QuickScorer: a fast algorithm to rank documents with additive ensembles of regression trees

Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto

HPC Lab, ISTI-CNR, Pisa, Italy & Tiscali SpA

Salvatore Orlando

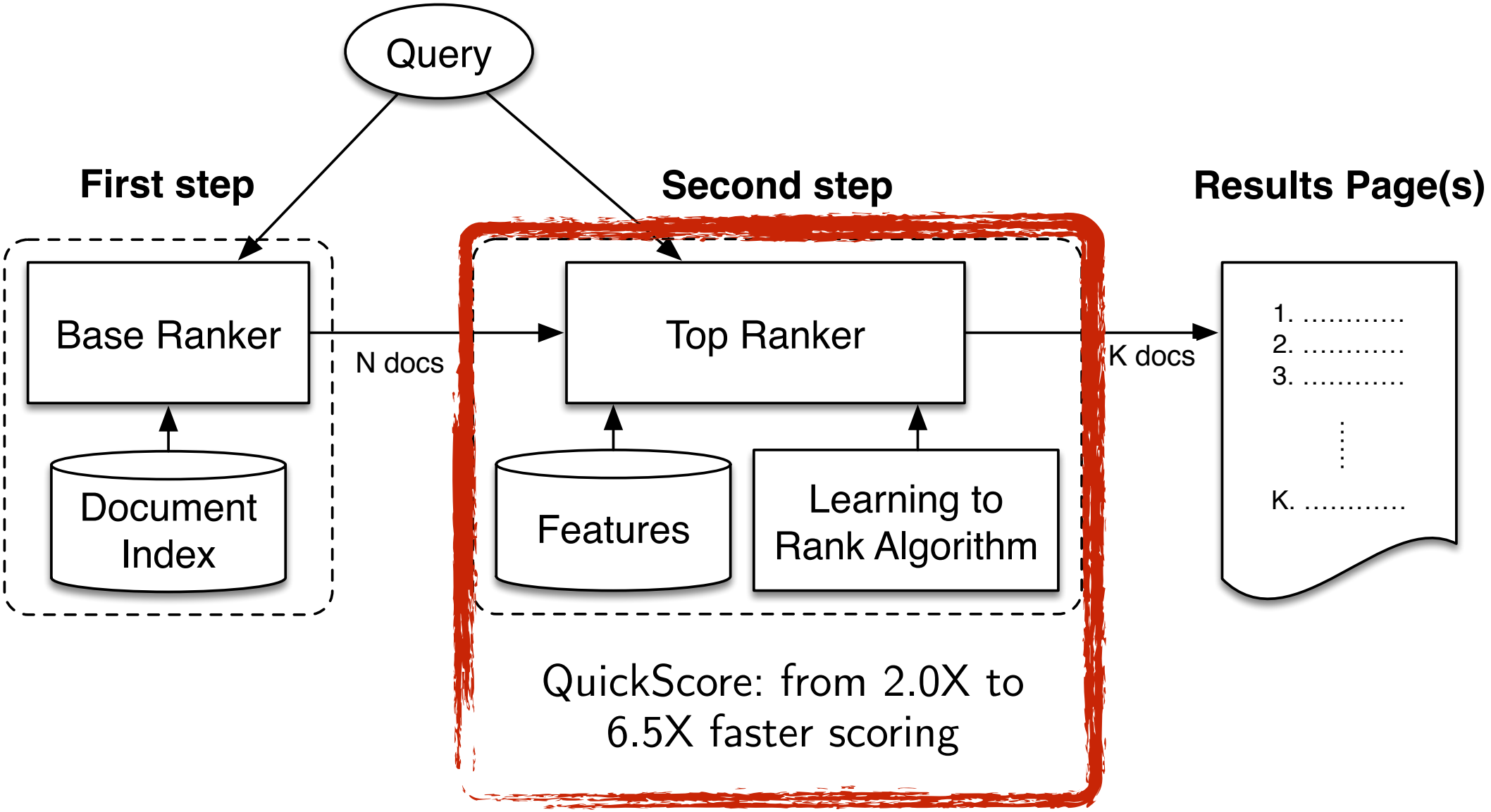
Università Ca' Foscari, Venice, Italy

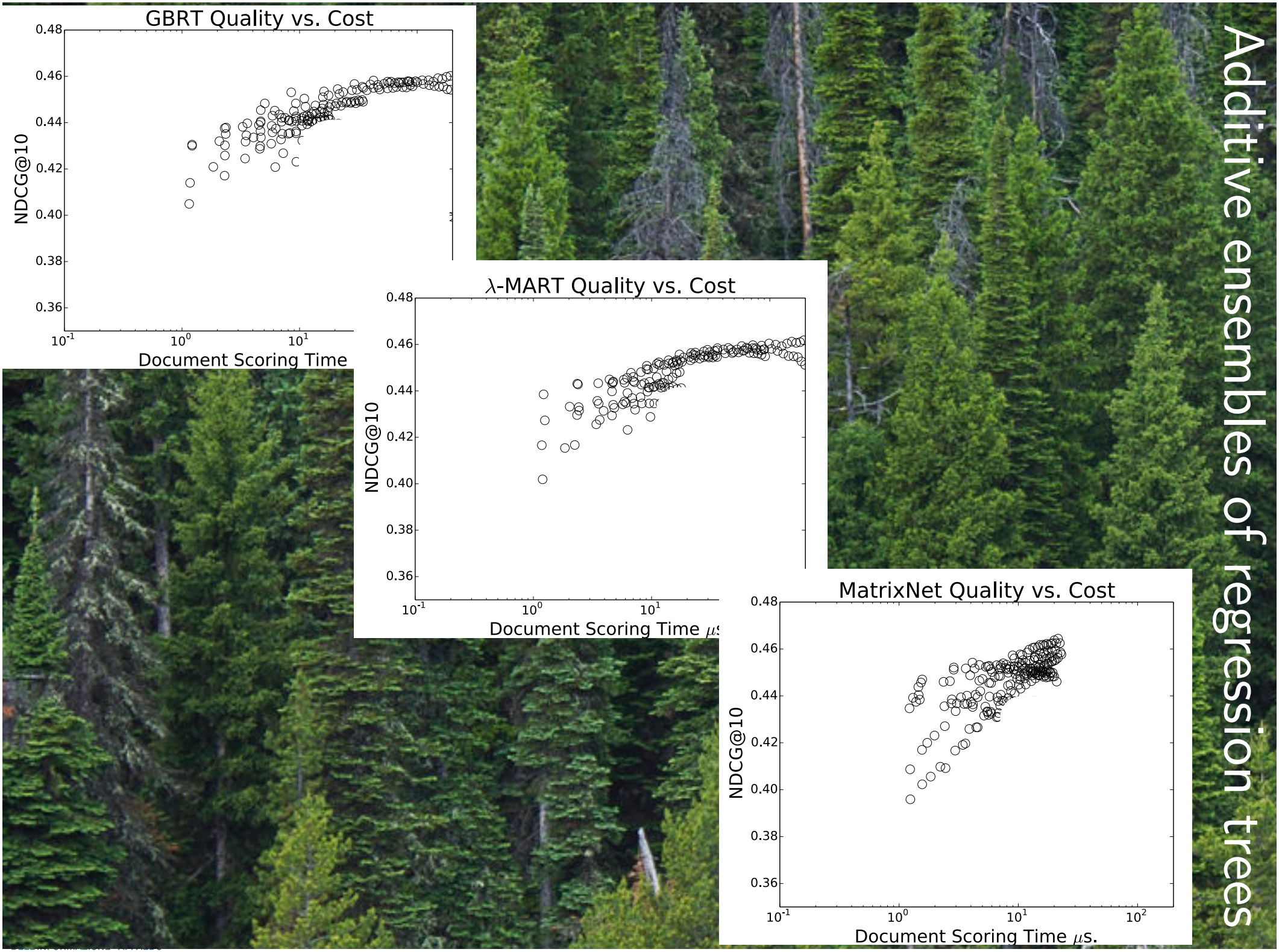
Rossano Venturini

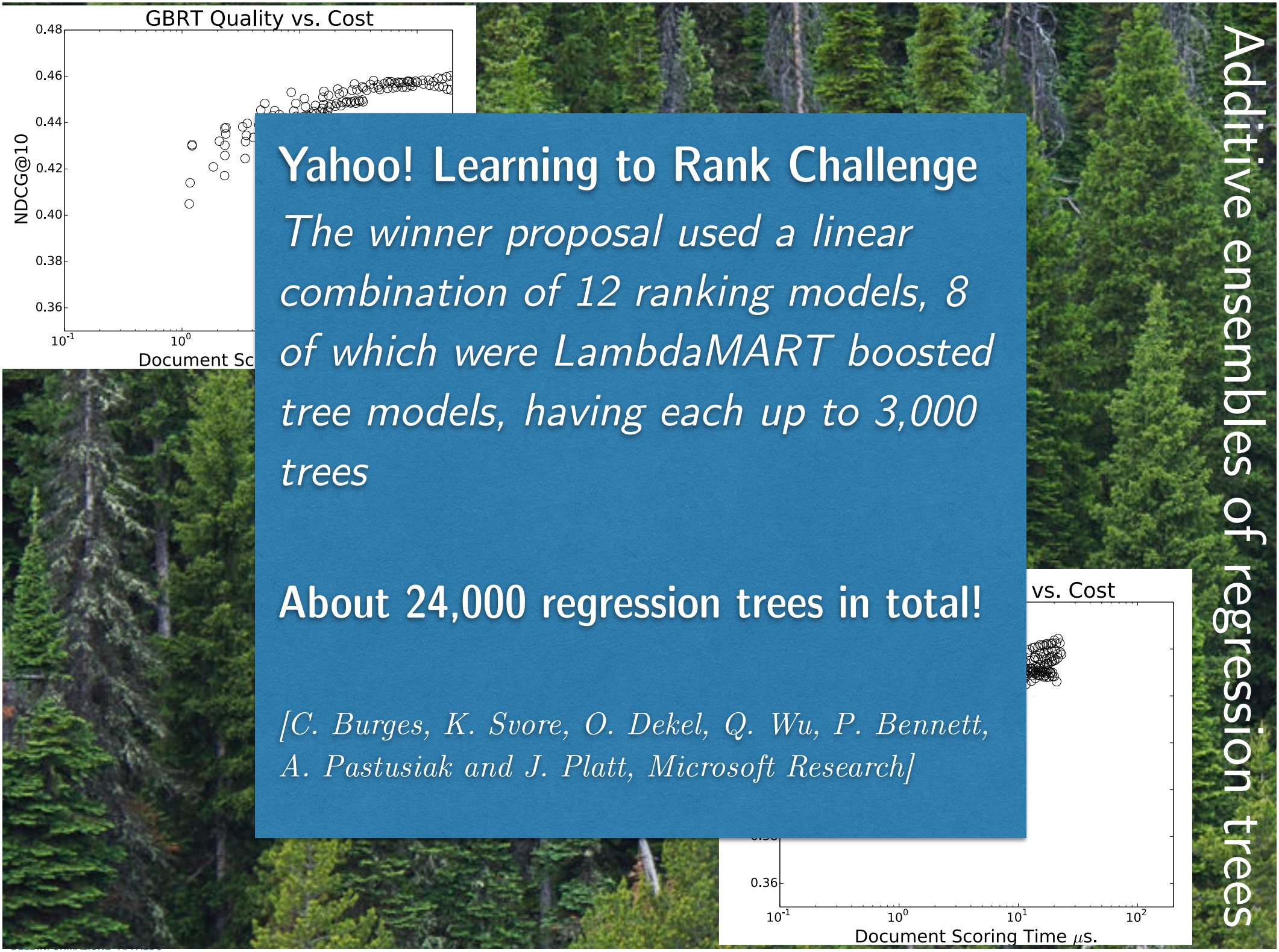
Università di Pisa, Pisa, Italy

Ranking (in web search) is computationally expensive and requires trade-offs between efficiency and efficacy to be devised

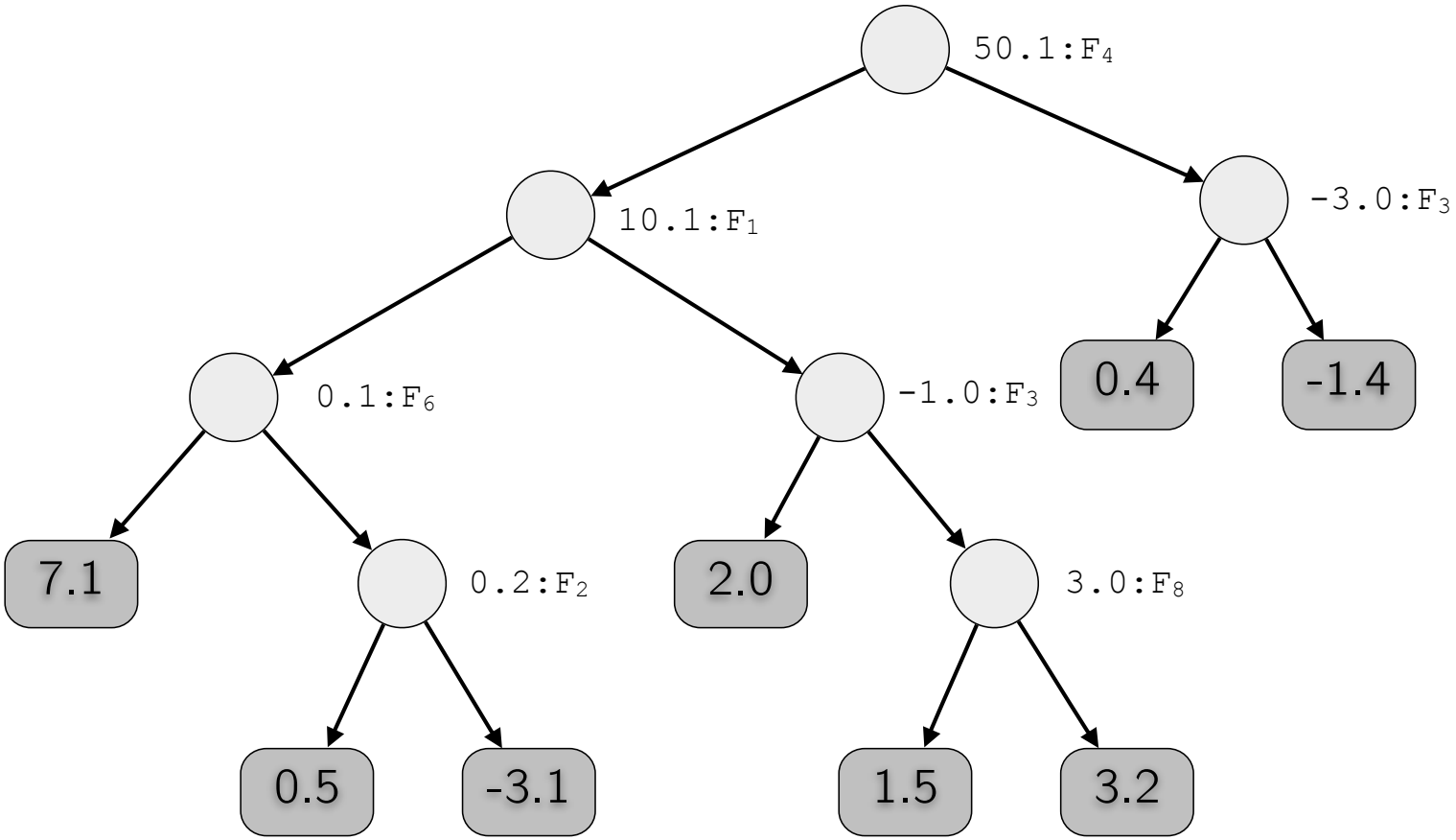
Additive ensembles of regression trees




GBRT Quality vs. Cost
NDCG@10
0.48
0.46
0.44
0.42
0.40
0.38
0.36
10^{-1}
10^{0}
10^{1}
Document Scoring Time
λ-MART Quality vs. Cost
NDCG@10
Document Scoring Time μs
MatrixNet Quality vs. Cost
NDCG@10
10^{2}
Document Scoring Time μs.
Additive ensembles of regression trees


GBRT Quality vs. Cost
NDCG@10
0.48
0.46
0.44
0.42
0.40
0.38
0.36
10^{-1}
10^{0}
Document Sc
Yahoo! Learning to Rank Challenge
The winner proposal used a linear combination of 12 ranking models, 8 of which were LambdaMART boosted tree models, having each up to 3,000 trees
About 24,000 regression trees in total!
[C. Burges, K. Svore, O. Dekel, Q. Wu, P. Bennett, A. Pastusiak and J. Platt, Microsoft Research]
vs. Cost
0.36
10^{-1}
10^{0}
10^{1}
10^{2}
Document Scoring Time μs.
Additive ensembles of regression trees

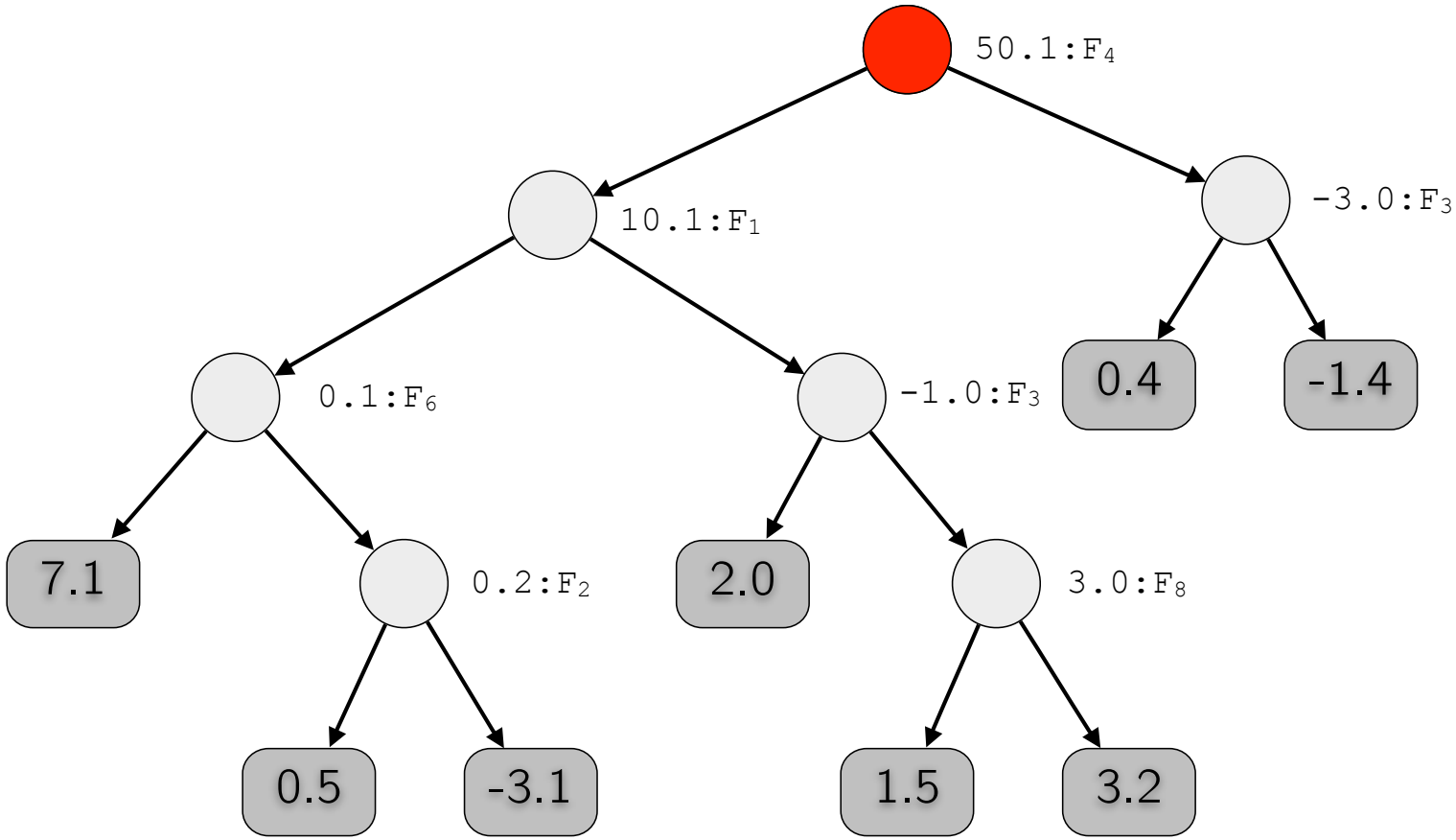
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

