Indexing Weighted-Sequences

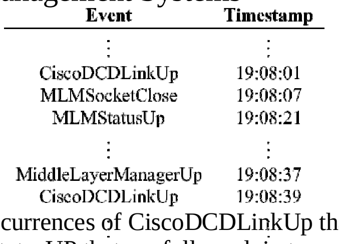
Presented by L. L. Cheng

Contents

- Motivation
- Definitions
- Iso-Depth Index
- Experiment Results
- Conclusion

Motivation

- Event Management Systems



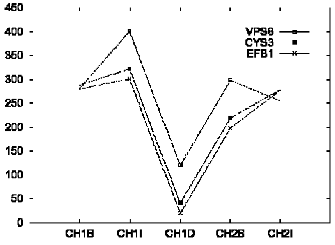
Event	Timestamp
⋮	⋮
CiscoDCDLinkUp	19:08:01
MLMSocketClose	19:08:07
MLMStatusUp	19:08:21
⋮	⋮
MiddleLayerManagerUp	19:08:37
CiscoDCDLinkUp	19:08:39
⋮	⋮

- Find all occurrences of CiscoDCDLinkUp that are followed by MLMStatusUP that are followed, in turn, by CiscoDCDLinkUp, under the condition that the interval between the first two events is about 20±2 seconds, and the interval between the 1st and 3rd events is about 40±3 seconds

Motivation

- DNA Micro-array Analysis
  - *Find all genes whose expression level in sample CH1I is about 100±5 units higher than that in CH2B, 280±10 units higher than that in CH1D, and 75±7 units higher than that in CH2I.*

```
       CH1B    CH1I    CH1D    CH2B    CH2I...
gene_a 90      110     190     210     290...
gene_b 190     210     390     410     590...
gene_c 90      110     110     90      120...
gene_d 200     100     400     90      600...
........
```

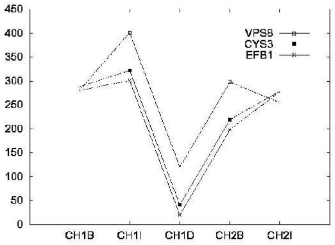


Motivation

- DNA Micro-array Analysis
  - Select * FROM dna-array WHERE 95≤(CH1I - CH2B)≤105 AND 270≤(CH1I - CH1D)≤290 AND 68≤(CH1I CH2I)≤82

```
Select * FROM dna-array
WHERE 95≤(CH1I - CH2B)≤105
AND 270≤(CH1I - CH1D)≤290
AND 68≤(CH1I   CH2I)≤82
```

	CH1B	CH1I	CH1D	CH2B	CH2I...
gene_a	90	110	190	210	290...
gene_b	190	210	390	410	590...
gene_c	90	110	110	90	120...
gene_d	200	100	400	90	600...
....					



Definition: Weighted-Sequence

- a sequence of (symbol, weight) pairs:

$$T=<(a_1,w_1), (a_2,w_2), ..., (a_n,w_n)>$$

- a_i is a symbol
- w_i is a real number
- E.g.

  <(CH1I, 401), (CH1B, 281), (CH1D, 120).....>
- However, it ONLY considers weights are in ascending order in the paper. ($w_i \leq w_{i+1}$)

  < (CH1D, 120), (CH1B, 281), (CH1I, 401), ...>

Definition: Notations

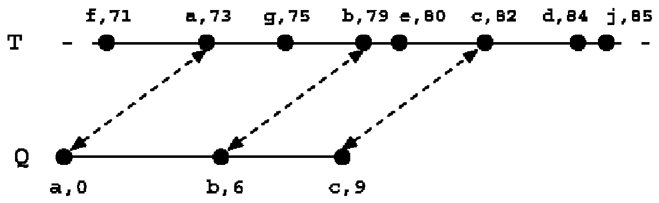
- T: a weighted sequence
  - e.g. T = <(a, 3), (c, 7), (h, 11), (d, 22)>
- T_i: the *i*-th item in T
  - e.g. T_2 = (c, 7)
- $s(T_i)$: the symbol of *i*-th item
  - e.g. $s(T_2)$ = c
- $w(T_i)$: the weight of *i*-th item
  - e.g. $w(T_2)$ = 7
- A: symbol set, i.e. $A = \cup_i \{s(T_i)\}$

Definition: Notations (con't)

