b)$$

where

$$|R_1(a, b)| \leq |E|^2 q^{\frac{d-3}{2}}$$
$$|R_2(a, b)| \leq g(a)^{\frac{1}{2}} |E|^{\frac{1}{2}} q^{\frac{d-1}{2}}$$

Theorem 1.8. Let $E \subset \mathbb{F}_q^d$ and define the following function

$$h(a, b) = |\{(x, y, z) \in E \times E \times E \colon ||x - y|| = a, x \cdot z = b\}|.$$

Then

$$h(a, b) = \frac{|E|^3}{q^2} + R_1(a, b) + R_2(a, b)$$

where

$$|R_1(a, b)| \leq |E|^2 q^{\frac{d-3}{2}}$$
$$|R_2(a, b)| \leq h(a, b)^{\frac{1}{2}} |E|^{\frac{1}{2}} q^{\frac{d-1}{2}}$$

2 Experiments

We have experimented the conjectures computationally using the following code:

```
import pandas as pd
from math import comb
from matplotlib import pyplot as plt
import time

n = 3000000

is_prime = [False, False] + [True] * (n - 1)
primes = [2]

```

```python
for j in range(4, n + 1, 2):
    is_prime[j] = False

for i in range(3, n + 1, 2):
    if is_prime[i]:
        primes.append(i)
        for j in range(i * i, n + 1, i):
            is_prime[j] = False


def findPrimes(q_min, q_max):
    q_cand = []
    for q in primes:
        if q >= q_min and q <= q_max:
            q_cand.append(q)
    return q_cand


def addmod(a, b, q):
    return (a+b)%q

def mulmod(a, b, q):
    return (a*b)%q


# Given a number q and a power, return a list of divisors > 0.95q^0.5 and <= 1.5q^0.5
def findDivisors(q_, power):
    divisors = []
    q = q_+1
    threshold = int(q**power)
    for i in range(int(threshold*0.95)+1, int(threshold*1.5)+1): # 0.95, 1.5
        if q_%i == 0:
            divisors.append(i)
    return divisors


def calMax(n):
    return comb(n, 2)+2*n


# return a list of elements in a finite group Fq
def createGroup(q):
    elements = []
    for i in range(q):
        elements.append(i)
```