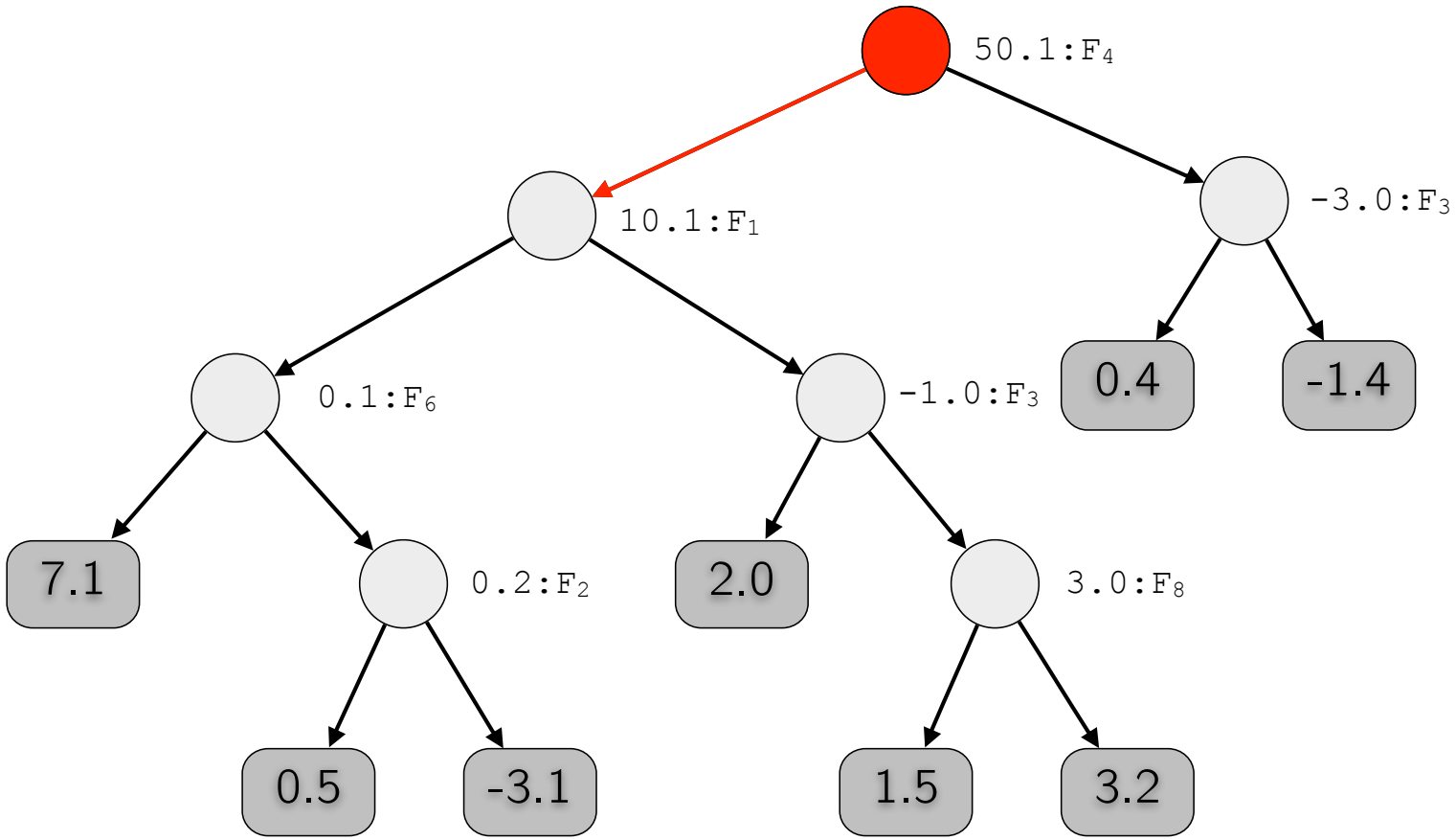
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

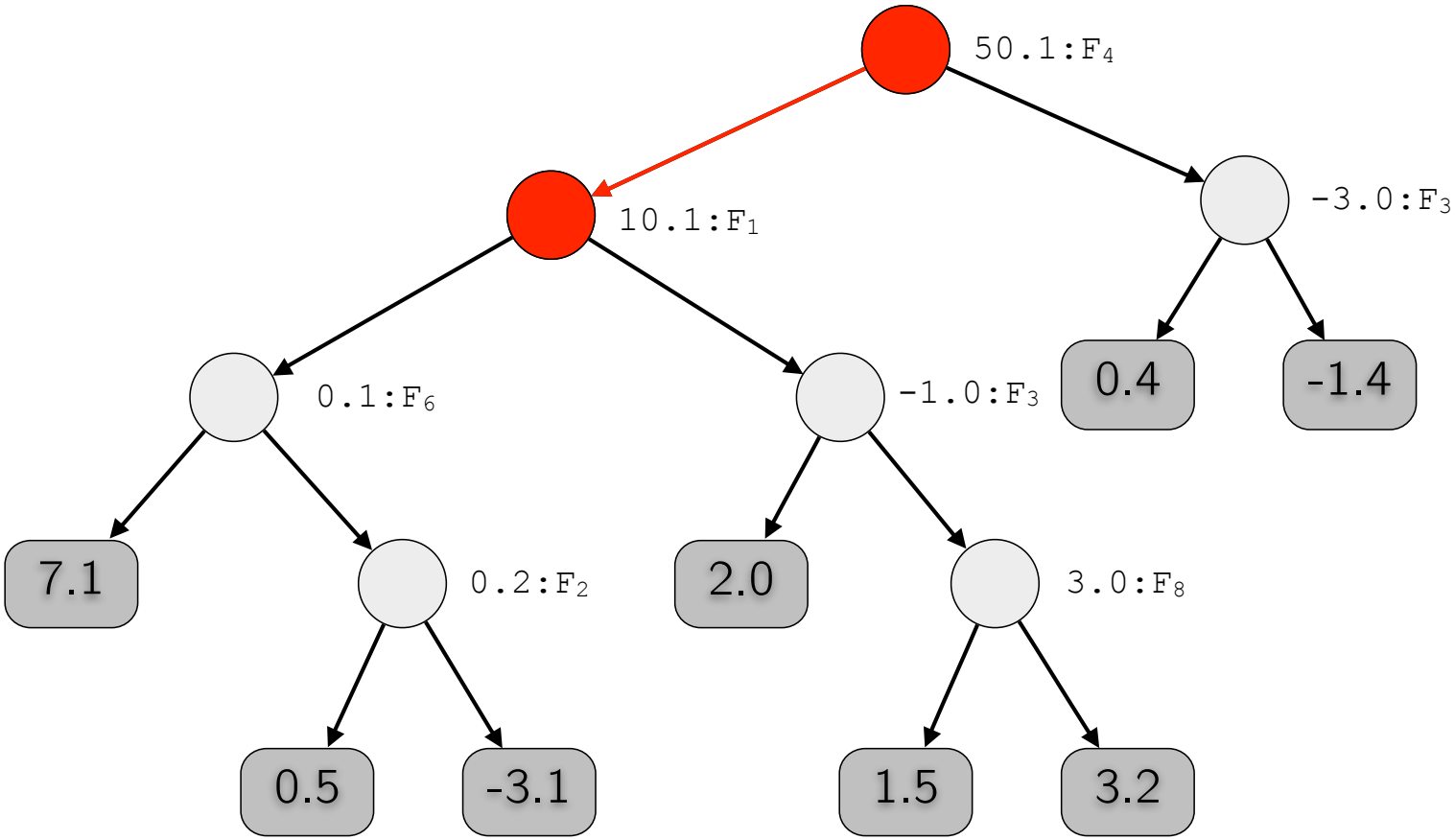
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

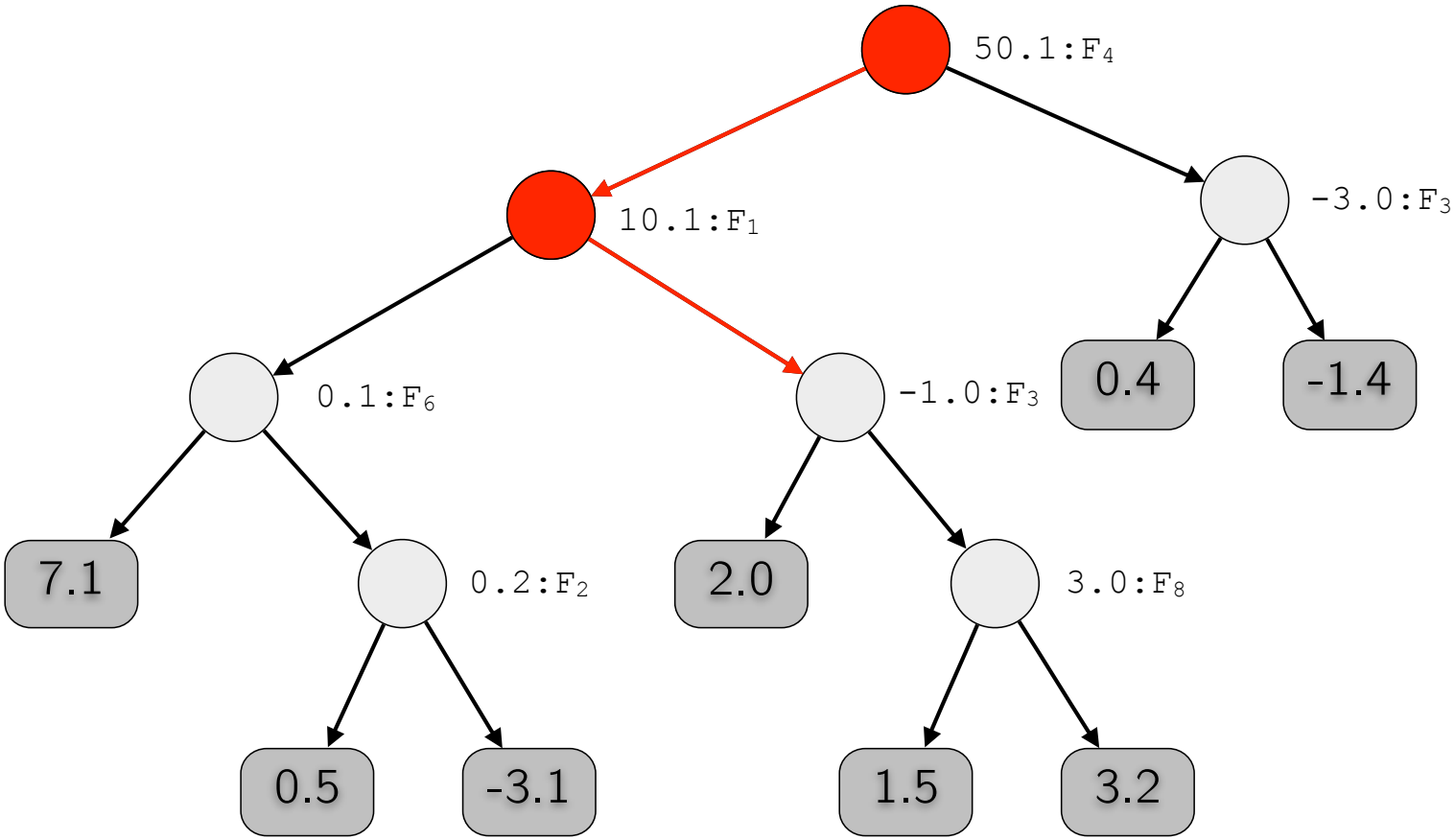
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

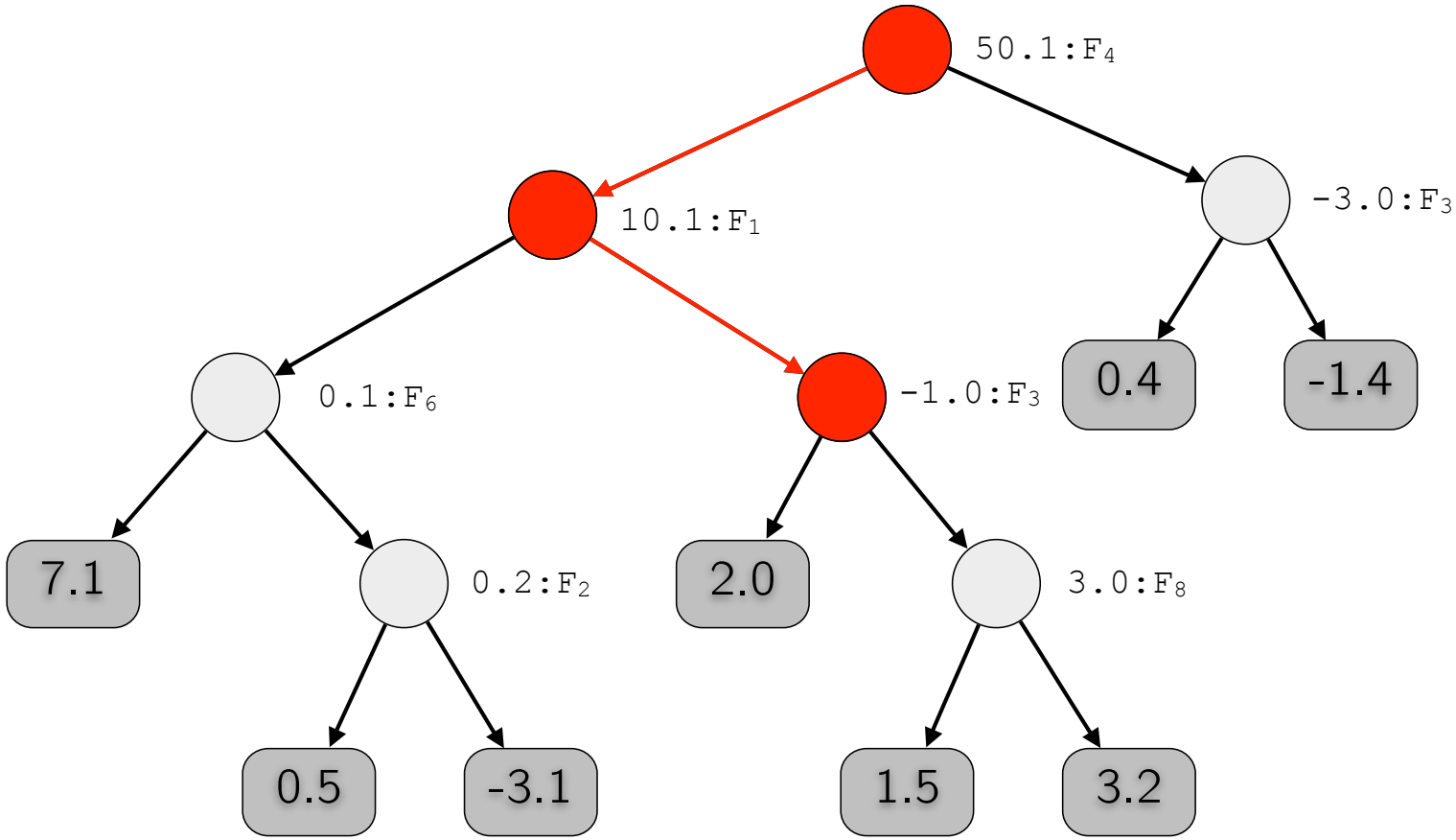
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

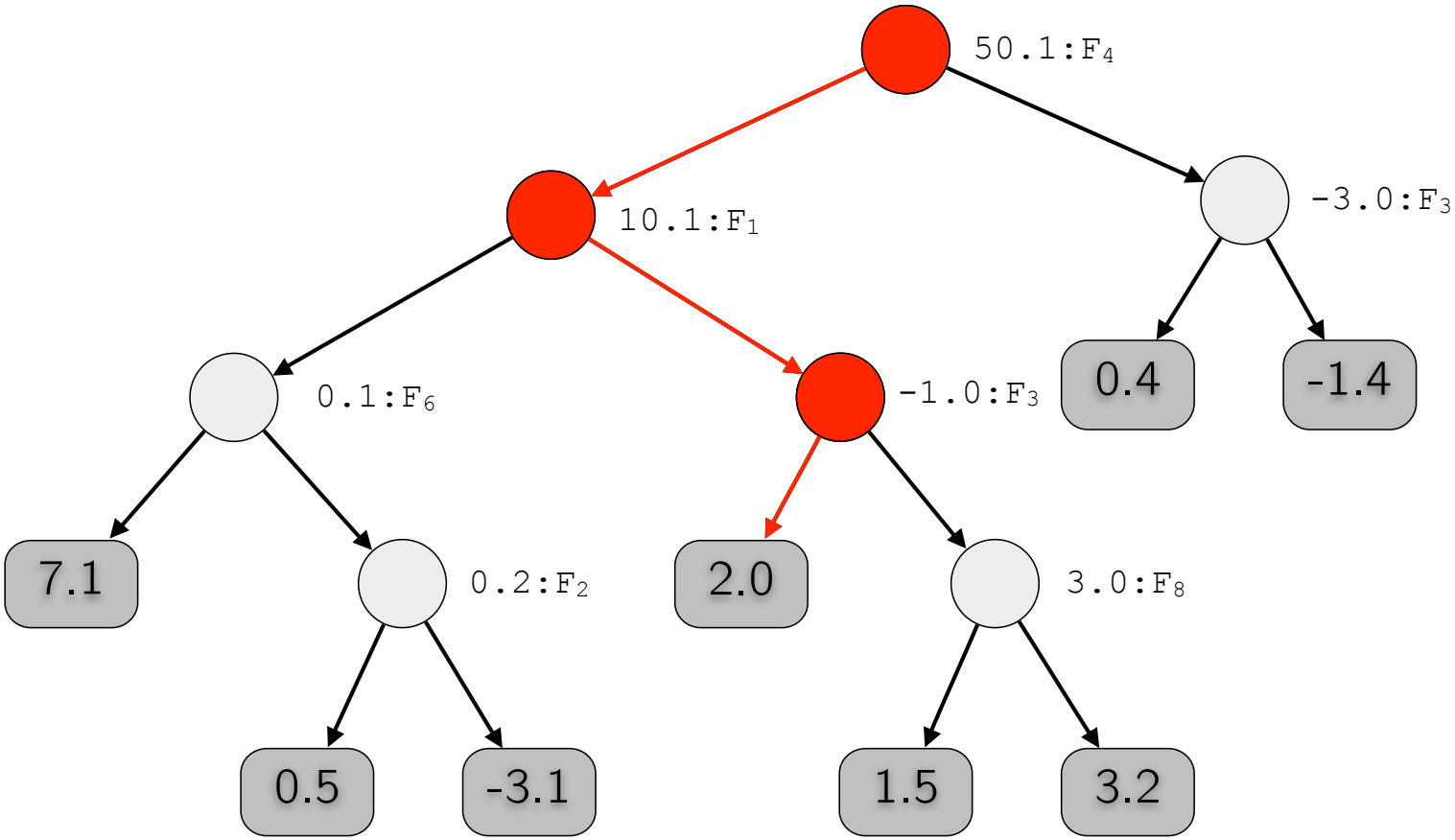
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

