Communication Complexity of Document Exchange

Graham Cormode, Mike Paterson,
Cenk Sahinalp, Uzi Vishkin

Document Exchange

- Two parties — each have a copy of a (huge) file
- The copies differ and there is no record of the changes
- Goal: the parties communicate to exchange their files
- If the files are size n and the "distance" is f, want the communication to be $f \cdot g(n)$
- Aim is to minimize communication, and number of rounds

Prior Work

Correcting *f* Hamming Differences

- Metzner 83, Metzner 91, Barbará & Lipton 91
- Abdel-Ghaffar and Abbadi (1994) communicate $O(f \log n)$ bits [based on Reed-Solomon codes]

Protocols fail if there are more than *f* differences

Edit Distance

Heuristics given by Schwarz, Bowdidge, Burkhard 90

and the simple Rsync utility (Tridgell, Mackerras 96)

No guarantees on performance

Correcting Differences

Correcting the differences is the easy part

(if we have a bound on their number)

- Divide-and-conquer approach to match substrings
  O(f log n log log n) bits for Hamming, edit distances
- Coding approach to send O(f log n) bits for Hamming, edit, block edit distances (Orlitsky 91, developed in CPSV 99)

The hard part is estimating a bound on the distance

Estimating the distance

Given two (binary) strings: x held by A and y held by B, what is the communication cost of estimating:

- Hamming distance $\Sigma_{i=1\ldots n}\,(x_i \neq y_i)$
- Edit distance minimum changes, inserts, deletes, of x into y
- Block edit distances minimum edit and block operations of x into y

For solutions to be interesting, communication cost must be o(n)

Negative results

Obviously, can't give exact answer with probability 1 (since we need $\Omega(n)$ bits just to test for exact equality)

Pang & Gamal (1986): need $\Omega(n)$ bits to estimate Hamming distance with constant probability.

Overcome this by trying to approximate distances:

find an estimate $\hat{d}(x, y)$ so whp $d(x, y) \leq \hat{d}(x, y) \leq c \cdot d(x, y)$

Estimating Hamming distance

Idea: sample a geometrically increasing number of places until differences are noticed. This size used to estimate distance.

Hash each sample to constant size to reduce communication.

Use the sample-XOR technique of Andersson, Miltersen, Riis, Thorup 96 to build a "signature" function
(also used by Kushilevitz, Ostrovsky, Rabani 98 in context of nearest neighbor search)

Pick probability of underestimation = ε. Set $\beta \leq 1 + \frac{\ln \phi}{\ln 1/\varepsilon}$

- For $i = 1 \ldots \log_\beta n$, pick β^i random locations $\boldsymbol{r}_i[1..\beta^i]$ from x
- Build the message $\boldsymbol{m}[1..\log \beta\, n]$ as $m_i(x) = \mathrm{XOR}_{j=1\ldots\beta^i}(x[r_{i,j}])$
-

Estimating Hamming Distance II

- A sends $m(x)$ to B, who computes $m(y)$ using same $\mathbf{r}$
- Compute $m(x)$ XOR $m(y) = 0,0,0,\ldots,0,1,\ldots$
- The first "1" is the first evidence of disagreement
- Let location of first "1"= k
- Estimate of Hamming distance is $\hat{h}(x,y) = n \cdot \dfrac{3(\beta - 1)\ln 1/\varepsilon}{2k}$

The communication cost is $O(\log 1/\varepsilon \cdot \log n)$

There is a single round of communication.

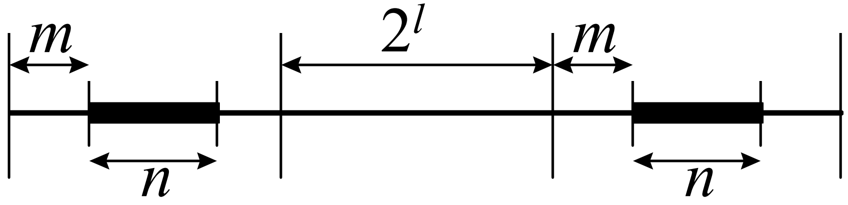
A limited block edit distance

Before estimating general block edit distances, we show how to transform a restricted block edit distance into Hamming distance.

The limited distance of x and y, $ltd(x,y)$ is the minimum number of moves to transform x into y. Permitted moves are:

- change a single bit
- swap "aligned" non-overlapping substrings
- copy a substring over an "aligned" substring as long as there is another aligned copy of the replaced substring

Two substrings of length n are aligned if their locations are $i2^l + m, j2^l + m \quad (n < 2^l)$



Limited Binary Histograms

If x is a string of length 2^k then $LT(x)$ is defined as follows:

For each possible substring z of length 2^i, $LT(x)[z]$ is 1 if z occurs starting at a location $m2^i$ in x ($\forall m$), and 0 otherwise.

Example: $x = 1011$

	0	1	00	01	10	11	
$LT(x)$	1	1	0	0	1	1	…

The histogram is exponentially big but only O(n) entries will be 1
It is never explicitly built, as it is represented by the string x

Transforming limited block edit distance into Hamming distance

Theorem: For strings x, y, length 2^k

$$\tfrac{1}{2}\, ltd(x,y) \leq h(LT(x), LT(y)) < 8k \cdot ltd(x,y)$$

- Upper bound: observe each "limited block" edit operation affects no more than O(k) elements of $LT(x)$

- Lower bound: construct y from x by at most $2h(LT(x), LT(y))$ moves

  Build intermediate strings $x_0, x_1, \ldots x_k$ so x_i has a superset of all length 2^i substrings of y which occur at locations $m2^i$

  Clearly, x_k must be equal to y

Inductive Step

Given x_{i-1} (has all length 2^{i-1} substrings of y occurring at $m2^{i-1}$ $\forall m$), how to build x_i?

- Build the missing length 2^i substrings from left to right
- Copy left and right half of each new substring w into its slot
- Use 2 'credits' from $\mathrm{LT}(x)[w]=\mathrm{LT}(x_i)[w]=0$, $\mathrm{LT}(y)[w]=1$
- If we are copying over the last occurrence of z, pay for this by using 2 'credits' to overcopy the left & right half of z from $\mathrm{LT}(x)[z]=1$, $\mathrm{LT}(y)[z]=0$ ❑

Therefore we can estimate this block edit distance by estimating the Hamming distance of the strings' histograms.

Extending to incorporate edit distance

Key ideas:

- Use a more powerful distance, $LZ(x,y)$
  It allows arbitrary block copies, deletions, as well as the edit distance operations so $LZ(x,y) \leq e(x,y)$

- Base the new histograms, $T(x)$, $T(y)$, on local labels
  Use Locally Consistent Parsing [Sahinalp Vishkin 96] (LCP) to overcome the need for alignment
  Create histogram entries which are 'cores' in LCP

Theorem: $h(T(x), T(y))$ is $O(k^2\, LZ(x,y))$ and $\Omega(LZ(x,y))$

Summary

• Can estimate Hamming distance with high probability

• Can transform edit distance, block edit distance into Hamming distance problems with up to a small poly-logarithmic factor

• Can then run a correction protocol with this estimated distance