6.896 Sublinear Time Algorithms March 6, 2007

Lecture 9

Lecturer: Ronitt Rubinfeld *Scribe: Khanh Do Ba*

1 Lempel-Ziv Compression

1.1 The algorithm

LZ77(w)

$t \leftarrow 1$
repeat
  find longest substring $w_t ... w_{t+\ell-1}$ s.t. $\exists$ index $p < t$ with $w_p ... w_{p+\ell-1} = w_t ... w_{t+\ell-1}$
  if none **then**
    next symbol $= w_t$
    $t \leftarrow t + 1$
  else
    next symbol $= (p, \ell)$
    $t \leftarrow t + \ell$
until $t > n$ $(= |w|)$

1.2 Some notation

The following notation will be used extensively.

$$
\begin{aligned}
n_\ell(w) &= \text{\# compressed segments of length } \ell \text{ in } w\text{, not counting alphabet symbols nor last compressed segment} \\
C_{LZ}(w) &= \text{\# symbols in compressed string (\# pairs + \# alphabet symbols} \\
d_\ell(w) &= \text{\# distinct substrings of length } \ell
\end{aligned}
$$

Examples:

1. $aaaaaaa$ is encoded as $a(1,6)$ and has $d_\ell(w) = 1 \ \forall \ell \in [7]$.
2. $abcd$ is encoded as $abcd$.
3. $abaabaaabaaa$ is encoded as $aa(1,1)(1,4)(4,5)$, where the compressed segments are broken up as $a, b, a, abaa, abaaa$.
4. $abcaabbccaaabbbccc$ is encoded as $abc(1,1)(1,2)(3,2)(3,3)(5,3)(7,3)(3,1)$. The compressed segments are broken up as $a, b, c, a, ab, bc, caa, abb, bcc, c$, and

$$
\begin{aligned}
d_1(w) &= 3 \\
d_2(w) &= 7 \\
d_3(w) &= 12 \\
d_4(w) &= 13 \\
n_1(w) &= 2 \\
n_2(w) &= 2 \\
n_3(w) &= 3 \\
n_{\ell>3}(w) &= 0.
\end{aligned}
$$

2 Estimating LZ Compressibility

2.1 Main theorem

The key idea is the following theorem.

Theorem 1 *For any integer $\ell_0 > 1$, let*

$$m = m(\ell_0) = \max_{\ell \in [\ell_0]} \frac{d_\ell(w)}{\ell}.$$

Then

$$m \leq C_{LZ}(w) \leq 4\left(m \log \ell_0 + \frac{n}{\ell_0}\right).$$

Given this theorem, to approximate $C_{LZ}(w)$ we need to compute $2m\sqrt{\log \ell_0}$ + (something), where (something) accounts for the n/ℓ_0 term. We cannot get a purely multiplicative approximation, so we will have to settle with a combination of multiplicative and additive errors.

2.2 The algorithm

We assume the algorithm EstimateColors$(w, \ell, \gamma, \delta)$, where w is a string, ℓ is a substring length, γ is a multiplicative approximation parameter, and δ is a confidence parameter. EstimateColors outputs, with probability $\geq 1 - \delta$, a γ-multiplicative estimate for $d_\ell(w)$, using $O\left(\frac{n}{\gamma^2} \ell \log \frac{1}{\delta}\right)$ queries. We essentially saw how to construct this subroutine in the last lecture (although instead of querying a random location in w, here we will pick a random start index i and look at the length ℓ word beginning at location i).

LZ-Approx(w, A, ε)

$\ell_0 \leftarrow 2/A\varepsilon$, $B \leftarrow A/2\sqrt{\log \ell_0}$
for each $\ell \leq \ell_0$ **do**
    $\hat{d}_\ell \leftarrow$ EstimateColors$(w, \ell, B, 1/3\ell_0)$
$\hat{m} \leftarrow \max_\ell(\hat{d}_\ell/\ell)$
output $\hat{C}_{LZ} = 2\hat{m}\sqrt{\log \ell_0} + \varepsilon n$

2.3 Algorithm behavior

Using the union bound, we can say that

$$\Pr[\text{all } \hat{d}_\ell\text{'s are good}] \geq 1 - \frac{\ell_0}{3\ell_0} \geq \frac{2}{3}.$$

Now, if each $\hat{d}_\ell$ is a good B-multiplicative estimate, then $\hat{m}$ is as well, giving us

$$\frac{\hat{m}}{B} \leq m \leq C_{LZ} \leq 4\left(m \log \ell_0 + \frac{n}{\ell_0}\right) \leq 4\left(B\hat{m} \log \ell_0 + \frac{n}{\ell_0}\right).$$

