


**15-826: Multimedia Databases
and Data Mining**


Lecture #15: Text - part II
C. Faloutsos



Must-read Material

- Textbook, Chapter 6

15-826 Copyright: C. Faloutsos (2009) 2



Optional (but terrific to read)

- ★ • McIlroy, M. D. (Jan. 1982). "Development of a Spelling List." IEEE Trans. on Communications COM-30(1): 91-99.
- ★ • Severance, D. G. and G. M. Lohman (Sept. 1976). "Differential Files: Their Application to the Maintenance of Large Databases." ACM TODS 1(3): 256-267.

15-826 Copyright: C. Faloutsos (2009) 3

CMU SCS

Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- ➔ • Indexing - similarity search
- Data Mining

15-826 Copyright: C. Faloutsos (2009) 4

CMU SCS

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- ➔ • text
- multimedia
- ...

15-826 Copyright: C. Faloutsos (2009) 5

CMU SCS

Text - Detailed outline

- text
 - problem
 - full text scanning
 - ➔ - inversion
 - signature files
 - clustering
 - information filtering and LSI

15-826 Copyright: C. Faloutsos (2009) 6

CMU SCS

Text - Inversion

15-826 Copyright: C. Faloutsos (2009) 7

CMU SCS

Text - Inversion

Q: space overhead?

15-826 Copyright: C. Faloutsos (2009) 8

CMU SCS

Text - Inversion

A: mainly, the postings lists

15-826 Copyright: C. Faloutsos (2009) 9

CMU SCS

Text - Inversion

- how to organize dictionary?
- stemming – Y/N?
- insertions?

15-826 Copyright: C. Faloutsos (2009) 10

CMU SCS

Text - Inversion

- how to organize dictionary?
 - B-tree, hashing, TRIEs, PATRICIA trees, ...
- stemming – Y/N?
- insertions?

15-826 Copyright: C. Faloutsos (2009) 11

CMU SCS

Text - Inversion

- variations:
- Parallelism [Tomasia+,93]
- Insertions [Tomasia+94], [Brown+]
 - ‘zipf’ distributions
- Approximate searching (‘glimpse’ [Wu+])

15-826 Copyright: C. Faloutsos (2009) 12

CMU SCS

Text - Inversion

- postings list – more Zipf distr.: eg., rank-frequency plot of ‘Bible’

log(freq)

log(rank)

freq ~ 1/rank / ln(1.78V)

15-826 Copyright: C. Faloutsos (2009) 13

CMU SCS

Text - Inversion

- postings lists
 - Cutting+Pedersen
 - (keep first 4 in B-tree leaves)
 - how to allocate space: [Faloutsos+92]
 - geometric progression
 - compression (Elias codes) [Zobel+] – down to 2% overhead!
 - Compression and doc reordering [Blandford+2002]

15-826 Copyright: C. Faloutsos (2009) 14

CMU SCS

Conclusions

- Conclusions: needs space overhead (2%-300%), but it is the fastest

15-826 Copyright: C. Faloutsos (2009) 15

CMU SCS

Text - Detailed outline

- text
 - problem
 - full text scanning
 - inversion
 - ➔ – signature files
 - clustering
 - information filtering and LSI

15-826 Copyright: C. Faloutsos (2009) 16

CMU SCS

Signature files

- idea: 'quick & dirty' filter

Signature File

Text File

15-826 Copyright: C. Faloutsos (2009) 17

CMU SCS

Signature files

- idea: 'quick & dirty' filter
- then, do seq. scan on sign. file and discard 'false alarms'
- Adv.: easy insertions; faster than seq. scan
- Disadv.: $O(N)$ search (with small constant)
- Q: how to extract signatures?

15-826 Copyright: C. Faloutsos (2009) 18

CMU SCS

Signature files

- A: superimposed coding!! [Mooers49], ...

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

m (=4 bits/word)
F (=12 bits sign. size)

15-826 Copyright: C. Faloutsos (2009) 19

CMU SCS

Signature files

- A: superimposed coding!! [Mooers49], ...

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

data ↑ ↑↑ ↑

actual match

15-826 Copyright: C. Faloutsos (2009) 20

CMU SCS

Signature files

- A: superimposed coding!! [Mooers49], ...

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

retrieval ↑ ↑ ↑↑

actual dismissal

15-826 Copyright: C. Faloutsos (2009) 21

CMU SCS

Signature files

- A: superimposed coding!! [Mooers49], ...

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

nucleotic ↑ ↑ ↑ ↑

false alarm ('false drop')

15-826 Copyright: C. Faloutsos (2009) 22

CMU SCS

Signature files

- A: superimposed coding!! [Mooers49], ...

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

'YES' is 'MAYBE'
'NO' is 'NO'

15-826 Copyright: C. Faloutsos (2009) 23

CMU SCS

Signature files

- Q1: How to choose F and m ?
- Q2: Why is it called 'false drop'?
- Q3: other apps of signature files?

