This is a pre NLP and pre Machine Learning era paper.

NDD detection between discrete data vs continuous data:

#discrete data:

1. Do a simple suppression of the uninteresting pieces of texts
2. Do an exact match on substrings of the remainder

#continuous data:

1. Need sophistication of feature extraction before matching can be done. because image copies or audio copies may have very different representation

==========

Winnow Algorithm:

```
void winnow(int w){
    hash_t h[w];
    for(int i = 0; i < w; ++i)
        h[i] = INT_MAX;
    int r = 0;
    int min = 0;
    ////////////////////////////////////
    while(true){
        r = (r+1)%w;
        h[r] = next_hash();
        if(min == r){
            for(int i=(r-1)%w; i!=r; i=(i-1+w)%w)
```

```
                    if(h[i]<h[min])
                        min = i;
                record(h[min], global_pos(min, r, w));
            }
            else{
                if(h[r] <= h[min]){
                    min = r;
                    record(h[min], global_pos(min, r, w));
                }
            }
        }

}
```

==========

Guaranteed threshold t

Match all substring at least as long as t

noise threshold k

do not detect any matches shorter than the noise threshold, k

consider only k grams to avoid wasting time with noise

size of k:

too large: loose sensitivity to reordering of document contents

too small: invite in noise

==========

Definition 1:

WINNOWING:

In each window select the minimum hash value. If there is more than one hash with the minimum value, select the rightmost occurence. Now save all selected hashes as the fingerprints of the document.

==========

Example winnowing sample text:

a. Sample Text

A do run run run, a do run run

b. Text with irrelevant features removed

adorunrunrunadorunrun

c. The sequence of 5-grams derived from the text

adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun

d. A hypothetical sequence of hashes of the 5-grams (number of hashes = 22-5 = 17)

77 74 42 17 98 50 17 98 8 88 67 39 77 74 42 17 98

e. Windows of hashes of length 4

77 74 42 (17) ---------- 74 42 17 98

42 17 98 50 ---------- 17 98 50 (17)

98 50 17 98 ---------- 50 17 98 (8)

17 98 8 88 ---------- 98 8 88 67

 8 88 67 39 ---------- 88 67 (39) 77

67 39 77 74 ---------- 39 77 74 42

77 74 42 (17) ---------- 74 42 17 98

f. Fingerprints selected by winnowing:

17 17 8 39 17

g. Fingerprints paired with 0-based positional information:

[17,3] [17,6] [8,8] [39,11] [17,15]

==========

EXPECTED DENSITY

DENSITY: density of a fingerprinting algorithm is the expected fraction of fingerprints selected from among all the hash values computed, given random input.

trade off: Guarantee threshold vs Number of fingerprints required

==========

==========

Definition 2 (Local Algorithms):

Let S be a selection function taking a w-tuple of hashes and returning an integer between zero and w−1, inclusive. A fingerprinting algorithm is local with selection function S, if, for every window h_i, ... h_{i+w-1}, the

hash at position i +S (h_i, ... h_{i+w-1}) is selected as a fingerprint.

Definition 3 (Robust Winnowing):

In each window select the minimum hash value. If possible break ties by selecting the same hash as the window one position to the left. If not select the rightmost minimal hash. Save all selected hashes as the fingerprints of the document.

==========