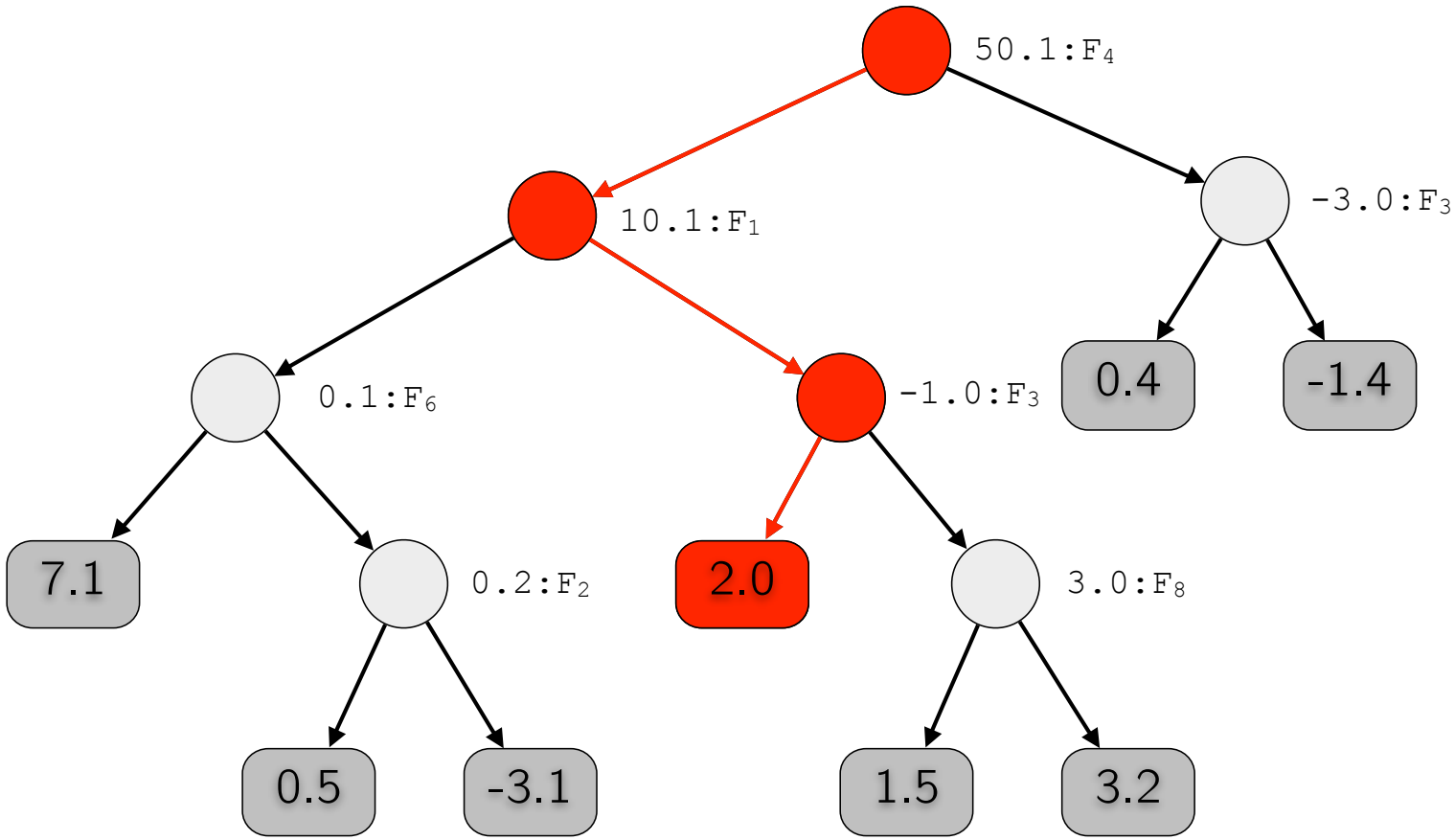
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

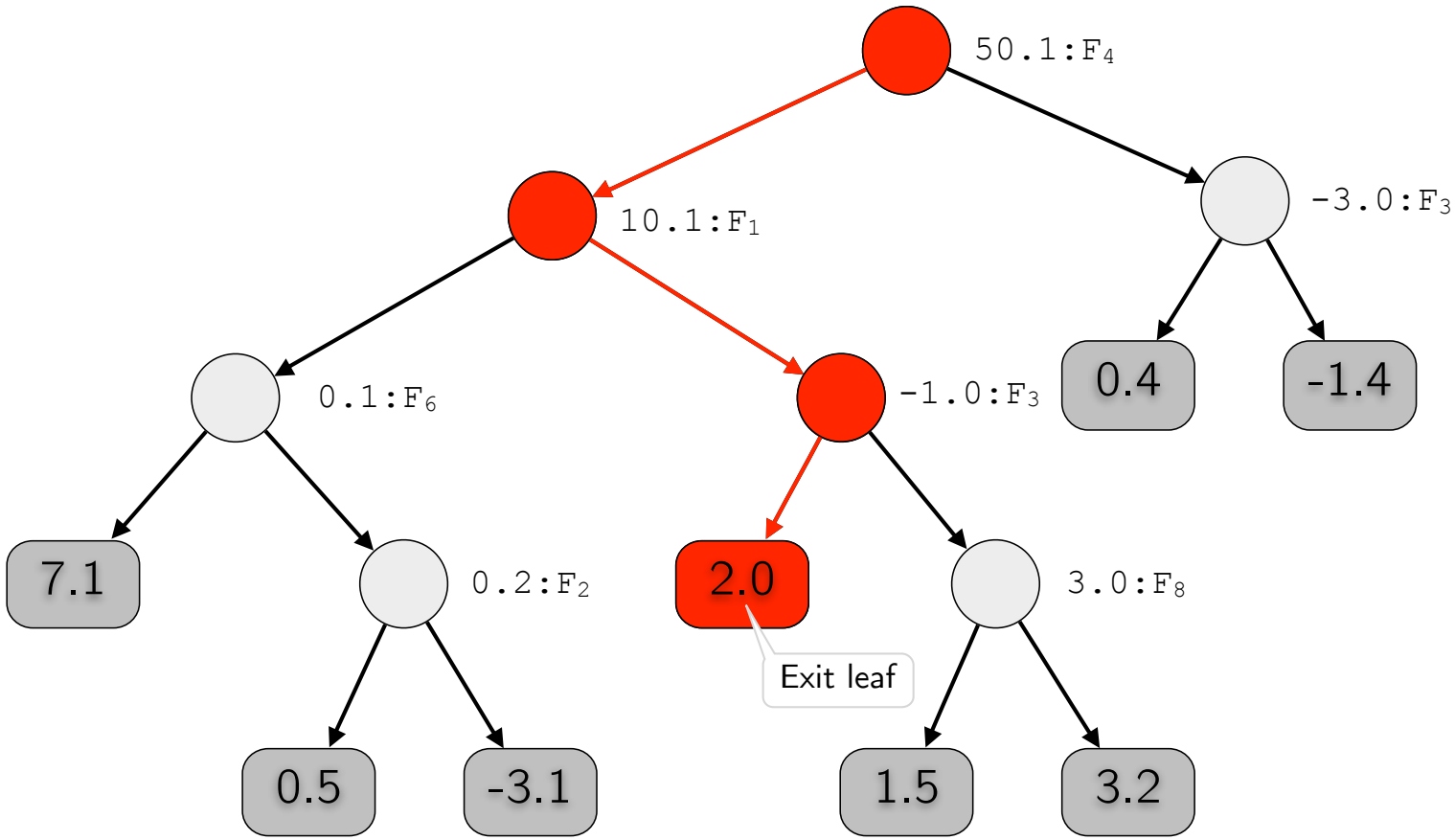
Process of Query-Document Scoring



Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

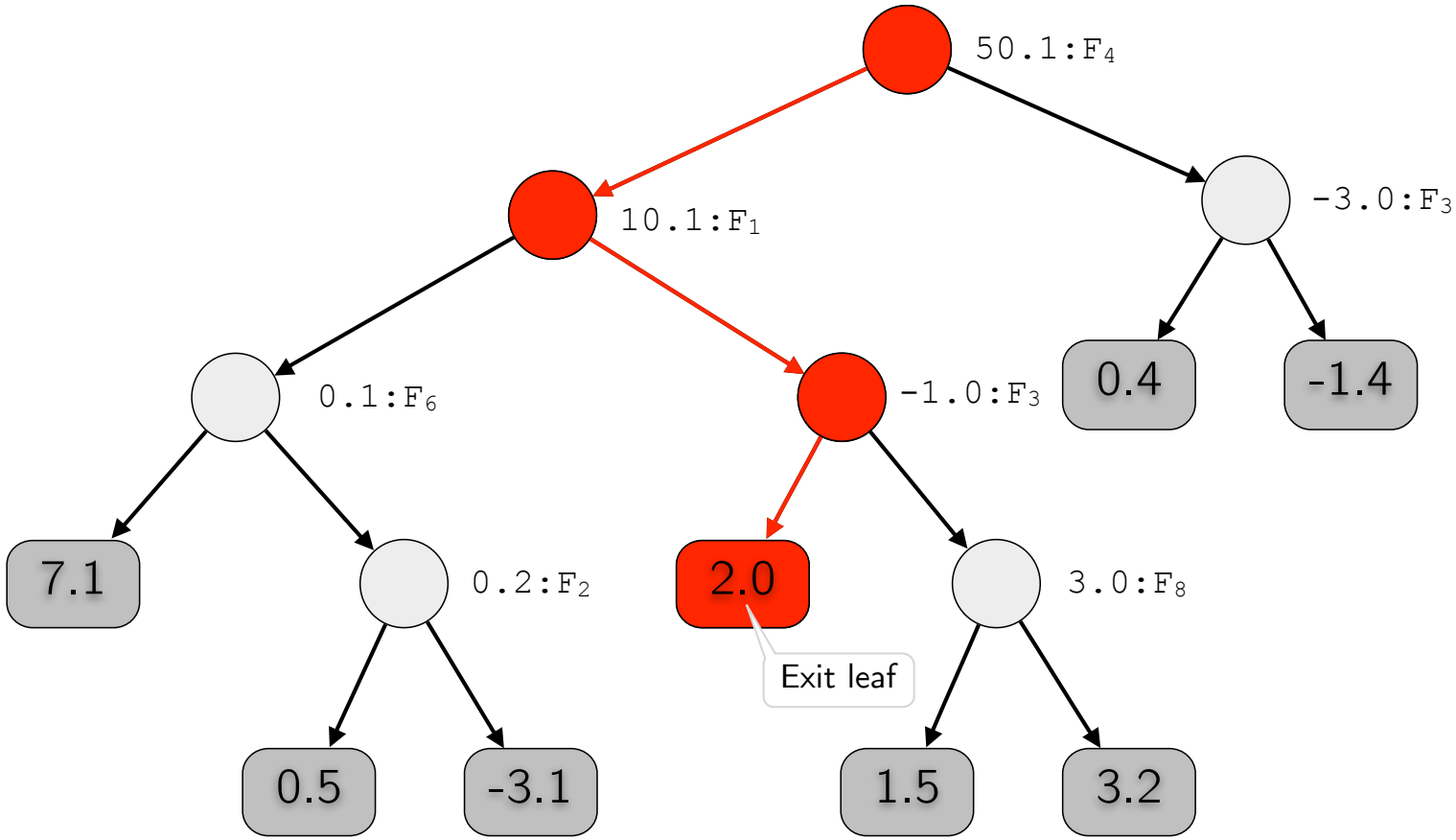
Process of Query-Document Scoring



Query-Document feature set

F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

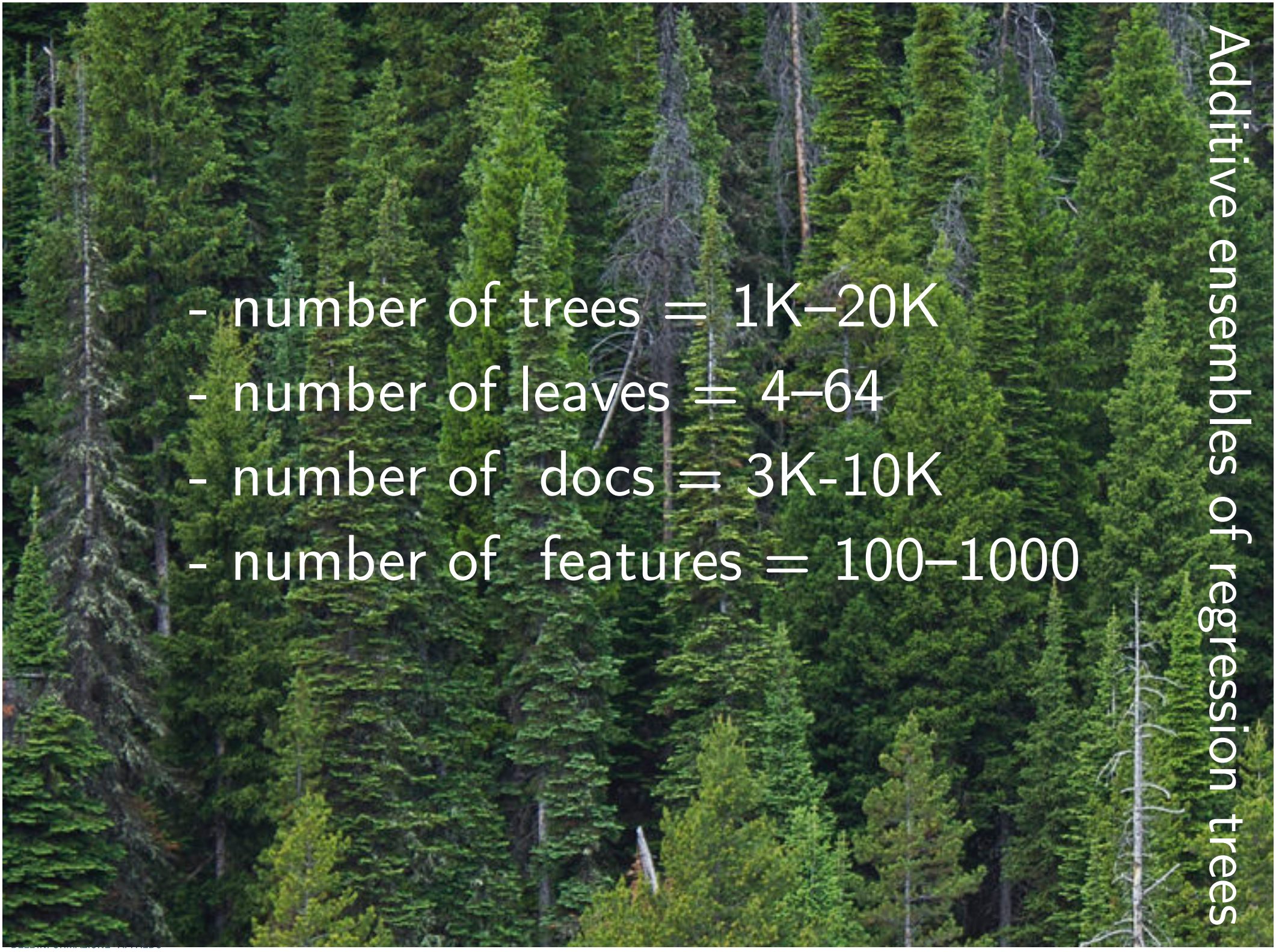
Process of Query-Document Scoring

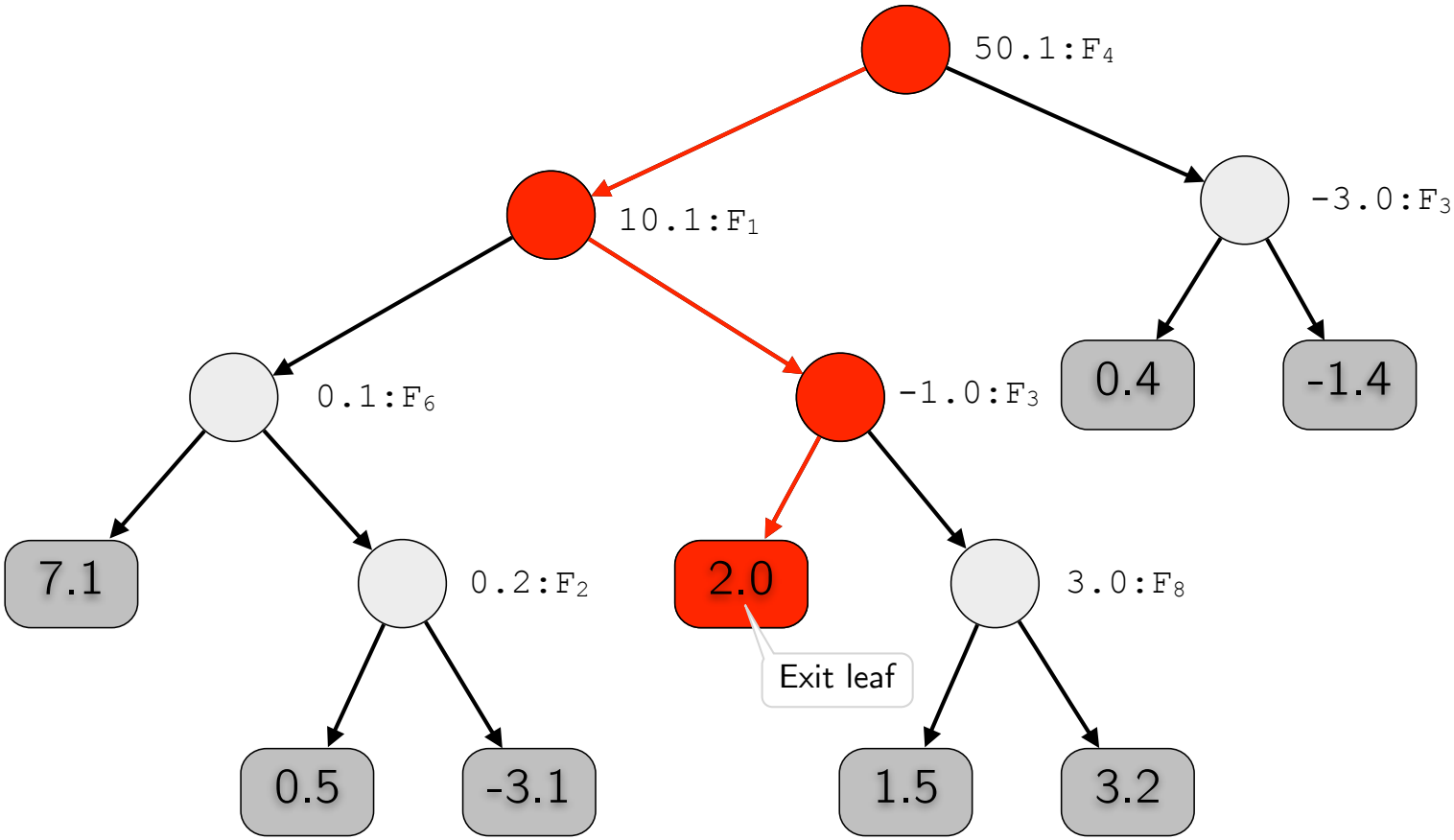


Query-Document feature set

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

Score += 2.0


Additive ensembles of regression trees
- number of trees = 1K–20K
- number of leaves = 4–64
- number of docs = 3K-10K
- number of features = 100–1000