15-826 Copyright: C. Faloutsos (2009) 24

CMU SCS

Signature files

- Q1: How to choose F and m ?

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

m (=4 bits/word)
F (=12 bits sign. size)

15-826 Copyright: C. Faloutsos (2009) 25

CMU SCS

Signature files

- Q1: How to choose F and m ?
- A: so that doc. signature is 50% full

Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011

m (=4 bits/word)
F (=12 bits sign. size)

15-826 Copyright: C. Faloutsos (2009) 26

CMU SCS

Signature files

- Q1: How to choose F and m ?
- ➔ Q2: Why is it called 'false drop'?
- Q3: other apps of signature files?

15-826 Copyright: C. Faloutsos (2009) 27

CMU SCS

Signature files

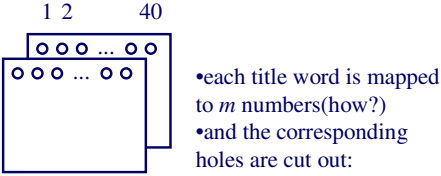
- Q2: Why is it called ‘false drop’?
- Old, but fascinating story [1949]
 - how to find qualifying books (by title word, and/or author, and/or keyword)
 - in $O(1)$ time?
 - *without computers*

15-826 Copyright: C. Faloutsos (2009) 28

CMU SCS

Signature files

- Solution: edge-notched cards



1 2 40

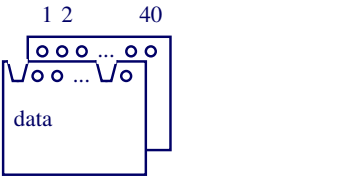
- each title word is mapped to m numbers(how?)
- and the corresponding holes are cut out:

15-826 Copyright: C. Faloutsos (2009) 29

CMU SCS

Signature files

- Solution: edge-notched cards



1 2 40

data

‘data’ -> #1, #39

15-826 Copyright: C. Faloutsos (2009) 30

CMU SCS

Signature files

- Search, e.g., for 'data': activate needle #1, #39, and shake the stack of cards!

'data' -> #1, #39

15-826 Copyright: C. Faloutsos (2009) 31

CMU SCS

Signature files

- Also known as 'zatocoding', from 'Zator' company.

15-826 Copyright: C. Faloutsos (2009) 32

CMU SCS

Signature files

- Q1: How to choose F and m ?
- Q2: Why is it called 'false drop'?
- ➔ Q3: other apps of signature files?

15-826 Copyright: C. Faloutsos (2009) 33

CMU SCS

Signature files

- Q3: other apps of signature files?
- A: anything that has to do with ‘membership testing’: does ‘data’ belong to the set of words of the document?


Word	Signature
data	001 000 110 010
base	000 010 101 001
doc.signature	001 010 111 011


15-826 Copyright: C. Faloutsos (2009) 34


CMU SCS

Signature files

- UNIX’s early ‘spell’ system [McIlroy]
- Bloom-joins in System R* [Mackert+] and ‘active disks’ [Riedel99]
- differential files [Severance+Lohman]


 McIlroy


 Riedel


 Lohman

15-826 Copyright: C. Faloutsos (2009) 35

CMU SCS

App#1: Unix’s spell

Dictionary

~30,000 words

aaron
apple
...
zoo

← ?

electroencephalogram

What to do if the dictionary does not fit in memory (~1980)?

15-826 Copyright: C. Faloutsos (2009) 36

CMU SCS

App#1: Unix's spell

A: allow for a few typos! And use

- huge bit string (2^{*27})
- Hash each dictionary word to a bit
- Compress the string

15-826 Copyright: C. Faloutsos (2009) 37

CMU SCS

App#1: Unix's spell

Sub-questions:

- Q1: How often do we allow typos?
- Q2: Will the (compressed) bit string fit in memory?
- Q3: How to compress the bit string?

15-826 Copyright: C. Faloutsos (2009) 38

CMU SCS

App#2: Bloom-joins

R @Chicago

A	B	C
1		
1		
3		
...		
1		
12		

S @NY

A	E	F
1		
18		
1		
...		
23		
2		

R join S (@PIT)

15-826 Copyright: C. Faloutsos (2009) 39

CMU SCS

App#2: Bloom-joins

R @Chicago

A	B	C
1		
1		
3		
...		
1		
12		

S @NY

A	E	F
1		
18		
1		
...		
23		
2		

Idea: reduce transmission cost: 'R semijoin S'

15-826 Copyright: C. Faloutsos (2009) 40

CMU SCS

App#2: Bloom-joins

Idea: reduce transmission cost: 'R semijoin S'

That is,

- 'S' ships its unique values of 'A'
- 'R' deletes non-matching tuples
- (and they both send their tuples to PIT)

Q: what if we want to send at most, say 100 bytes NY -> Chicago?

15-826 Copyright: C. Faloutsos (2009) 41

CMU SCS

App#2: Bloom-joins

Q: what if we want to send at most, say 100 bytes NY -> Chicago?