- Let T = <(a, 3), (c, 7), (h, 11), (d, 22)>
- |T|: length (number of items) of T
  - |T| = 4
- ||T||: the range of T, $||T||=w(T_{|T|}) - w(T_1)$
  - ||T|| = 22-3 =19
- T' ⊂T: T' is a (non-contiguous) subsequence of T
  - e.g. T'=<(c, 7), (d, 22)>
- ξ: window size
- *Q*: a query sequence which is a weighted-sequence in form of $<(b_1, 0), (b_2, w_2),..., (b_m, w_m)>$

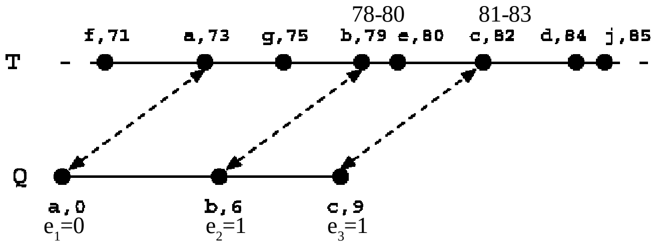
Weighted-Sequence Matching

- A query sequence Q matches sequence T if there exists a (non-contiguous) subsequence $T' \subset T$ such that $|Q| = |T'|$, $s(Q_i)=s(T'_i)$, and $w(Q_i) = w(T'_i)-w(T'_1)$, $\forall i \in 1,...,|Q|$.



Approximate Matching of Weighted Sequences

- Given A query sequence Q and tolerance $e_i \geqslant 0$, $i \in 1,...,|Q|$. Q approximately matches sequence T if there exists a (non-contiguous) subsequence $T' \subset T$ such that $|Q| = |T'|$, $s(Q_i)=s(T'_i)$, and $|w(Q_i) - (w(T'_i) - w(T'_1))| \leqslant e_i$, $\forall i \in 1,...,|Q|$.



The Iso-Depth Index – Overview

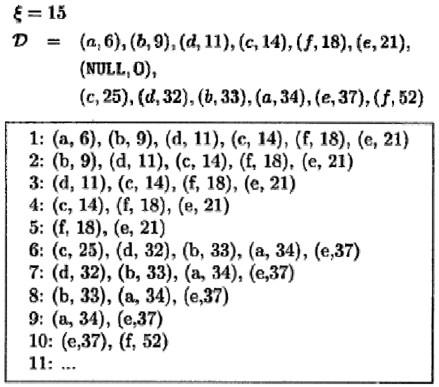
- It supports fast accesses of (non-contiguous) subsequences that match a query sequence.
- We need to discretize the weights in sequences into a number of equi-width units.
- It embodies a compact index to all the distinct, non-empty sequences whose weight range is less than ξ, the window size provided by the user.
- Queries are constrained by the ξ.

Index Building (1)

- In the database, it contains certain number of weighted sequences.
- Concatenate all the sequences together and the sequences are separated by (NULL, 0).
- <(a, 6), (b, 9), (d, 11), (c, 14), (f, 18), (e, 21), (NULL, 0), (c, 25), (d, 32), (b, 33), (a, 34), (e, 37), (f, 52)>
- We let the concatenated weighted sequence be D.

Index Building (2)

- We find all the continuous subsequences T in D such that ||T|| ⩽ ξ and |T|>1 (without across the boundary)



$\xi = 15$

$\mathcal{D} = (a,6),(b,9),(d,11),(c,14),(f,18),(e,21),$
$(\mathrm{NULL},0),$
$(c,25),(d,32),(b,33),(a,34),(e,37),(f,52)$

1: (a, 6), (b, 9), (d, 11), (c, 14), (f, 18), (e, 21)
2: (b, 9), (d, 11), (c, 14), (f, 18), (e, 21)
3: (d, 11), (c, 14), (f, 18), (e, 21)
4: (c, 14), (f, 18), (e, 21)
5: (f, 18), (e, 21)
6: (c, 25), (d, 32), (b, 33), (a, 34), (e,37)
7: (d, 32), (b, 33), (a, 34), (e,37)
8: (b, 33), (a, 34), (e,37)
9: (a, 34), (e,37)
10: (e,37), (f, 52)
11: ...

Index Building (3)

- We transform each subsequence, T, in to one dimensional sequence *S*.