SoA: Struct+

Query-Document feature sets

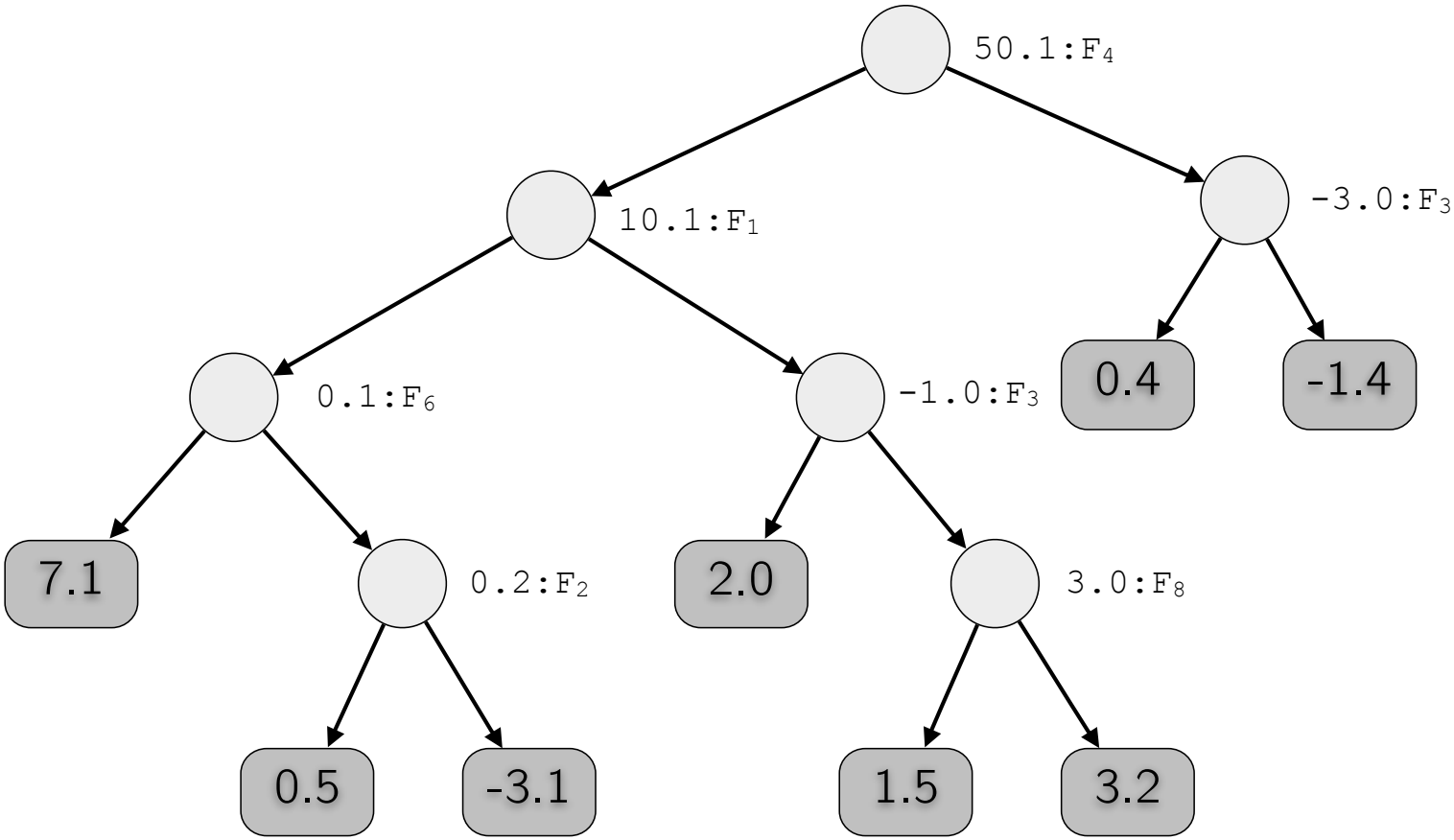
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

Naïve baseline

Each tree node is represented by a C++ object containing the feature id, the associated threshold and the left and right pointers.

SoA: If-then-else



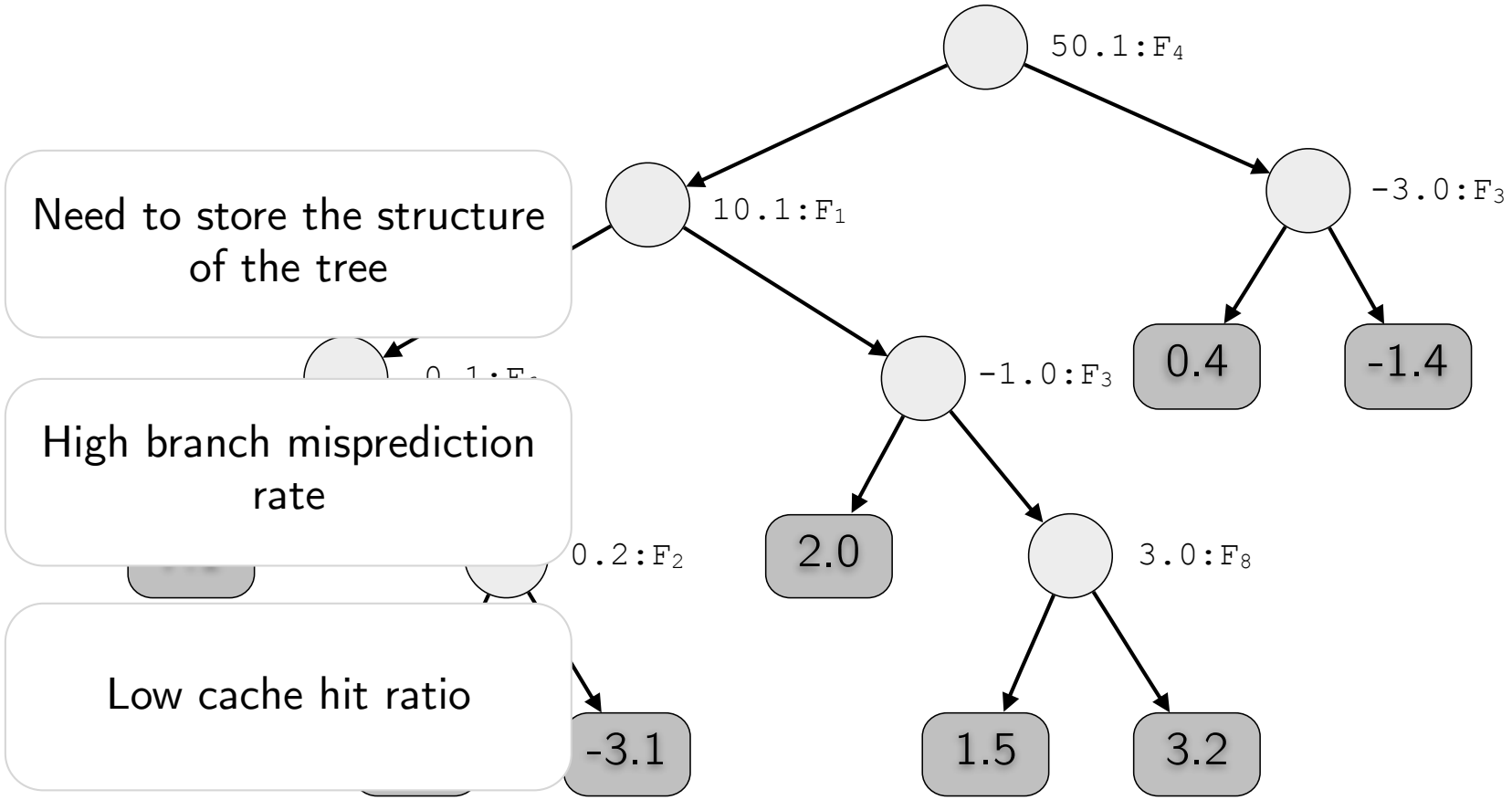
Query-Document feature sets

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
if (x[4] <= 50.1) {
    // recurses on the left subtree
    ...
} else {
    // recurses on the right subtree
    if(x[3] <= -3.0)
        result = 0.4;
    else
        result = -1.4;
}
```

SoA: If-then-else



Need to store the structure of the tree

High branch misprediction rate

Low cache hit ratio

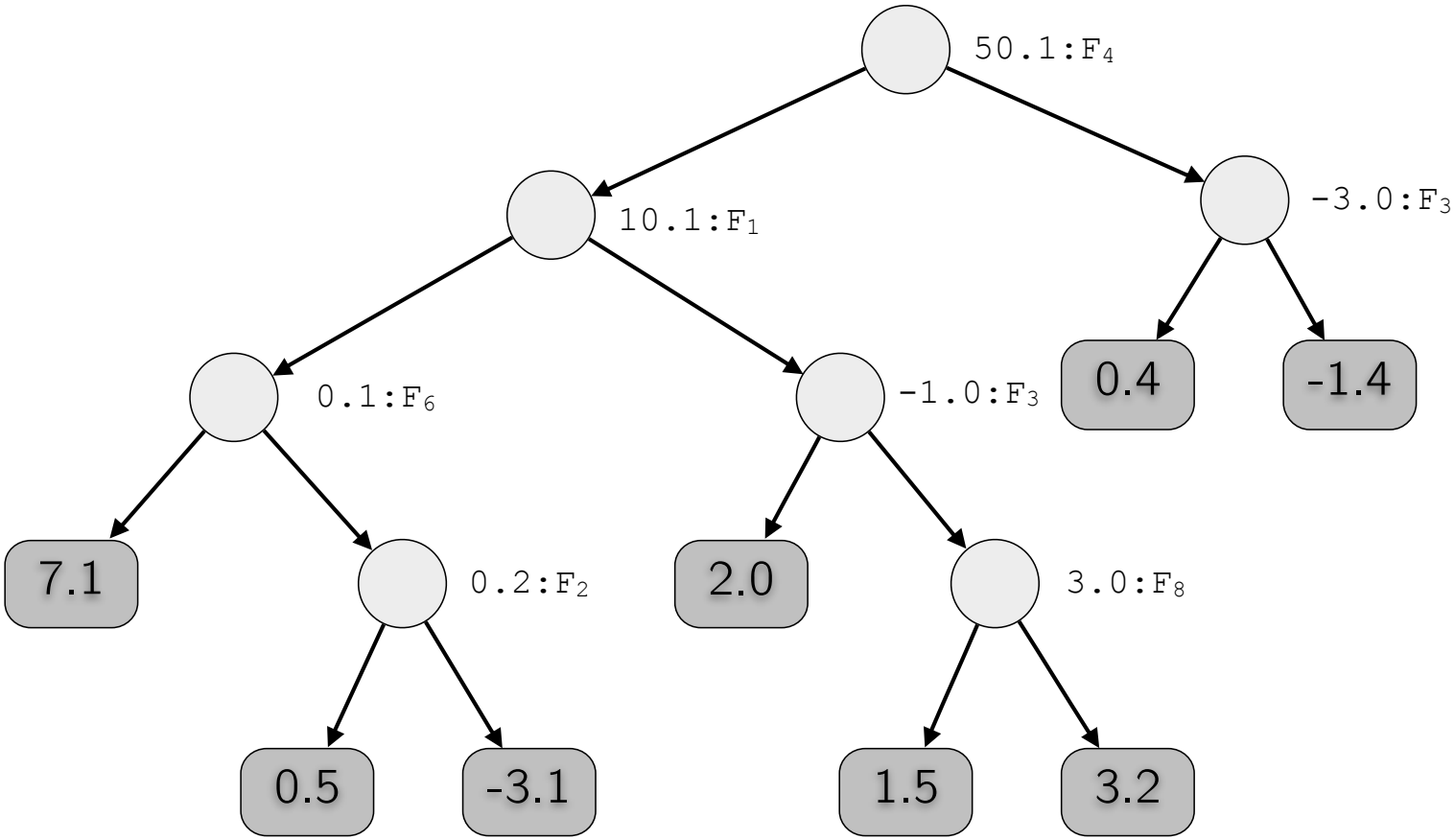
Query-Document feature sets

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
if (x[4] <= 50.1) {
    // recurses on the left subtree
    ...
} else {
    // recurses on the right subtree
    if(x[3] <= -3.0)
        result = 0.4;
    else
        result = -1.4;
}
```

SoA: VPred [Asadi et Al. TK



Query-Document feature sets

16 docs

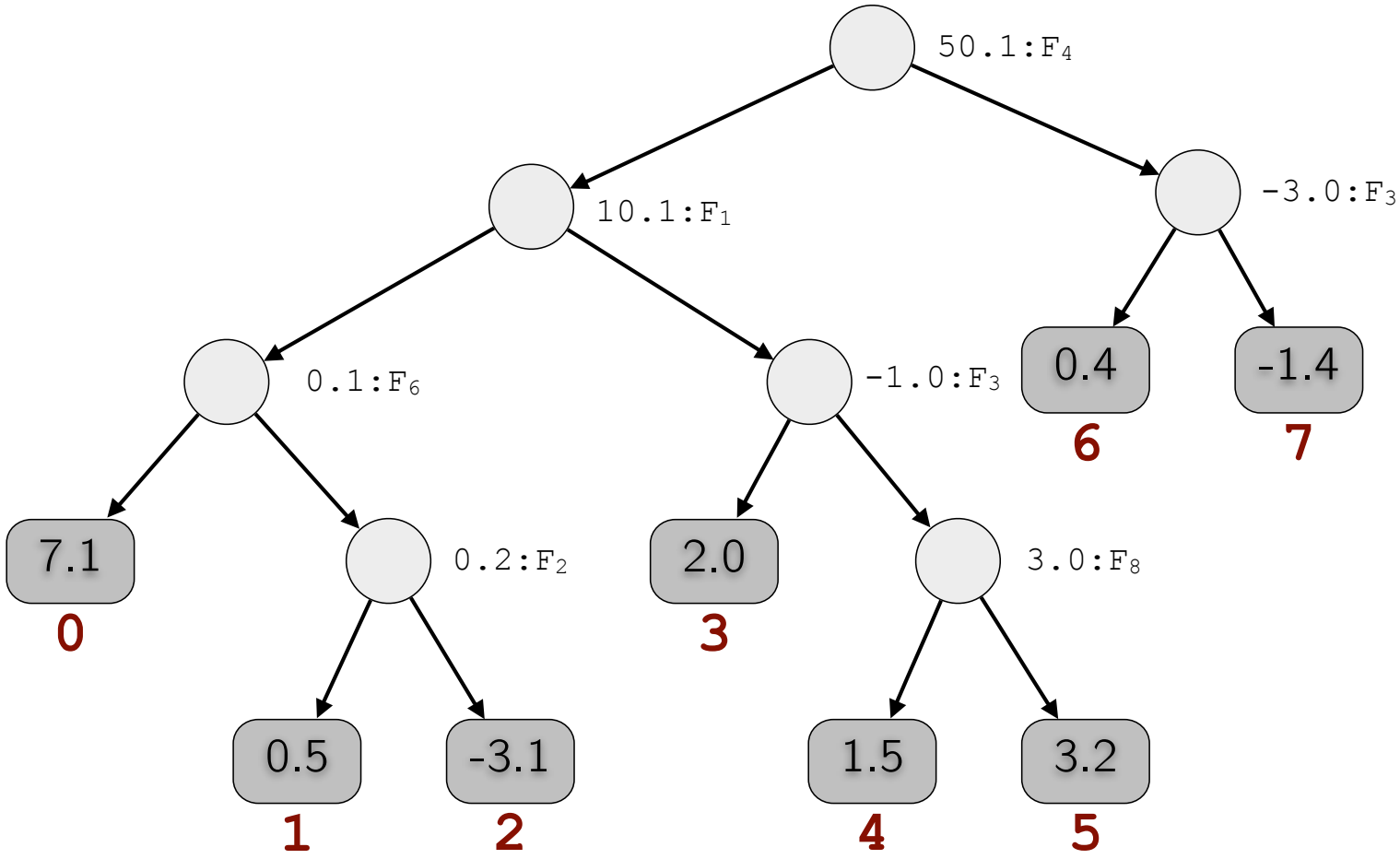
F1	F2	F3	F4	F5	F6
13.3	0.12	-1.2	43.9	11	-0.4
10.9	0.08	-1.1	42.9	15	-0.3
11.2	0.6	-0.2	54.1	13	-0.5

...

```
double depth4(float* x, Node* nodes) {
    int nodeId = 0;
    nodeId = nodes->children[x[nodes[nodeId].fid] > nodes[nodeId].theta];
    nodeId = nodes->children[x[nodes[nodeId].fid] > nodes[nodeId].theta];
    nodeId = nodes->children[x[nodes[nodeId].fid] > nodes[nodeId].theta];
    nodeId = nodes->children[x[nodes[nodeId].fid] > nodes[nodeId].theta];
    return scores[nodeId];
    }
```


QuickScore, a new efficient algorithm for the interleaved traversal of additive ensembles of regression trees by means of simple logical bitwise operations
PATENT
filed for international patent
PENDING