A: Bloom-join! Send a bloom filter of the S.A values

15-826 Copyright: C. Faloutsos (2009) 42

CMU SCS

App#3: Differential files

Problem definition:

- A large file (eg., with EMPLOYEE records), nicely packed and organized (eg., B-tree)
- A few insertions/deletions, that we would like to keep separate, and merge, at night
- How to search, eg., for employee #123?

15-826 Copyright: C. Faloutsos (2009) 43

CMU SCS

App#3: Differential files

ssn	name	...
1		
5		
12		
...		
503		
509		

flag	ssn	name, ..
i	123	
i	55	
d	17	
d	33	

Differential file

15-826 Copyright: C. Faloutsos (2009) 44

CMU SCS

App#3: Differential files

- Q: How to search, eg., for employee #123?
- A: bloom-filter, for keys of diff. file

- Q: What are the advantages of differential files?
- A: <see paper, for 10(!) of them>

15-826 Copyright: C. Faloutsos (2009) 45

CMU SCS

Signature files - conclusions

- easy insertions; slower than inversion
- brilliant idea of ‘quick and dirty’ filter: quickly discard the vast majority of non-qualifying elements, and focus on the rest.

15-826 Copyright: C. Faloutsos (2009) 46

CMU SCS

References

- Aho, A. V. and M. J. Corasick (June 1975). "Fast Pattern Matching: An Aid to Bibliographic Search." CACM 18(6): 333-340.
- Boyer, R. S. and J. S. Moore (Oct. 1977). "A Fast String Searching Algorithm." CACM 20(10): 762-772.


15-826 Copyright: C. Faloutsos (2009) 47

CMU SCS

References

- Blandford, D. and Blelloch, G. 2002. *Index Compression through Document Reordering*. Data Compression Conference (DCC '02) (April 02 - 04, 2002).
- Brown, E. W., J. P. Callan, et al. (March 1994). *Supporting Full-Text Information Retrieval with a Persistent Object Store*. EDBT conference, Cambridge, U.K., Springer Verlag.


15-826 Copyright: C. Faloutsos (2009) 48

 CMU SCS

References - cont'd

- Faloutsos, C. and H. V. Jagadish (Aug. 23-27, 1992). On B-tree Indices for Skewed Distributions. 18th VLDB Conference, Vancouver, British Columbia.
- Karp, R. M. and M. O. Rabin (March 1987). "Efficient Randomized Pattern-Matching Algorithms." IBM Journal of Research and Development 31(2): 249-260.
- Knuth, D. E., J. H. Morris, et al. (June 1977). "Fast Pattern Matching in Strings." SIAM J. Comput 6(2): 323-350.


15-826 Copyright: C. Faloutsos (2009) 49

 CMU SCS

References - cont'd

- Mackert, L. M. and G. M. Lohman (August 1986). R* Optimizer Validation and Performance Evaluation for Distributed Queries. Proc. of 12th Int. Conf. on Very Large Data Bases (VLDB), Kyoto, Japan.
- Manber, U. and S. Wu (1994). GLIMPSE: A Tool to Search Through Entire File Systems. Proc. of USENIX Techn. Conf.
- ★ McIlroy, M. D. (Jan. 1982). "Development of a Spelling List." IEEE Trans. on Communications COM-30(1): 91-99.


15-826 Copyright: C. Faloutsos (2009) 50

 CMU SCS

References - cont'd


- Mooers, C. (1949). Application of Random Codes to the Gathering of Statistical Information
- Bulletin 31. Cambridge, Mass, Zator Co.
- Pedersen, D. C. a. J. (1990). Optimizations for dynamic inverted index maintenance. ACM SIGIR.
- Riedel, E. (1999). Active Disks: Remote Execution for Network Attached Storage. ECE, CMU. Pittsburgh, PA.

15-826 Copyright: C. Faloutsos (2009) 51

 **References - cont'd**

- ★ Severance, D. G. and G. M. Lohman (Sept. 1976). "Differential Files: Their Application to the Maintenance of Large Databases." ACM TODS 1(3): 256-267.
- Tomasic, A. and H. Garcia-Molina (1993). Performance of Inverted Indices in Distributed Text Document Retrieval Systems. PDIS.
- Tomasic, A., H. Garcia-Molina, et al. (May 24-27, 1994). Incremental Updates of Inverted Lists for Text Document Retrieval. ACM SIGMOD, Minneapolis, MN.

15-826 Copyright: C. Faloutsos (2009) 52

 **References - cont'd**

- Wu, S. and U. Manber (1992). "AGREP- A Fast Approximate Pattern-Matching Tool." .
- Zobel, J., A. Moffat, et al. (Aug. 23-27, 1992). An Efficient Indexing Technique for Full-Text Database Systems. VLDB, Vancouver, B.C., Canada.

15-826 Copyright: C. Faloutsos (2009) 53