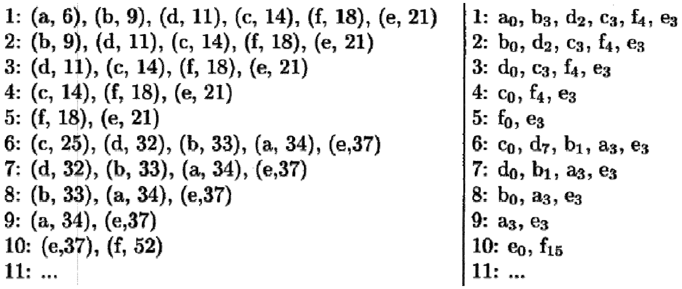
$$f(\langle \mathcal{T}_1, \cdots, \mathcal{T}_k \rangle) = \langle \mathcal{S}_1, \cdots, \mathcal{S}_k \rangle$$

where:

$$\mathcal{S}_i = \begin{cases} s(\mathcal{T}_i)_0 & : \quad i = 1 \\ s(\mathcal{T}_i)_{w(\mathcal{T}_i)-w(\mathcal{T}_{i-1})} & : \quad i > 1 \end{cases}$$

Index Building (3)

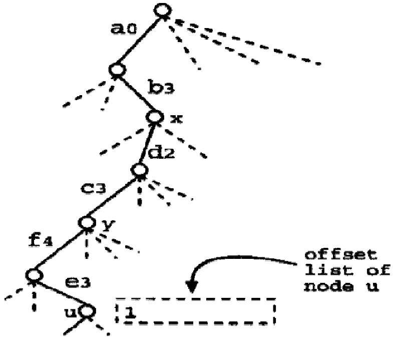
- We transform each subsequence, T, in to one dimensional sequence *S*.



1: (a, 6), (b, 9), (d, 11), (c, 14), (f, 18), (e, 21)	1: a_0, b_3, d_2, c_3, f_4, e_3
2: (b, 9), (d, 11), (c, 14), (f, 18), (e, 21)	2: b_0, d_2, c_3, f_4, e_3
3: (d, 11), (c, 14), (f, 18), (e, 21)	3: d_0, c_3, f_4, e_3
4: (c, 14), (f, 18), (e, 21)	4: c_0, f_4, e_3
5: (f, 18), (e, 21)	5: f_0, e_3
6: (c, 25), (d, 32), (b, 33), (a, 34), (e,37)	6: c_0, d_7, b_1, a_3, e_3
7: (d, 32), (b, 33), (a, 34), (e,37)	7: d_0, b_1, a_3, e_3
8: (b, 33), (a, 34), (e,37)	8: b_0, a_3, e_3
9: (a, 34), (e,37)	9: a_3, e_3
10: (e,37), (f, 52)	10: e_0, f_{15}
11: ...	11: ...

Index Building (4)

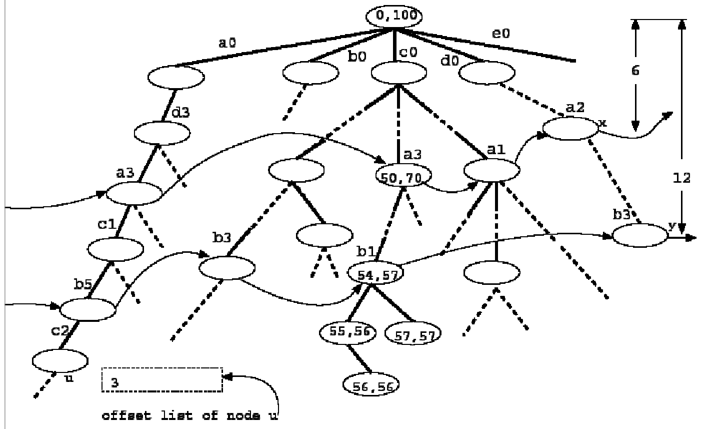
- We insert all the transformed sequences found in previous steps into a trie.
- Each node in the trie have an offset list which store all the window offsets associated with the node.



Index Building (5)

- we assign IDs to each node in the trie in DFS traversal order (starting with 0 for the root node)
- for each node, we also record the largest ID of its descendants
- So, each node is assigned a pair of labels, (v_s, v_m). v_s is the ID of the node and v_m is the largest ID of the node's descendant nodes.

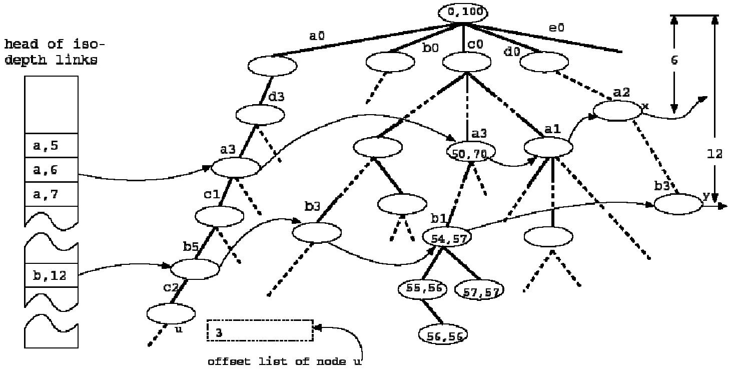
Index Building (5)