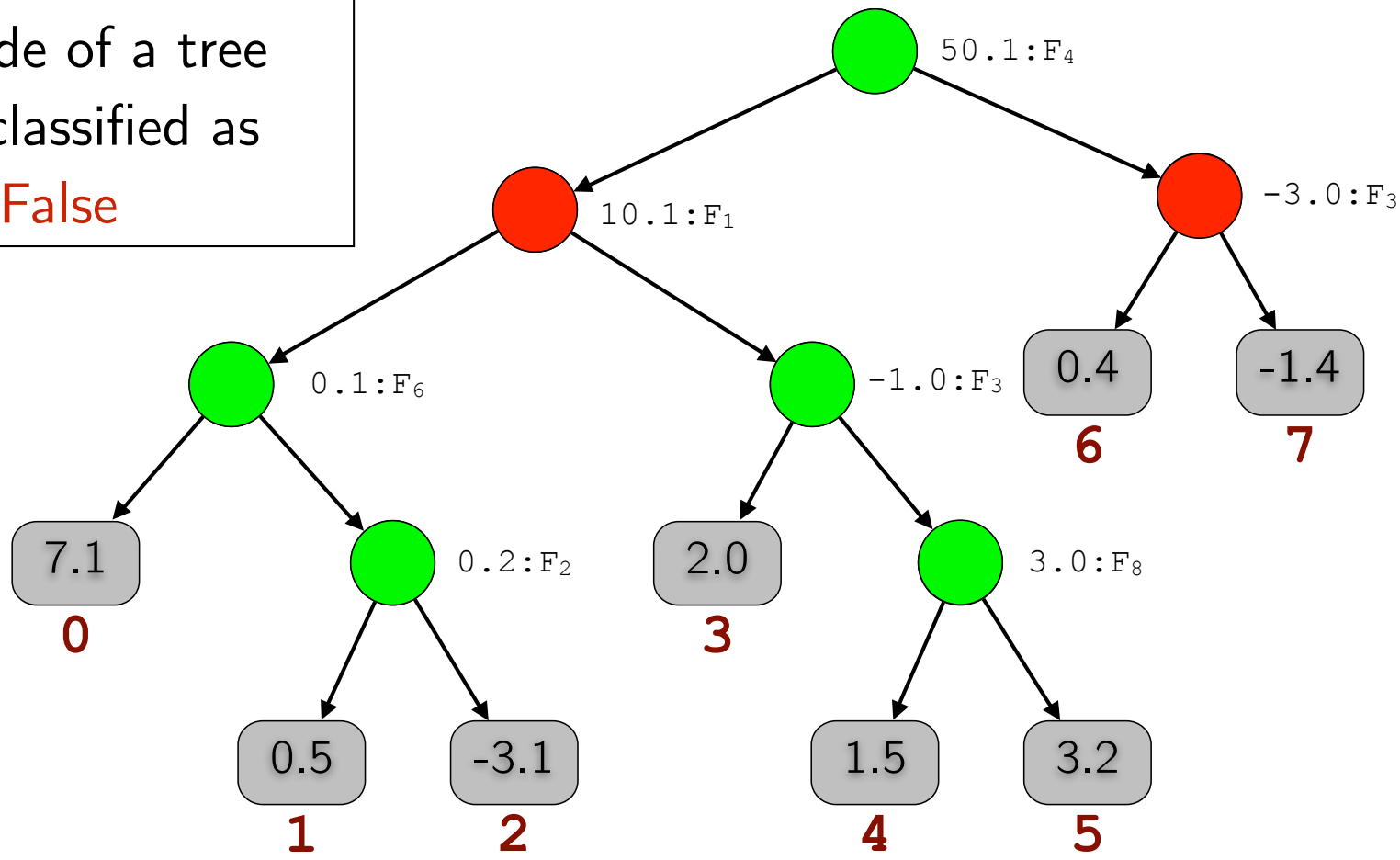
QuickScore: false and true nodes



F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: false and true nodes

Given the feature set, each node of a tree can be classified as True or False



F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

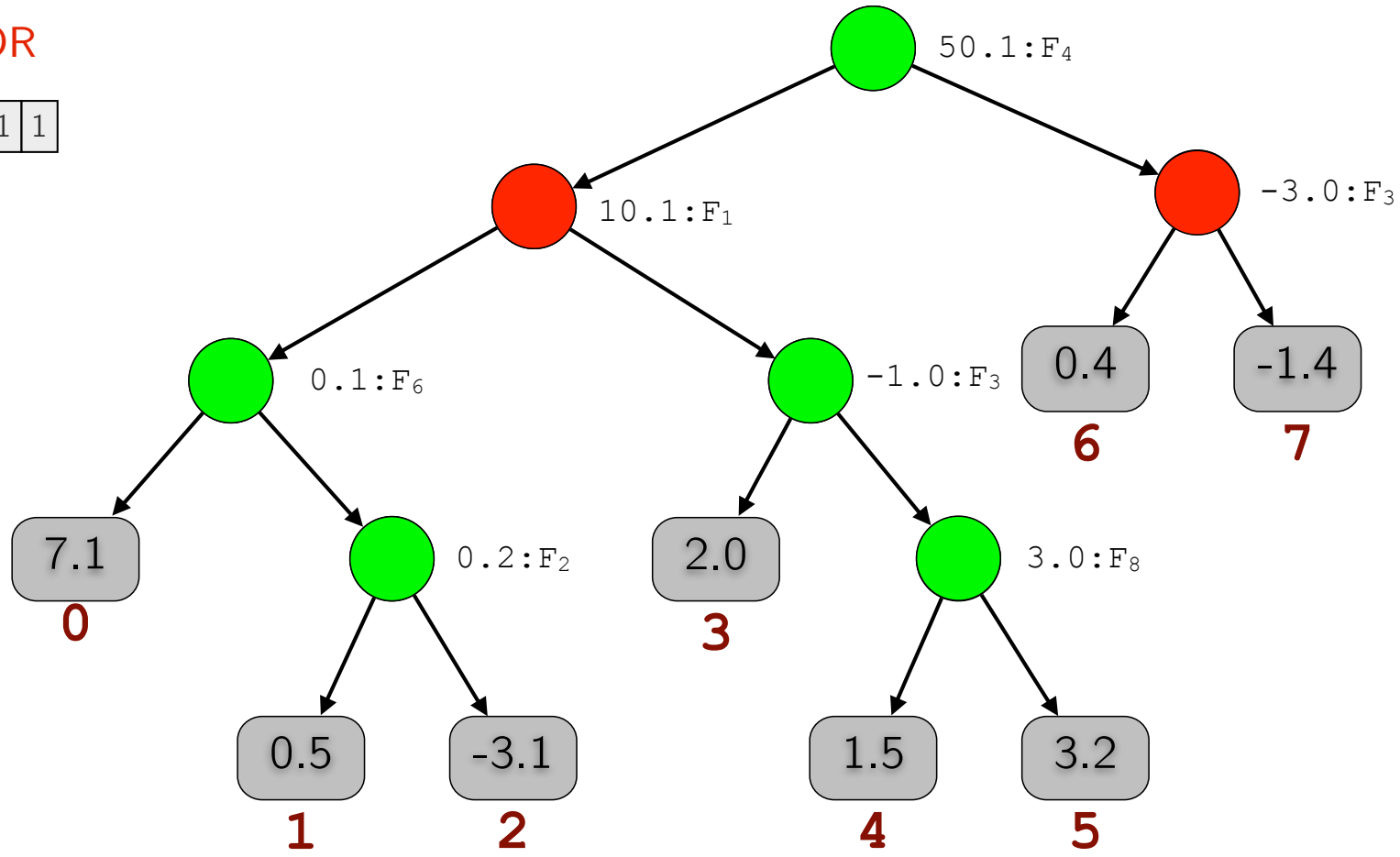
True Node

False Node

QuickScore: Single Tree Traversal

BITVECTOR

1	1	1	1	1	1	1	1

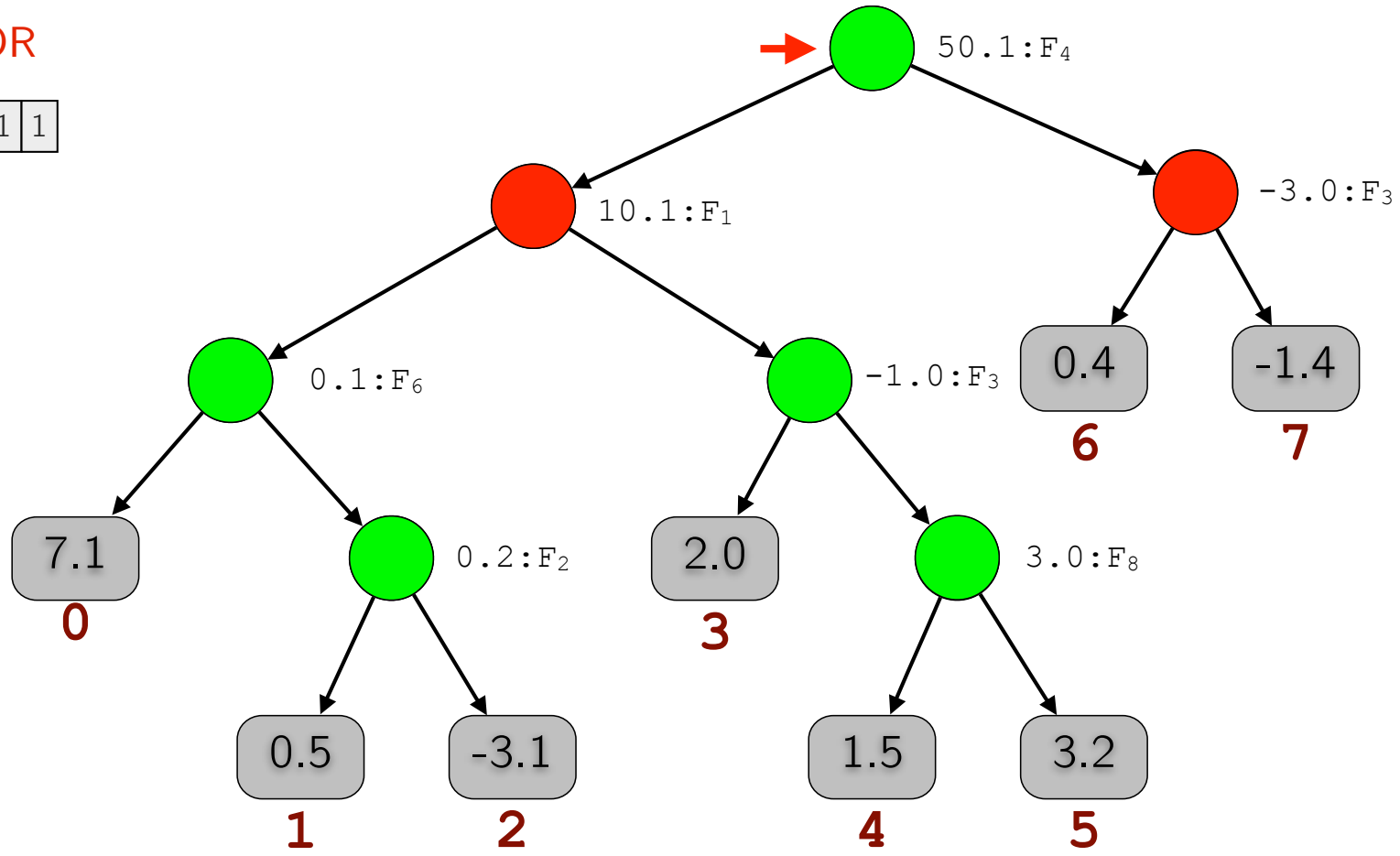


F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal

BITVECTOR

1	1	1	1	1	1	1	1

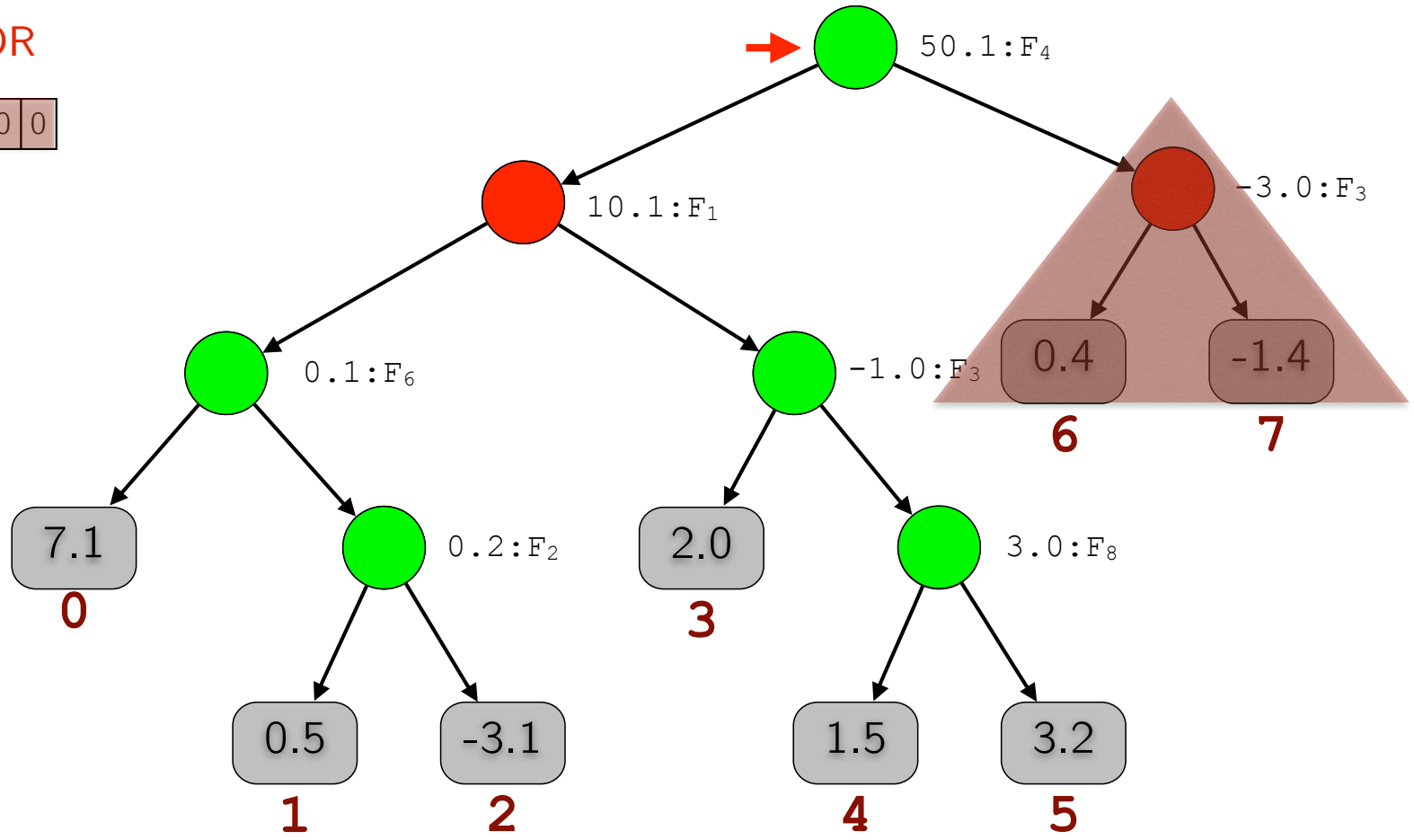


F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal

BITVECTOR

1	1	1	1	1	1	0	0

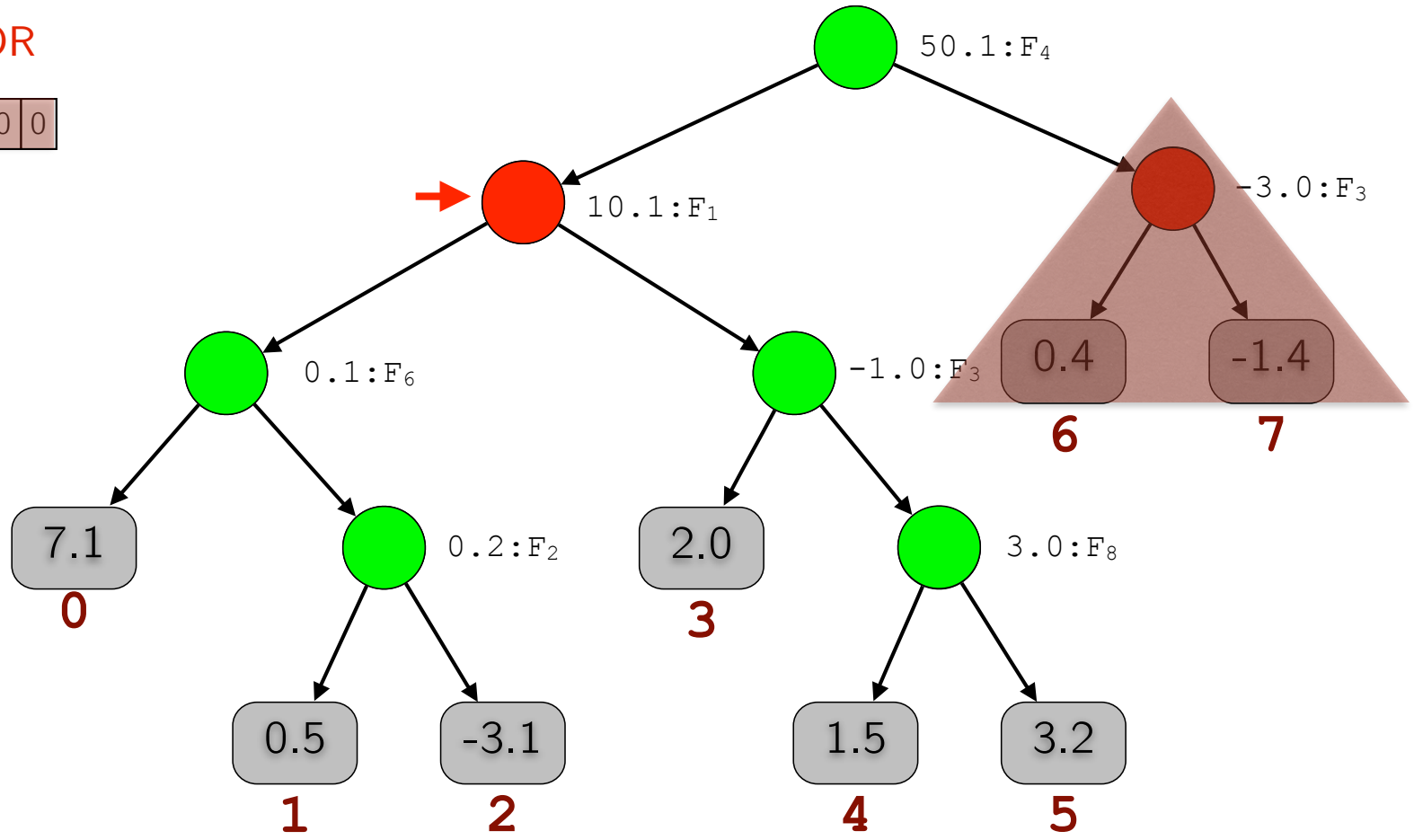


F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal