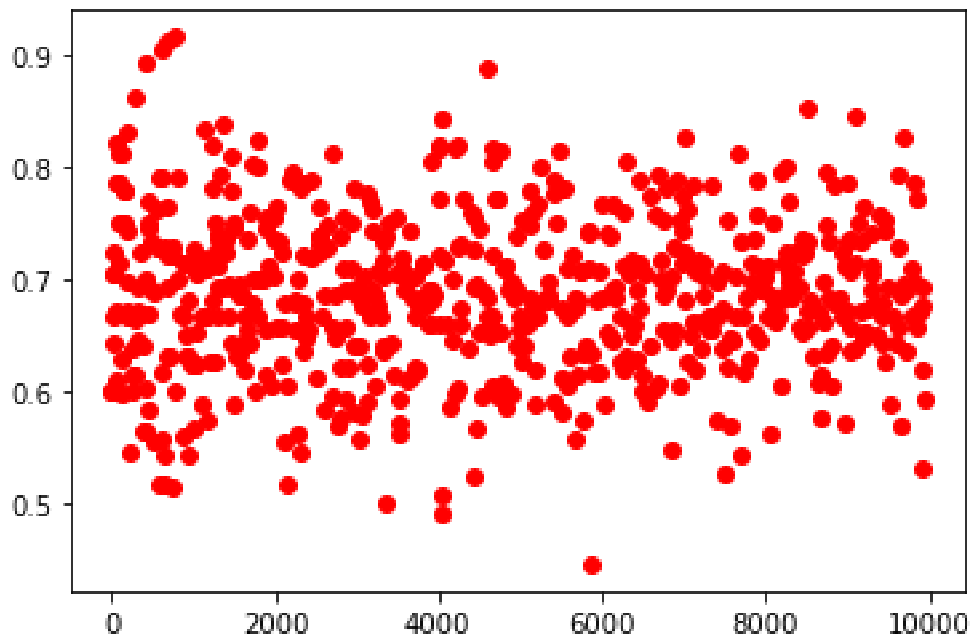
```
    return elements


# param:
# q: |G|
# g: the generator
def generateCyclicGroup(q, g):
    cyclicGroup = [g]
    a = g
    while a != 1:
        a = mulmod(a, g, q)
        cyclicGroup.append(a)
    return cyclicGroup


# param:
# mulG: G*
# n: one possible order of multiplicative subgroups. Divides |G| = q
def findMulSubgroup(mulG, n):
    generators = mulG.copy()
    generators.remove(1)
    q = len(mulG)+1
    mulSubgroup = []
    for g in generators:
        mulSubgroup = generateCyclicGroup(q, g)
        if len(mulSubgroup) == n:
            break
    return mulSubgroup


def compute2A2(A, q):
    twoA2_set = set()
    for a1 in A:
        for a2 in A:
            twoa2 = addmod(a1, a2, q)
            twoA2_set.add(twoa2)
    twoA2 = list(twoA2_set)
    return twoA2


# MAIN
# Find all multiplicative subgroups A of G with orders around q^0.5 and compute 2A^2.
# If |2A^2|/(q-1)>0.5, add q,|A|,A,|2A^2|,2A^2,|2A^2|/q to the dataframe
# params:
# q_min, q_max: range of primes you want to use
```

```
# method: one integer of [1, 2, 3, 4, 5], specifies which method you want to use to ca
#         please refer to findDivisors1-4 for specific parameters

def main1(q_min, q_max, method):
    q_cand = findPrimes(q_min, q_max)
    q_A_list = []
    for q in q_cand:
        G = createGroup(q)
        divisors = findDivisors(q-1, 1/2)
        mulG = G.copy()
        mulG.remove(0)
        for divisor in divisors:
            A = findMulSubgroup(mulG, divisor) # find actual multiplicative subgroups A
            q_A_list.append([q, A])
    return q_A_list


def main2(q_A_list):
    p50_, p00, p10, p20, p30, p40, p50, p60, p70, p80, p90, p = (0 for i in range(12))
    validSubgroups_list, invalidSubgroups_list, worstCases_list = ([] for i in range(3))
    col = ["q","|A|","A","|2A^2|","2A^2","(|A| 2)","|2A^2|/(|A| 2)","|2A^2|/q"]
    worstCaseRatio = 1
    worstCase = [None]*8
    curr_q = 0
    index = -1
    for q_A in q_A_list:
        q = q_A[0]
        A = q_A[1]
        index += 1
        if q != curr_q:
            curr_q = q
            if worstCase[0] != None:
                worstCases_list.append(worstCase)
            worstCaseRatio = 1
            worstCase = [None]*8
        if len(A) != 0:
            divisor = len(A)
            print(q, divisor)
            p+=1
            start = time.time()
            dA2 = compute2A2(A, q) # compute 2A^2
            print(time.time() - start)
            length = len(dA2)
            pctg = length/q
            bestCase = calMax(divisor)
```