Index Building (6)

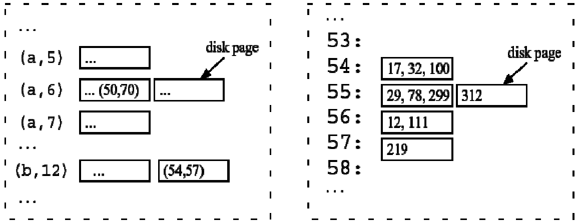
- We create iso-depth links for each (x,d) pair, where x is the symbol, and d is the depth of the node, d = 1, ..., ξ.
- The depth of node v is the distance between the root and v. i.e. summing up the subscripts of the symbols from root to v.
- Nodes in an iso-depth link are sorted by their IDs in ascending order.

Index Building (6)



Index Building (7)

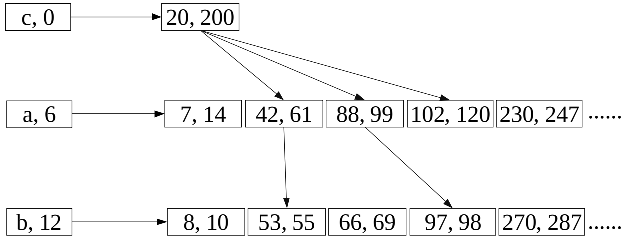
- In reality, each iso-depth link stores the (v_s, v_m) pairs in secondary memory.
- The offset lists are also stored in secondary memory.
- The trie is only for index construction and will be throw away after construction.



PART I: disk pages of iso-depth arrays PART II:disk pages of offset lists

Exact Matching

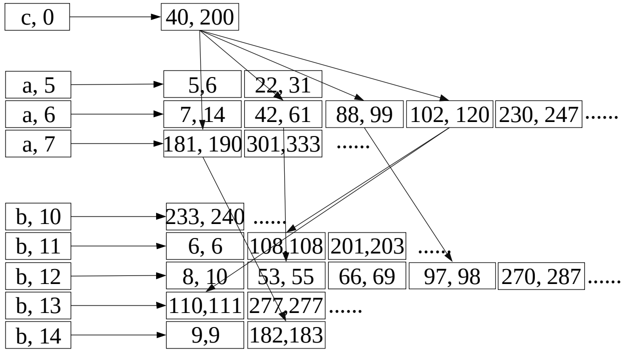
- *Q*=<(c, 0), (a, 6), (b, 12)>



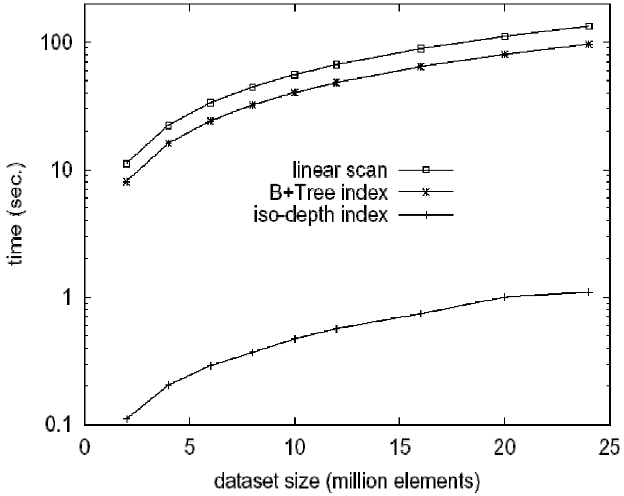
Finally, we find the offset lists of nodes 53, 54, 55, 97 and 98. Those are the offsets in the data sequence where subsequence *Q* occur

Approximate Matching

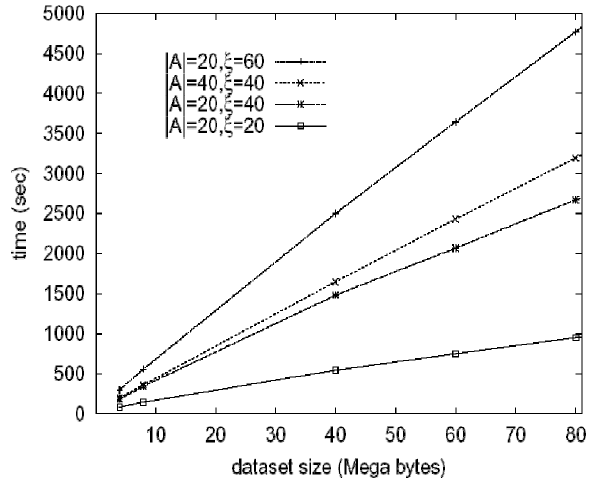
- *Q*=<(c, 0), (a, 6), (b, 12)>, e_1=0, e_2=1, e_3=2.