BITVECTOR

1	1	1	1	1	1	0	0

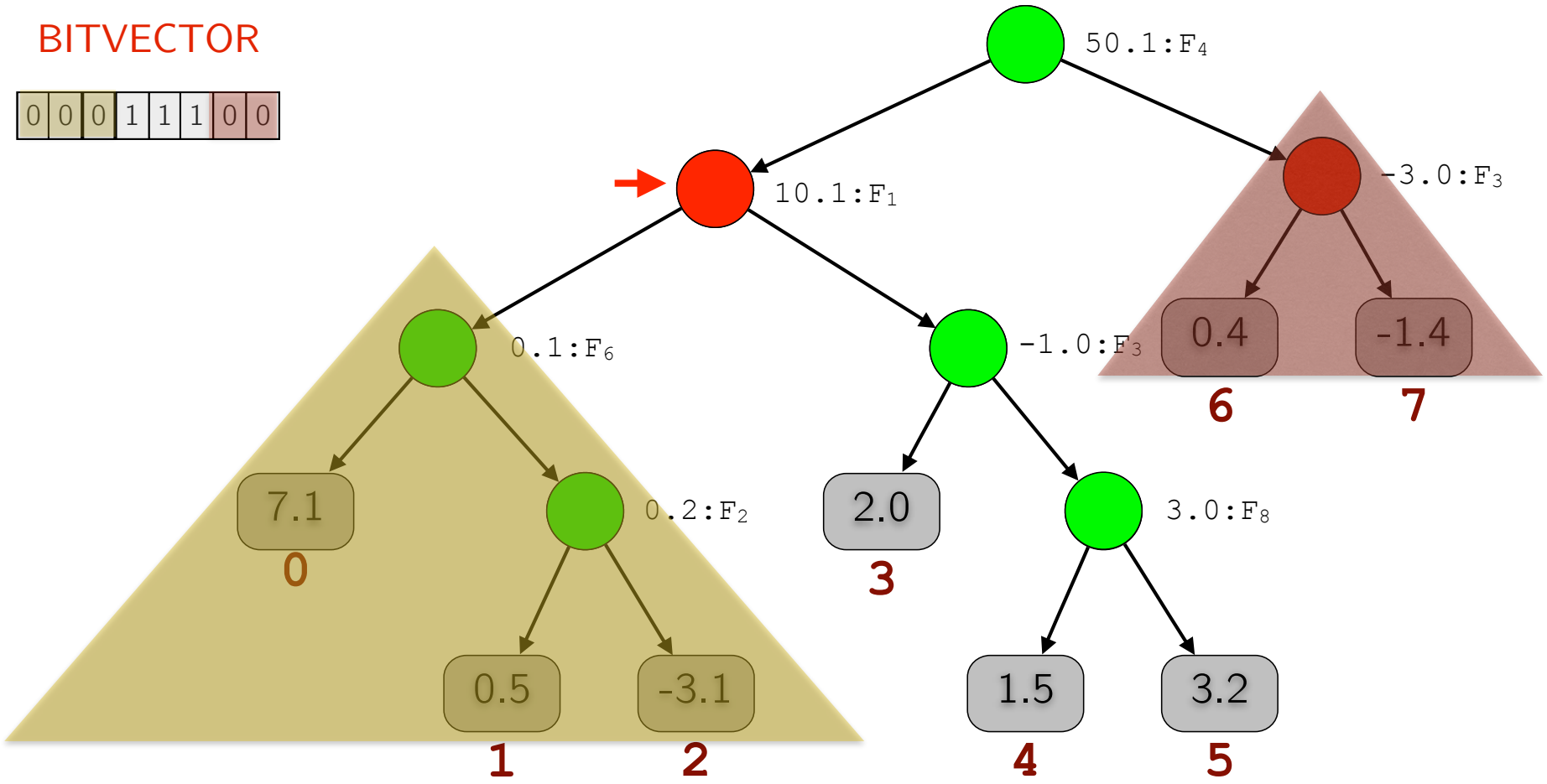


F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal

BITVECTOR

0	0	0	1	1	1	0	0

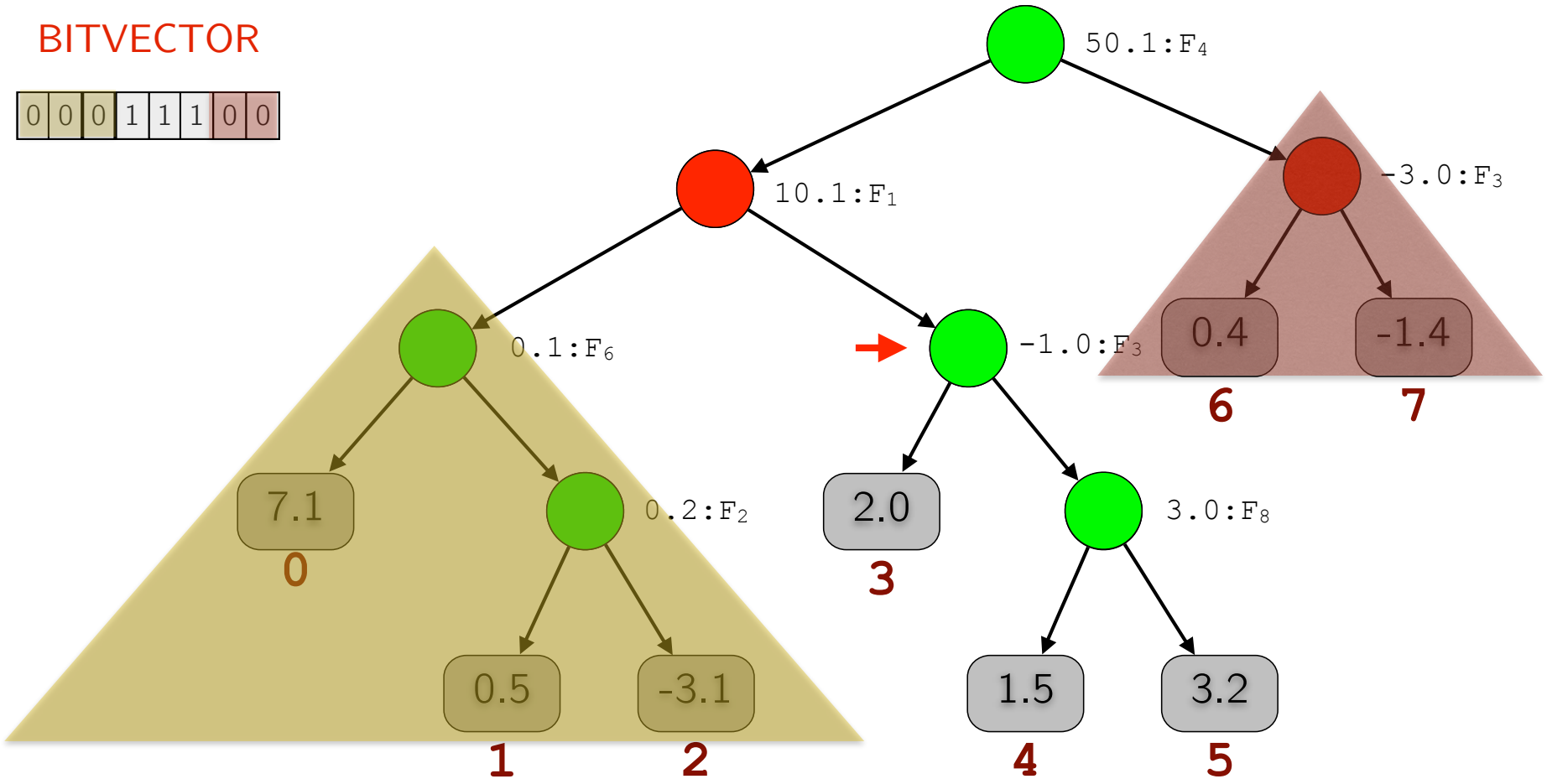


F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal

BITVECTOR

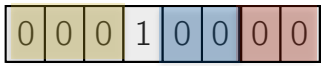
0	0	0	1	1	1	0	0



F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

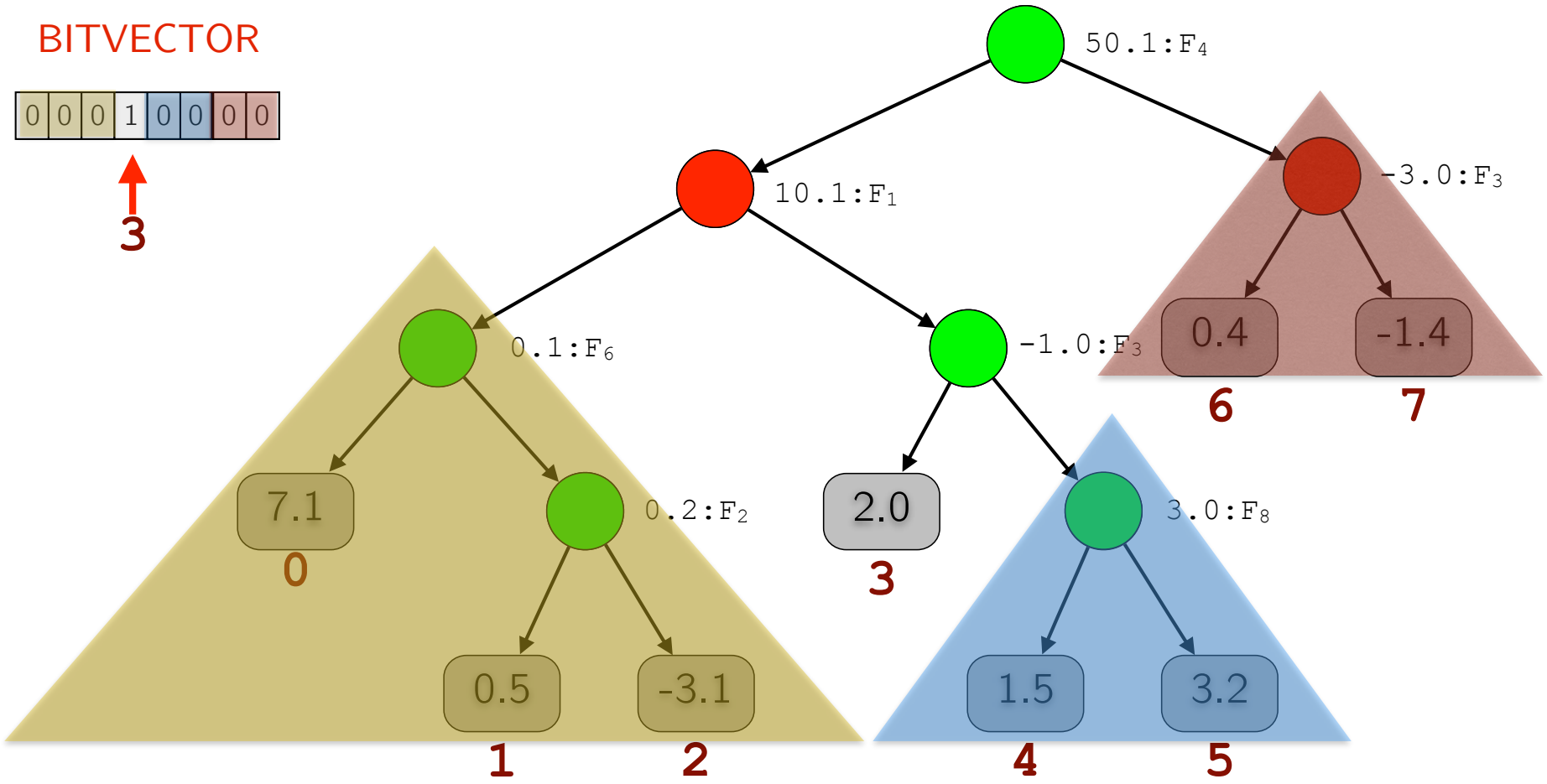
QuickScore: Single Tree Traversal

BITVECTOR



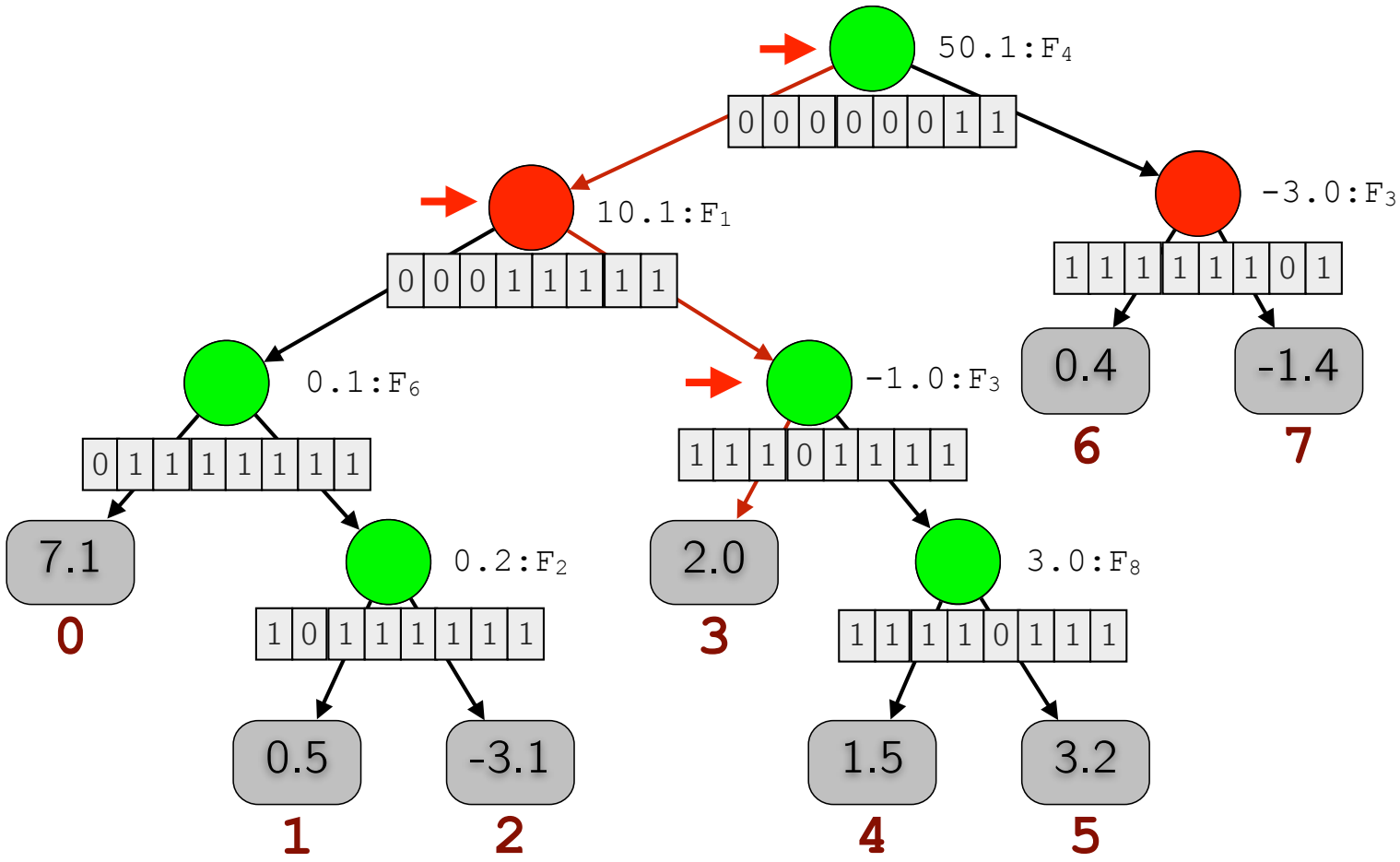
F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: Single Tree Traversal



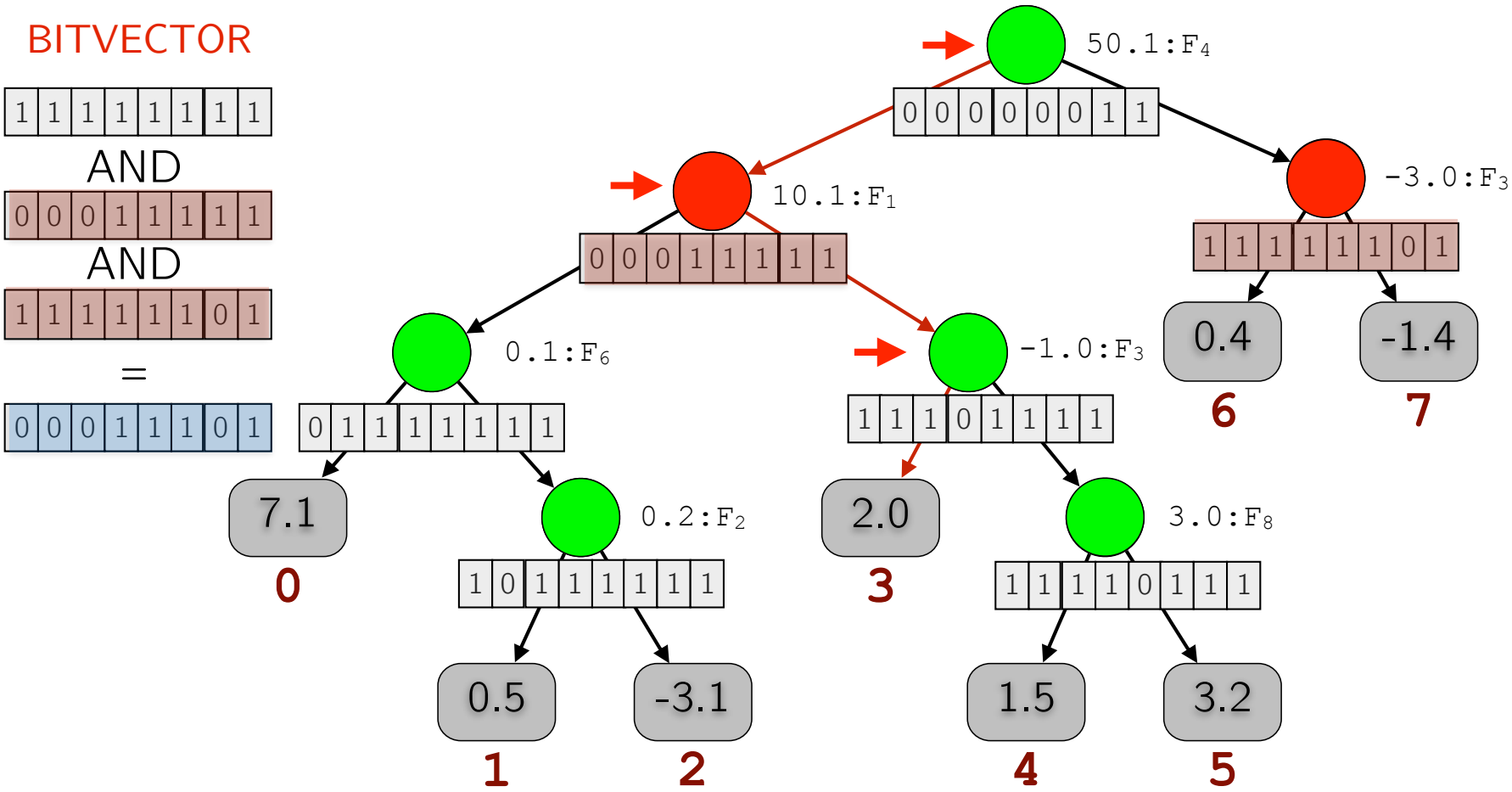
F1	F2	F3	F4	F5	F6	F7	F8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: use of false nodes' masks