Rearranging the first, third and fifth term, we get

$$\frac{A}{B} \cdot \frac{C_{LZ} - 4n/\ell_0}{4B \log \ell_0} + \varepsilon n \leq \frac{A}{B} \cdot \hat{m} + \varepsilon n \leq \frac{A}{B} \cdot B \cdot C_{LZ} + \varepsilon n = A \cdot C_{LZ} + \varepsilon n.$$

Plugging in $A/B = 2\sqrt{\log \ell_0}$ and $4n/\ell_0 = 2nA\varepsilon$, we get

$$(2\sqrt{\log \ell_0}) \cdot \frac{C_{LZ} - 2nA\varepsilon}{\frac{4A}{2\sqrt{\log \ell_0}} \cdot \log \ell_0} = \frac{C_{LZ}}{A} - \varepsilon n \leq 2\hat{m}\sqrt{\log \ell_0} + \varepsilon n \leq A \cdot C_{LZ} + \varepsilon n,$$

which gives us our A-multiplicative, εn-additive approximation (with probability $\geq 2/3$).

2.4 Query complexity

Each call to EstimateColors uses $O\left(\frac{n}{B^2}\ell_0 \log 3\ell_0\right)$ queries, so overall we use $O\left(\frac{n}{B^2}\ell_0^2 \log 3\ell_0\right)$ queries. Observe, however, that the calls to EstimateColors can reuse queries: their respective probabilities of success would no longer be independent, but this is not a problem as we only use the union bound. Thus our query complexity works out to $\tilde{O}\left(\frac{n}{A^3\varepsilon}\right)$. For instance, with $\varepsilon = n^{-1/2}$ and $A = n^{1/4}$, we use $O(n^{3/4})$ queries to get a $n^{1/4}$-multiplicative, $n^{1/2}$-additive estimate.

2.5 Proof of theorem

It remains to prove our main theorem. We do this via two lemmas, each of which implies one of the inequalities in our theorem.

Lemma 2 (first inequality) *For every $\ell \in [\ell_0]$, $d_\ell(w) \leq C_{LZ}(w) \cdot \ell$.*

Proof
($\ell = 1$):

$$d_1(w) = \#\text{ distinct alphabet characters} \leq C_{LZ}(w)$$

since when LZ77 sees a new alphabet character it simply outputs it.
($\ell > 1$):

We scan from left to right and count the number of new length-ℓ substrings. Observe that if a substring is entirely contained within some compressed segment, it cannot be new! Hence, any new substring must start in one compressed segment and end in another (i.e., it must straddle a boundary). But before each boundary, there are only $\ell - 1$ possible starting positions that would allow a length-ℓ substring to straddle the boundary, and there are only $C_{LZ}(w) - 1$ boundaries, so

$$d_\ell(w) \leq (C_{LZ}(w) - 1)(\ell - 1) \leq C_{LZ}(w) \cdot \ell,$$

proving our lemma. ■

Lemma 3 (second inequality) *Fix any $\ell_0 \in [n]$. If m is such that for every $\ell \in [\ell_0]$, $d_\ell(w) \leq m\ell$, then*

$$C_{LZ}(w) \leq 4\left(m \log \ell_0 + \frac{n}{\ell_0}\right).$$

Proof Let $n_\ell = n_\ell(w)$ and $d_\ell = d_\ell(w)$. We will show two things:

1. For every $\ell \in \{1, \ldots, \ell_0/2\}$ (wlog, assume ℓ_0 is even),
$$\sum_{k=1}^{\ell} n_k \leq 2(m+1) \sum_{k=1}^{\ell} \frac{1}{k}.$$
2. For every $\ell \geq 1$,
$$\sum_{k=\ell+1}^{n} n_k \leq \frac{n}{\ell+1}.$$

The proof of 2 is easy:

$$(\ell+1) \sum_{k=\ell+1}^{n} n_k \leq \sum_{k=\ell+1}^{n} k n_k \leq n.$$

Where the last inequality follows since the compressed segments are disjoint.

The proof of 1 we defer until next lecture. Now, we put them together with $\ell = \ell_0/2$ to get

$$\begin{aligned}
C_{LZ}(w) &= \sum_{k=1}^{n} n_k + 1 \\
&\leq 2(m+1)\sum_{k=1}^{\ell_0/2} \frac{1}{k} + \frac{n}{\ell_0/2+1} + 1 \\
&\leq 2(m+1)(\ln \ell_0 + 1) + \frac{2n}{\ell_0} \\
&\leq 4\left(m \log \ell_0 + \frac{n}{\ell_0}\right),
\end{aligned}$$

completing the proof of our lemma and hence our theorem (modulo the proof of 1). ■