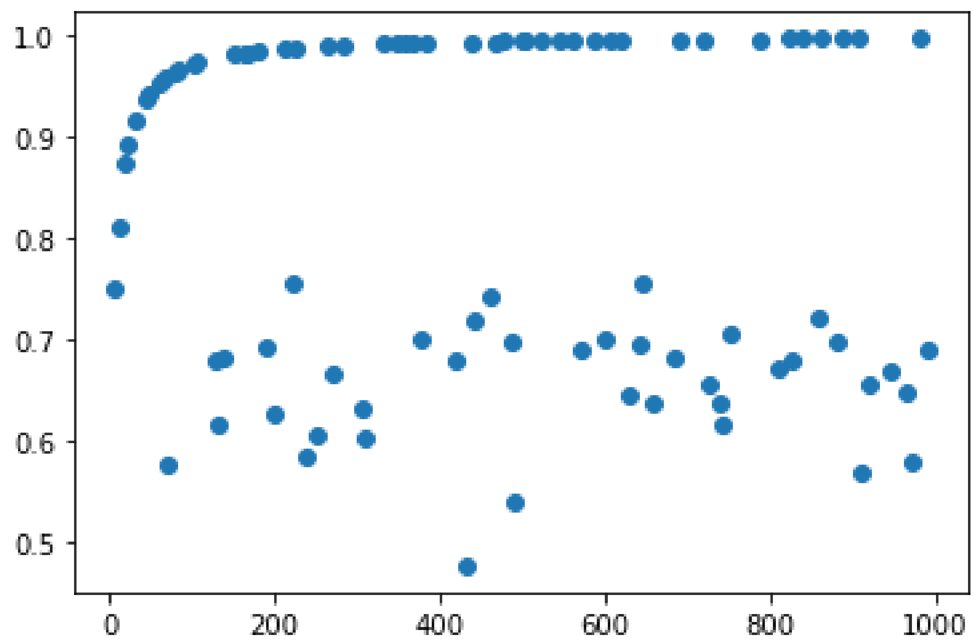
```
            if bestCase >= q:
                bestCase = q
            currentCaseRatio = length/bestCase
            if currentCaseRatio < worstCaseRatio:
                worstCaseRatio = currentCaseRatio
                worstCase = [q, divisor, A, length, dA2, bestCase, currentCaseRatio, pct
            if pctg >= 0.5: # check if 2A^2 is large enough (> |G|/2)
                p50_+=1
                validSubgroups_list.append([q, divisor, A, length, dA2, bestCase, curren
                if pctg < 0.6:
                    p50+=1
                elif pctg < 0.7:
                    p60+=1
                elif pctg < 0.8:
                    p70+=1
                elif pctg < 0.9:
                    p80+=1
                else:
                    p90+=1
            else:
                invalidSubgroups_list.append([q, divisor, A, length, dA2, bestCase, curr
                if pctg >= 0.4:
                    p40+=1
                elif pctg >= 0.3:
                    p30+=1
                elif pctg >= 0.2:
                    p20+=1
                elif pctg >= 0.1:
                    p10+=1
                else:
                    p00+=1
        if (index+1) == len(q_A_list) and worstCase[0] != None:
            worstCases_list.append(worstCase)


    # general info and stats
    validSubgroups = pd.DataFrame(validSubgroups_list, columns = col) # info of valid su
    invalidSubgroups = pd.DataFrame(invalidSubgroups_list, columns = col) # info of inva
    worstCases = pd.DataFrame(worstCases_list, columns = col)
    stat_list = [p00/p, p10/p, p20/p, p30/p, p40/p, p50/p, p60/p, p70/p, p80/p, p90/p, p
    stat = pd.Series(stat_list, index=["0-10%","10-20%","20-30%","30-40%","40-50%","50-6
    print("|2A^2|/q distribution")
    print(stat)

    results = {"validSubgroups":validSubgroups, "invalidSubgroups":invalidSubgroups, "wo
```

```
    return results
```

Basically, the code find all the primes up to $1,000,000$. For each prime q, the algorithm computes all subgroups A with $|A| > q^{\frac{1}{2}}$, and calculate $A^2 + A^2$ for each A. It then documents $|A^2 + A^2|$ and computes the portion of $\mathbb{F}_q^*$ it covers. Then, the program selects the subgroup that covers the least portion of the multiplicative finite field, which we call "the worst cases". Finally, it stores the statistical data to a dataframe and save it to a local file. The following is a visualization of the statistics of primes up to $10,000$ with the x-axis being the primes and the y-axis being the portion of the multiplicative finite field the worst cases cover.



As you can see, $A^2 + A^2$, even in the worst cases, consistently covers more than half of $\mathbb{F}_q^*$. We also built a program to test Conjecture 1.4. as well. The algorithm is similar, with only $\mathbb{F}_q^*$ being replaced by $\mathbb{F}_q^*[i]$ and primes up to $1,000$. The code is obmitted here since it's similar to the above code. In this case, we obtain the following graph of the statistics of the worst cases:



As you can see, $A^2 + A^2$, even in the worst cases, also consistently covers more than half of $\mathbb{F}_q^*[i]$.