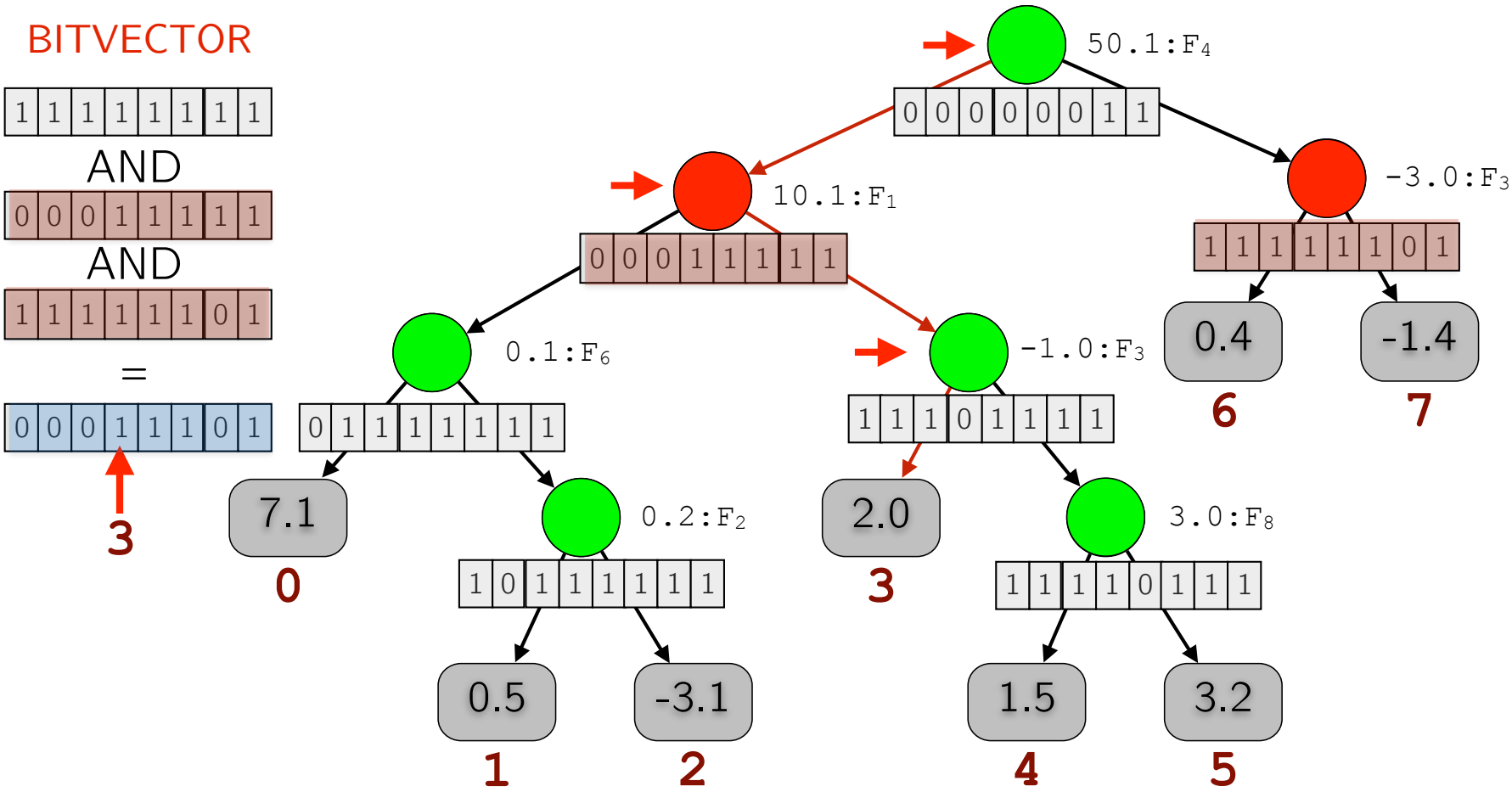
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: use of false nodes' masks



F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

QuickScore: use of false nodes' masks



F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55

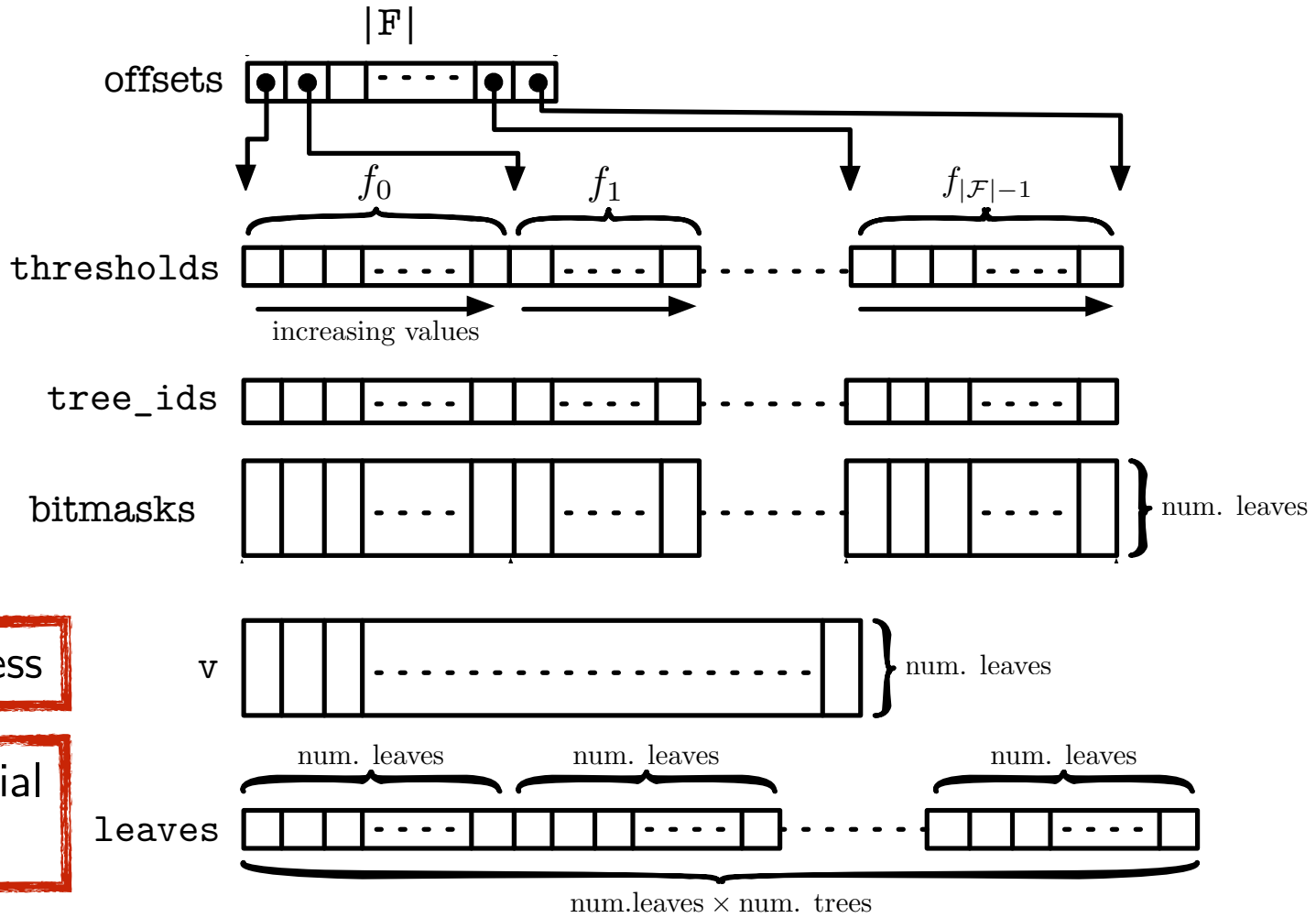
Few operations, insensitive to nodes' processing order!

QuickScore: data structures

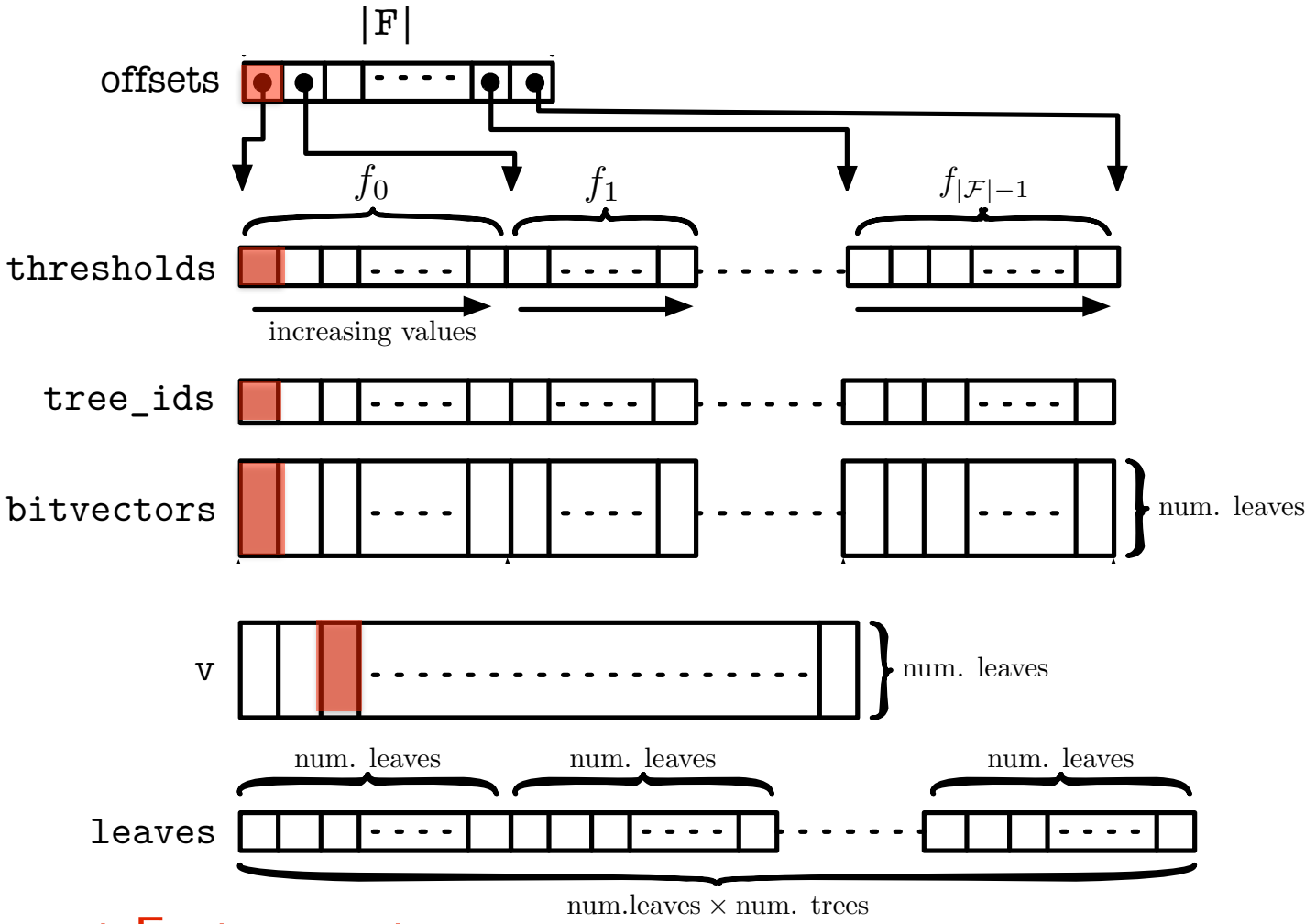
Read-only, sequential data access

R/W, random access

Read-only, sequential access



QuickScore: interleaved tree traversals

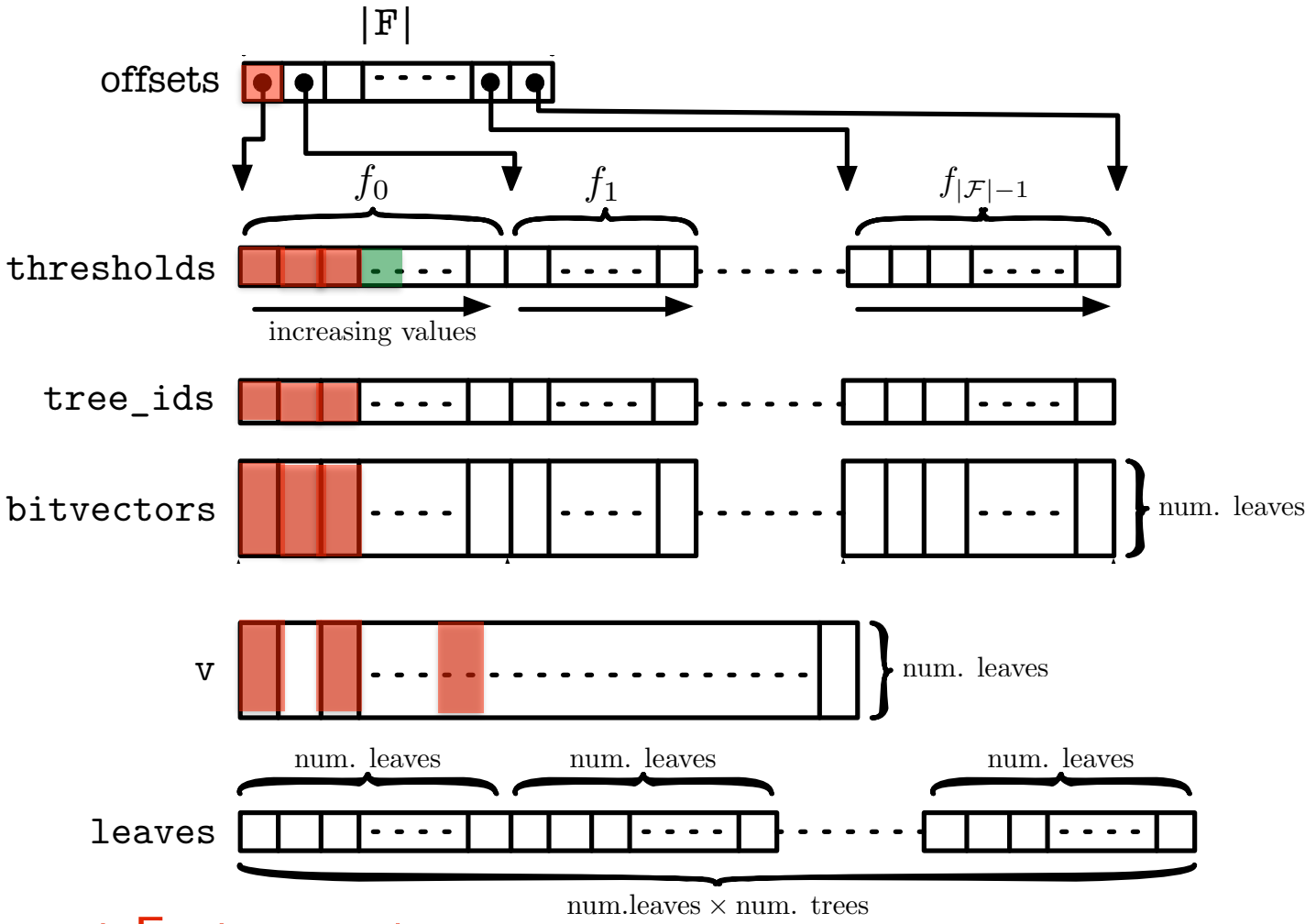


Query-Document Features sets

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

QuickScore: interleaved tree traversals

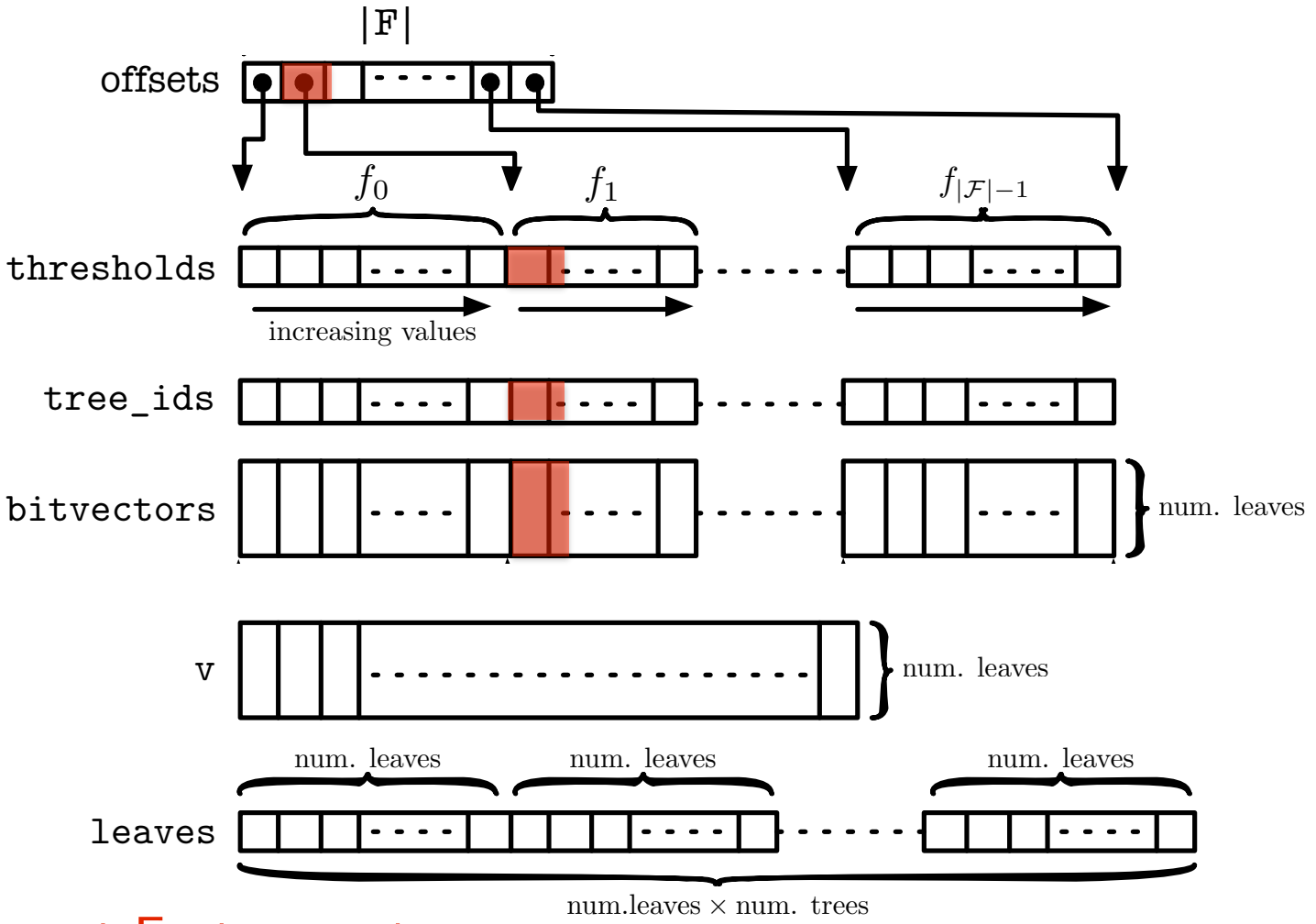


Query-Document Features sets

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

QuickScore: interleaved tree traversals

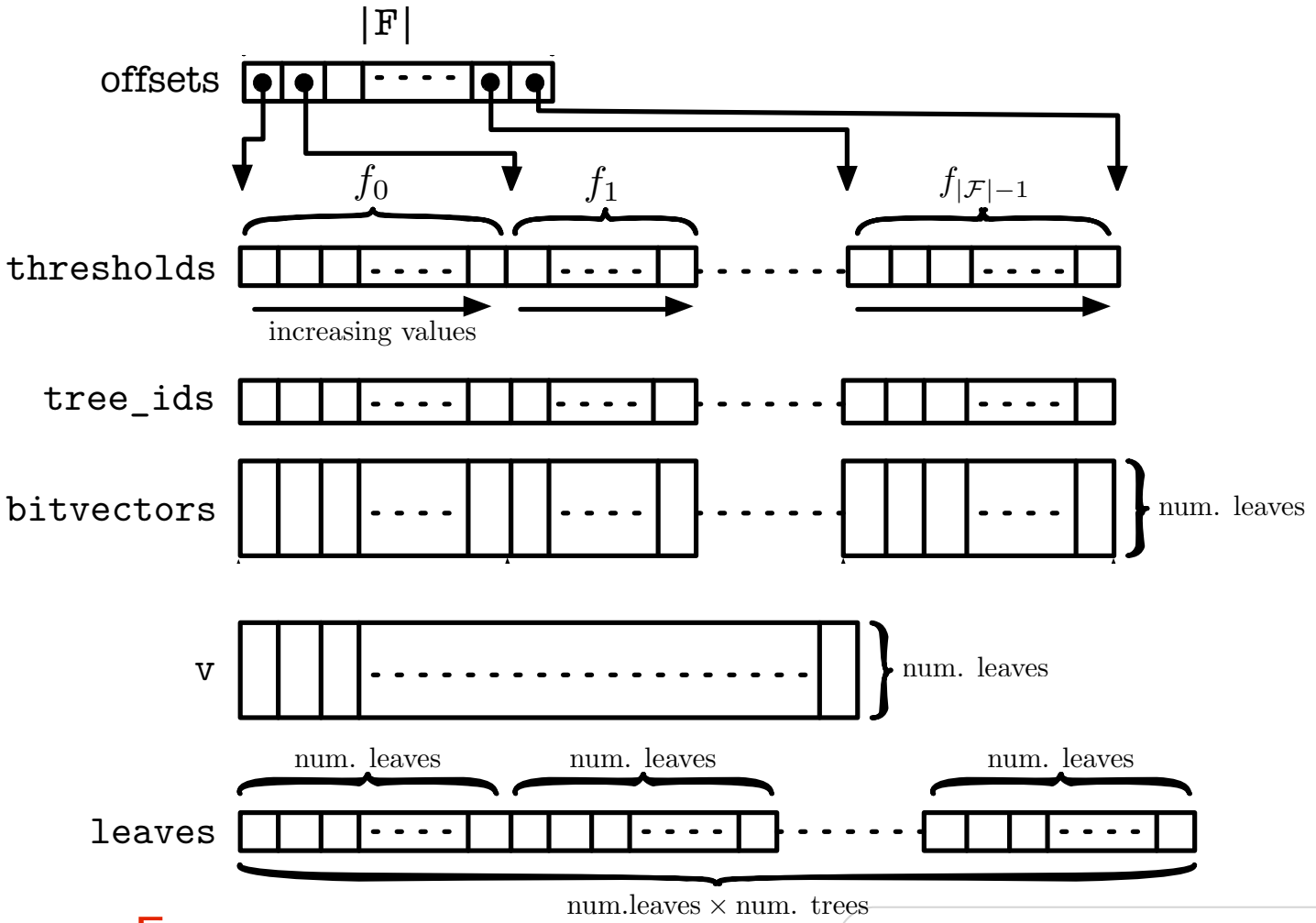


Query-Document Features sets

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

QuickScore: interleaved tree traversals



Query-Document Features sets

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

Low branch misprediction rate

High cache hit ratio

Experimental Settings

Lambda-MART ranking models optimizing NDCG@10 learned with RankLib from MSN and Yahoo LETOR datasets