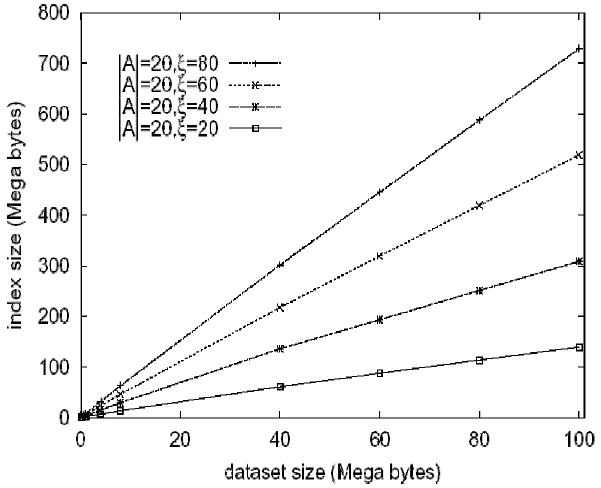
Experiment: Query Time(D?-A200-U-U100)



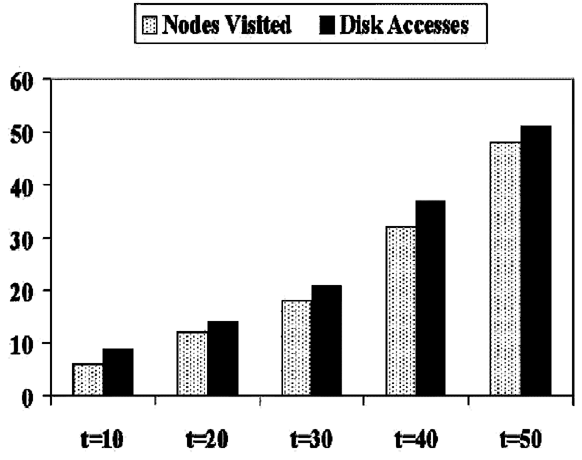
Experiment: Index Building Time



Experiment: Index Size



Experiment: Query Form <(x,0),(y,t),(z,59)>; D5000K-A100-U-U10



Experiment: Microarray Data

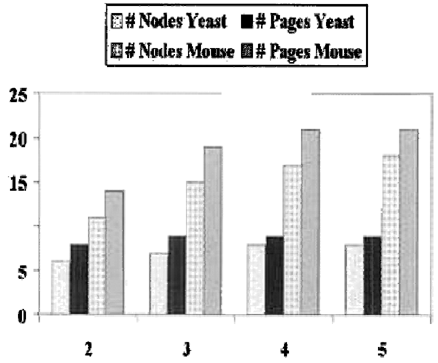


Figure 9. Random queries against DNA micro-arrays of Yeast and Mouse gene expression, with varying number of elements in the query from 2 to 5.

Limitations

- The tolerances given in a query must not disturb the order of the elements in the sequence.
  - $w(Q_i) + e_i < w(Q_{i+1}) - e_{i+1}$
  - Q=<(b, 0), (a, 5), (c, 6)>, e_1=0, e_2=1, e_3=1 **Invalid!!**
- $||Q|| \leqslant \xi$
- Indexing building requires to build a trie of all the subsequences within the window size. The trie can be too big and cannot fit into main memory. This may be the bottleneck.

Conclusion and Discussion

- Some data, e.g. timestamped event sequence, microarray data, can be map to the weighted-sequence model.
- Iso-depth indexing structure can support searching of weighted-sequence efficiently.
- However, there are still some limitations.

References

- *Indexing Weighted Sequences in Large Databases*, by Haixun Wang, Chang-Shing Perng, Wei Fan, Sanghyun Park, Philip S Yu, in IEEE International Conference on Data Engineering (ICDE) 2003, Bangalore, India.
- *An Indexing Structure for Similarity Searching in Microarray Data*, by Haixun Wang, Charles Perng, Wei Fan, and Philip S. Yu, in Proceedings of the First IEEE Computer Society Bioinformatics Conference (CSB 2002), August, 2002, Palo Alto, California, USA.