3 Chains with underlying function as a mixture of functions

This study proves the following result:

Theorem 1.7. Let $E \subset \mathbb{F}_q^d$ and define the following function

$$g(a) = |\{(x, y, z) \in E \times E \times E \colon x \cdot y = a, x \cdot z = a\}|.$$

Then if $|E| \geq q^{\frac{d+1}{2}}$,

$$g(a) = \frac{|E|^3}{q^2} + R_1(a, b) + R_2(a, b)$$

where

$$|R_1(a)| \leq |E|^2 q^{\frac{d-3}{2}}$$
$$|R_2(a)| \leq g(a)^{\frac{1}{2}} |E|^{\frac{1}{2}} q^{\frac{d-1}{2}}$$

Proof. Another way to write $g(a)$ is:

$$g(a) = \sum_{x \cdot y = a, x \cdot z = a} E(x)E(y)E(z)$$

Let $f(x) = \left(\sum_{x\cdot z=a} E(z)\right) E(x)$, and then by Lemma 1.3, we have:

$$\sum_{x\cdot y=a} f(x)E(y) = ||f||_1 |E| q^{-1} + R_2(a) \tag{1}$$

$$|R_2(a)| \leq ||f||_2 |E|^{\frac{1}{2}} q^{\frac{d-1}{2}} \tag{2}$$

Applying Lemma 1.3 again to $||f||_1$, we have:

$$||f||_1 = \sum_{x\cdot z=a} E(x)E(z) = |E|^2 q^{-1} + R(a) \tag{3}$$

$$|R(a)| \leq |E| q^{\frac{d-1}{2}}$$

And

$$||f||_2^2 = \sum_{x\cdot z=a, x\cdot w=a} E(x)E(z)E(w) = g(a)$$

$$||f||_2 = g^{\frac{1}{2}}(a) \tag{4}$$

Plug in Equation (3) to (1) and (4) to (2), we have:

$$g(a) = |E|^3 q^{-2} + R_1(a) + R_2(a)$$

where

$$|R_1(a)| \leq |E| q^{\frac{d-1}{2}} \times |E| q^{-1} = |E|^2 q^{\frac{d-3}{2}}$$

$$|R_2(a)| \leq g^{\frac{1}{2}}(a) |E|^{\frac{1}{2}} q^{\frac{d-1}{2}}$$

To bound $|R_1(a)|$, we want

$$|R_1(a)| \leq |E|^2 q^{\frac{d-3}{2}} \leq |E|^3 q^{-2}$$

$$|E| \geq q^{\frac{d+1}{2}} \tag{5}$$

To bound $|R_2(a)|$, we want

$$|R_2(a)| \leq g^{\frac{1}{2}}(a) |E|^{\frac{1}{2}} q^{\frac{d-1}{2}} \leq |E|^3 q^{-2}$$

$$g(a) \leq \frac{|E|^5}{q^{d+3}} \tag{6}$$

Assuming Equation (5) is satisfied and plugging it in Equation (6),

$$\frac{|E|^5}{q^{d+3}} \geq q^{\frac{5}{2}(d+1)-(d+3)} = q^{\frac{3}{2}d - \frac{1}{2}}$$

Notice if Equation (5) is satisfied then the following is also true:

$$|E|^3 q^{-2} \geq q^{\frac{3}{2}d-\frac{1}{2}}$$

If $g(a) \leq q^{\frac{3}{2}d-\frac{1}{2}}$ is true, then $|R_2(a)| \leq |E|^3 q^{-2}$ is small. Suppose that is not the case, then

$$g(a) \geq q^{\frac{3}{2}d-\frac{1}{2}} = |E|^3 q^{-2}$$

So in this case we can ignore the error, $R_2(a)$. □

Theorem 1.8. Let $E \subset \mathbb{F}_q^d$ and define the following function

$$h(a,b) = |\{(x,y,z) \in E \times E \times E \colon ||x-y|| = a, x \cdot z = b\}|.$$

Then if $|E| \geq q^{\frac{d+1}{2}}$,

$$h(a,b) = \frac{|E|^3}{q^2} + R(a,b)$$

where

$$|R(a,b)| \leq 3|E|^2 q^{\frac{d-3}{2}}$$

Proof. To prove Theorem 1.8, we would use Theorem 1.7. The idea is similar as well. Another way to write $h(a,b)$ is:

$$h(a,b) = \sum_{||x-y||=a, x\cdot z=b} E(x)E(y)E(z)$$

Let $f(x) = (\sum_{x\cdot z=b} E(z))E(x)$, and then by Lemma 1.4, we have:

$$\sum_{||x-y||=a} f(x)E(y) = ||f||_1 |E| q^{-1} + R'(a,b) \tag{7}$$

$$|R'(a,b)| \leq 2||f||_2 |E|^{\frac{1}{2}} q^{\frac{d-1}{2}} \tag{8}$$

Applying Lemma 1.3 to $||f||_1$, we have:

$$||f||_1 = \sum_{x\cdot z=b} E(x)E(z) = |E|^2 q^{-1} + R(b) \tag{9}$$

$$|R(b)| \leq |E| q^{\frac{d-1}{2}}$$

And

$$||f||_2^2 = \sum_{x\cdot z=b, x\cdot w=b} E(x)E(z)E(w) = g(b)$$

$$||f||_2 = g^{\frac{1}{2}}(b)$$

By Theorem 1.7, $g(b) \sim |E|^3 q^{-2}$ given $|E| \geq q^{\frac{d+1}{2}}$, so Equation (8) becomes:

$$\begin{aligned}|R'(a,b)| &\leq 2||f||_2 |E|^{\frac{1}{2}} q^{\frac{d-1}{2}} \sim 2|E|^{\frac{3}{2}} q^{-1} \times |E|^{\frac{1}{2}} q^{\frac{d-1}{2}} \\ &= 2|E|^2 q^{\frac{d-3}{2}}\end{aligned}$$

Plugging Equation (9) to Equation (7),

$$\begin{aligned}h(a,b) &= \sum_{||x-y||=a} f(x)E(y) = |E|^2 q^{-1} \times |E| q^{-1} + R(b)|E|q^{-1} + R'(a,b) \\ &= |E|^3 q^{-2} + R(b)|E|q^{-1} + R'(a,b)\end{aligned}$$

Because $|R(b)| \leq |E| q^{\frac{d-1}{2}}$,

$$|R(b)|E|q^{-1}| \leq |E|^2 q^{\frac{d-3}{2}}$$

Because $|R'(a,b)| \leq 2|E|^2 q^{\frac{d-3}{2}}$ nad the Triangle Inequality,

$$|R(a,b)| = |R(b)|E|q^{-1} + R'(a,b)| \leq |R(b)|E|q^{-1}| + |R'(a,b)| \leq 3|E|^2 q^{\frac{d-3}{2}}$$

$\square$

The above are my proof for now, please let me know anything I missed. Also, I do have some question about Theorem 1.7, I'm sure I'm missing something here. Using the notation of my proof above, we have that if $|E| \geq q^{\frac{d+1}{2}}$, $|R_1(a)| \leq |E|^2 q^{\frac{d-3}{2}}$. So it is possible that $|R_1(a)| = -|E|^2 q^{\frac{d-3}{2}} = -|E|^3 q^{-2}$ when $|E| \sim q^{\frac{d+1}{2}}$. Also it could be that $g(a) = \frac{|E|^5}{q^{d+3}}$ so that it could be that $R_2(a) = -|E|^2 q^{\frac{d-3}{2}} = -|E|^3 q^{-2}$. So according to the theorem it's possible $g(a) = |E|^3 q^{-2} - |E|^3 q^{-2} - |E|^3 q^{-2} = -|E|^3 q^{-2}$ which is clearly not possible since $g(a) \geq 0$. So it can only garantee $g(a) \geq 0$. Besides, suppose we could ignore $R_1(a)$ and we only look at $R_2(a)$. Suppose $g(a) > \frac{|E|^5}{q^{d+3}} = |E|^3 q^{-2}$, we have no upper bound for $g(a)$ or $R_2(a)$ and thus no upper bound for $R'(a,b)$ in the proof of Theorem 1.8. So $g(a)$ could be as large as q^d, so in the proof of Theorem 1.8 $|R'(a,b)| \leq 2|E|^{\frac{1}{2}} q^{d-\frac{1}{2}} \sim 2q^d$. And so it is possible that $R'(a,b) = -2q^d$ and which makes $h(a,b)$ theoratically less than 0 which is clearly impossible so it has to be that $h(a,b) = 0$ in this case. So it can only garantee $h(a,b) \geq 0$. Please let me know where I thought wrong. Thank you.

References

[1] D. Hart and A. Iosevich, *Sums and products in finite fields: an integral geometric viewpoint* Contemporary Mathematics, 129–135, (2008).

[2] D. Hart, A. Iosevich, D. Koh, and M. Rudnev, *Averages over hyperplanes, sum-product theory in vector spaces over finite fields and the Erdős-Falconer distance conjecture* Transactions of the American Mathematical Society, 363(06), 3255–3255, (2011).