Ensembles with 1K, 5K, 10K, or 20K regression trees, each with up to 8, 16, 32, or 64 leaves

Intel Core i7-4770K @ 3.50Ghz CPU, with 32GB RAM, Ubuntu Linux 3.13.0

Experimental Results

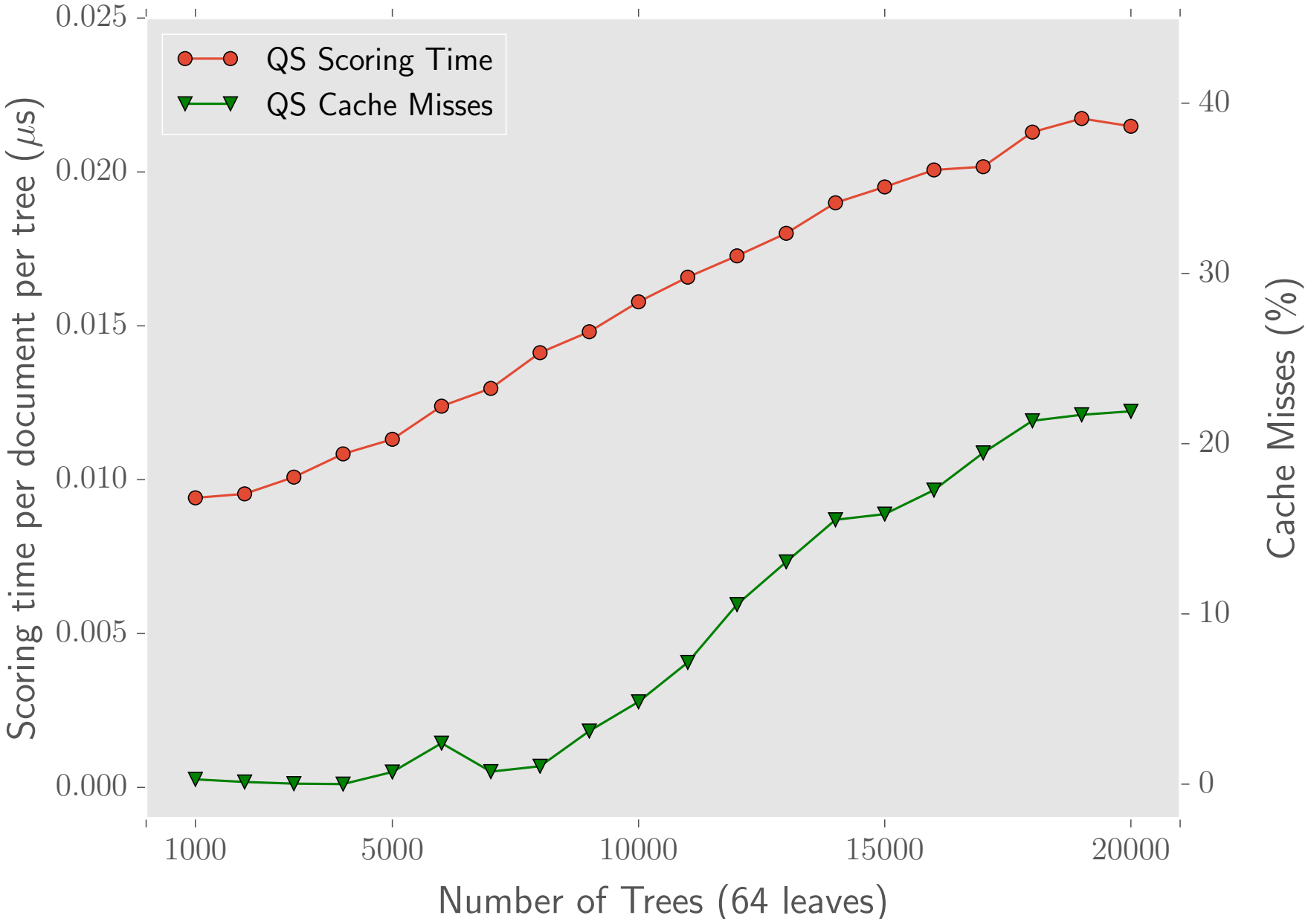
Per-document scoring time in microsecs and speedups

Method	Λ	Number of trees/dataset							
		1,000		5,000		10,000		20,000	
		MSN-1	Y!S1	MSN-1	Y!S1	MSN-1	Y!S1	MSN-1	Y!S1
QS	8	**2.2** (–)	**4.3** (–)	**10.5** (–)	**14.3** (–)	**20.0** (–)	**25.4** (–)	**40.5** (–)	**48.1** (–)
VPred		7.9 (3.6x)	8.5 (2.0x)	40.2 (3.8x)	41.6 (2.9x)	80.5 (4.0x)	82.7 (3.3)	161.4 (4.0x)	164.8 (3.4x)
If-Then-Else		8.2 (3.7x)	10.3 (2.4x)	81.0 (7.7x)	85.8 (6.0x)	185.1 (9.3x)	185.8 (7.3x)	709.0 (17.5x)	772.2 (16.0x)
Struct+		21.2 (9.6x)	23.1 (5.4x)	107.7 (10.3x)	112.6 (7.9x)	373.7 (18.7x)	390.8 (15.4x)	1150.4 (28.4x)	1141.6 (23.7x)
QS	16	**2.9** (–)	**6.1** (–)	**16.2** (–)	**22.2** (–)	**32.4** (–)	**41.2** (–)	**67.8** (–)	**81.0** (–)
VPred		16.0 (5.5x)	16.5 (2.7x)	82.4 (5.0x)	82.8 (3.7x)	165.5 (5.1x)	165.2 (4.0x)	336.4 (4.9x)	336.1 (4.1x)
If-Then-Else		18.0 (6.2x)	21.8 (3.6x)	126.9 (7.8x)	130.0 (5.8x)	617.8 (19.0x)	406.6 (9.9x)	1767.3 (26.0x)	1711.4 (21.1x)
Struct+		42.6 (14.7x)	41.0 (6.7x)	424.3 (26.2x)	403.9 (18.2x)	1218.6 (37.6x)	1191.3 (28.9x)	2590.8 (38.2x)	2621.2 (32.4x)
QS	32	**5.2** (–)	**9.7** (–)	**27.1** (–)	**34.3** (–)	**59.6** (–)	**70.3** (–)	**155.8** (–)	**160.1** (–)
VPred		31.9 (6.1x)	31.6 (3.2x)	165.2 (6.0x)	162.2 (4.7x)	343.4 (5.7x)	336.6 (4.8x)	711.9 (4.5x)	694.8 (4.3x)
If-Then-Else		34.5 (6.6x)	36.2 (3.7x)	300.9 (11.1x)	277.7 (8.0x)	1396.8 (23.4x)	1389.8 (19.8x)	3179.4 (20.4x)	3105.2 (19.4x)
Struct+		69.1 (13.3x)	67.4 (6.9x)	928.6 (34.2x)	834.6 (24.3x)	1806.7 (30.3x)	1774.3 (25.2x)	4610.8 (29.6x)	4332.3 (27.0x)
QS	64	**9.5** (–)	**15.1** (–)	**56.3** (–)	**66.9** (–)	**157.5** (–)	**159.4** (–)	**425.1** (–)	**343.7** (–)
VPred		62.2 (6.5x)	57.6 (3.8x)	355.2 (6.3x)	334.9 (5.0x)	734.4 (4.7x)	706.8 (4.4x)	1309.7 (3.0x)	1420.7 (4.1x)
If-Then-Else		55.9 (5.9x)	55.1 (3.6x)	933.1 (16.6x)	935.3 (14.0x)	2496.5 (15.9x)	2428.6 (15.2x)	4662.0 (11.0x)	4809.6 (14.0x)
Struct+		109.8 (11.6x)	116.8 (7.7x)	1661.7 (29.5x)	1554.6 (23.2x)	3040.7 (19.3x)	2937.3 (18.4x)	5437.0 (12.8x)	5456.4 (15.9x)

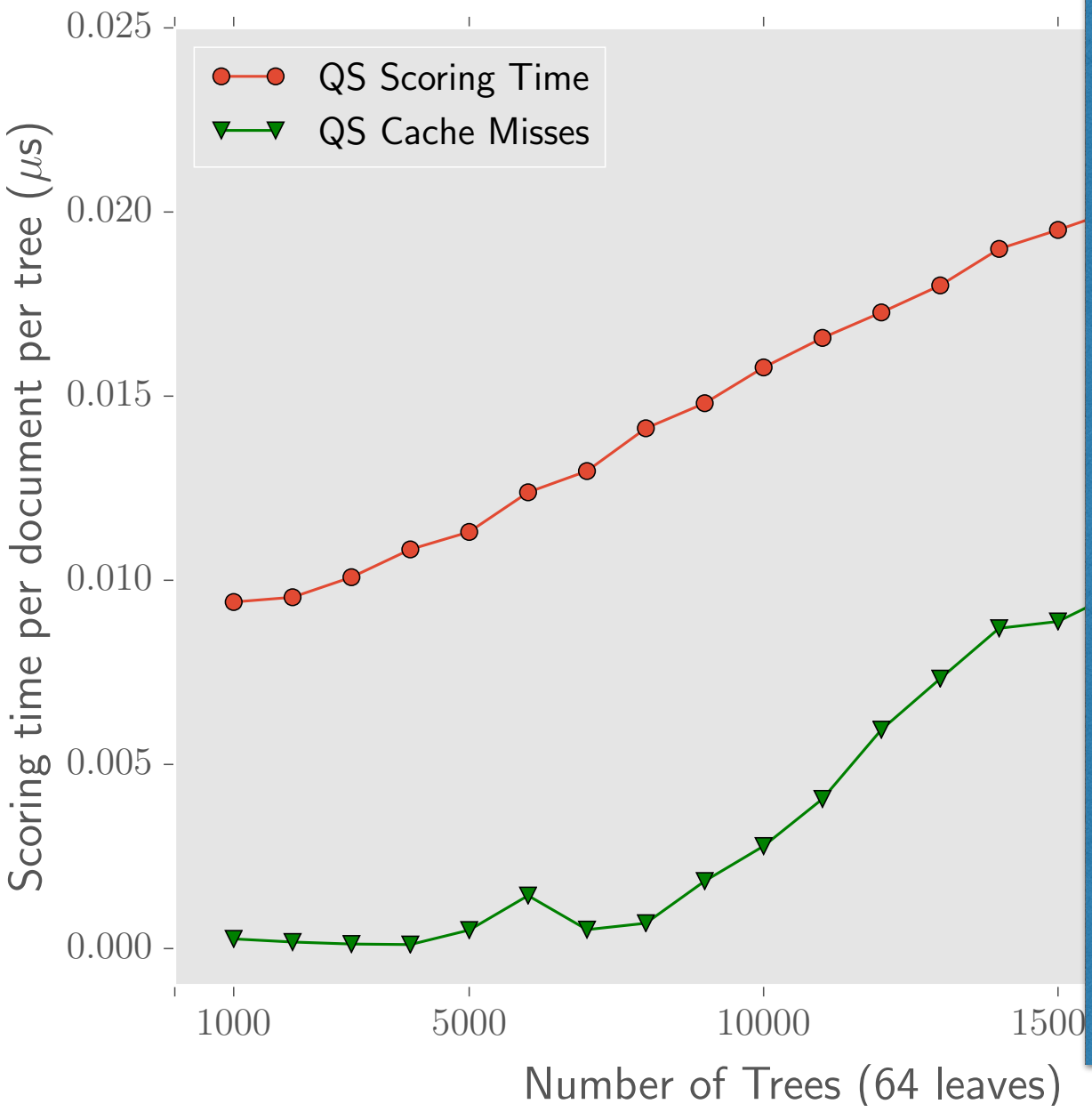
Per-tree per-document low-level statistics on MSN-1 with 64-leaves λ-MART models.

Method	Number of Trees				
	1,000	5,000	10,000	15,000	20,000
Instruction Count					
QS	**58**	**75**	**86**	**91**	**97**
VPred	580	599	594	588	516
If-Then-Else	142	139	133	130	116
Struct+	341	332	315	308	272
Num. Visited Nodes (above) Visited Nodes/Total Nodes (below)					
QS	**9.71** **15%**	**13.40** **21%**	**15.79** **25%**	**16.65** **26%**	**18.00** **29%**
VPred	54.38 86%	56.23 89%	55.79 89%	55.23 88%	48.45 77%
Struct+	40.61	39.29	37.16	36.15	31.75
If-Then-Else	64%	62%	59%	57%	50%

MSN-1: Scoring Time and Cache Misses

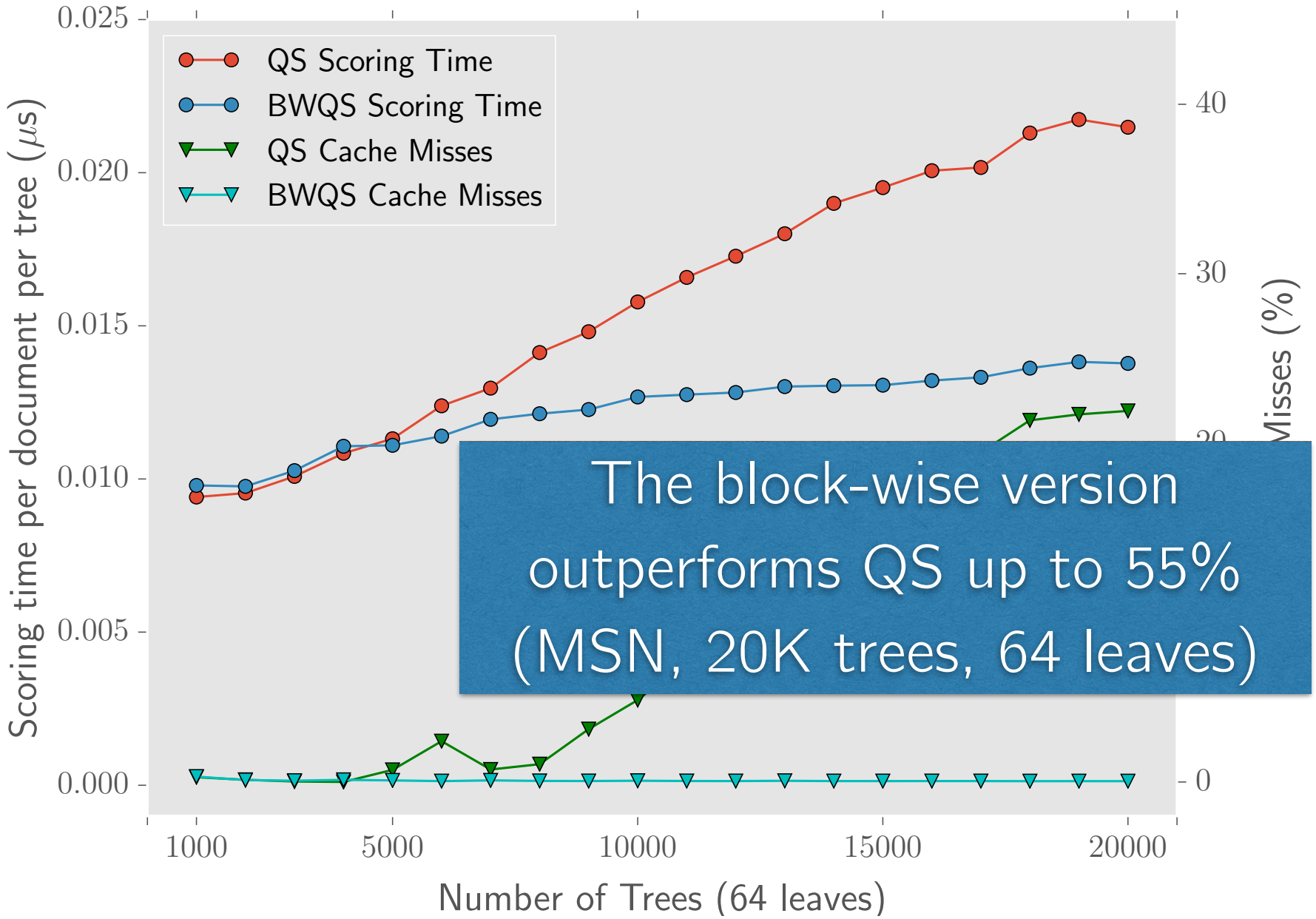


MSN-1: Scoring Time and Cache Misses



We can split the tree ensemble in disjoint blocks processed separately in order to let the corresponding data structures fit into the faster levels of the memory hierarchy.

MSN-1: Scoring Time and Cache Misses



Thank you!

National Research Council of Italy

HPC laboratory