Information Storage and Processing for Web Search

Nicola Tonellotto
ISTI-CNR
nicola.tonellotto@isti.cnr.it

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Università La Sapienza – 18 October 2016

Outline

- Some info about ISTI-CNR
- Introduction to Information Retrieval and Web Search
- Classical query processing
- Learning to Rank and query processing

CNR -> ISTI -> HPC

7 researchers

2 post-doc fellows

3 research associates

6 PhD students

Main Research Topics

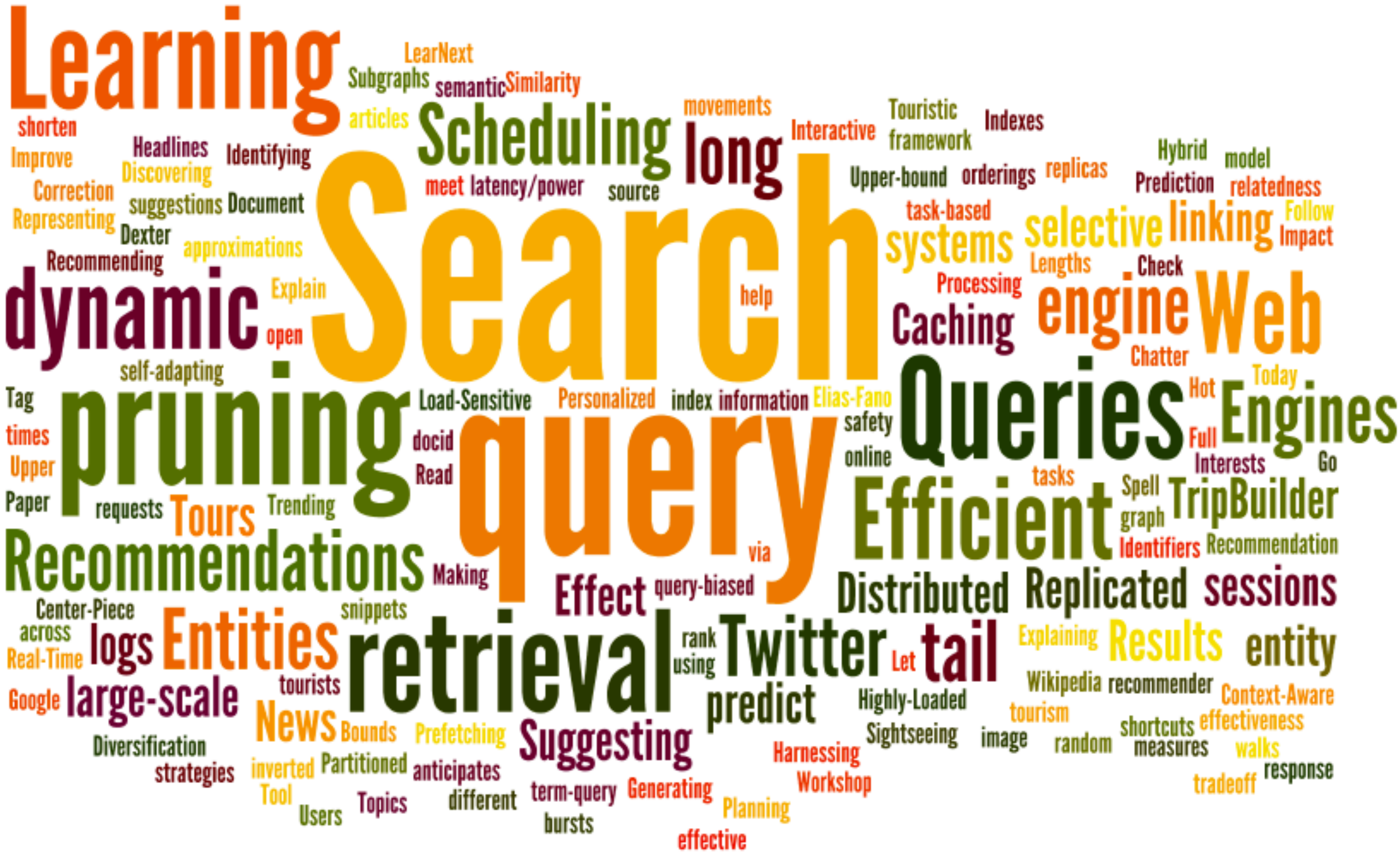
★ **Web Search & data mining**

- Responsiveness of large-scale search systems
- (Machine learned) ranking, prediction, recommendation, diversification
- Social media analysis
- Semantic Enrichment and Entity Linking
- Storage and Indexing of large amounts of data

★ **Cloud and Distributed computing**

- Cloud federations, Resource Management
- Network overlays for P2P and Big Data
- Scalable data analysis with Hadoop Map-Reduce, Giraph, Spark, etc

From paper titles 2011-2013



Our Strengths

- Highly-motivated group of (mostly 😃) young researchers
- Papers accepted at all the main top conferences on Web IR & DM
- Attractive to HQ students, former PhDs now at **Twitter**, **Facebook**, **Yahoo!** and **Tiscali**
- Good portfolio of EC projects, good international and national connections with academia and industry

Products / Achievements

Production Systems

Learning to Rank for Tiscali istella, feature tuning, near-duplicate detection, massive Hadoop MapReduce computations

Learning to Rank Metadata Records for Europeana, Entity suggestion

Research Prototypes

Framework for implementing and evaluating entity linking algorithms. http://dexter.isti.cnr.it

Fast and Scalable Learning to Rank with QuickRank http://quickrank.isti.cnr.it

Budgeted Sightseeing Tours Planning exloiting Social Media http://tripbuilder.isti.cnr.it

Collaboration with istella

Information Retrieval

- **Information Retrieval** (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

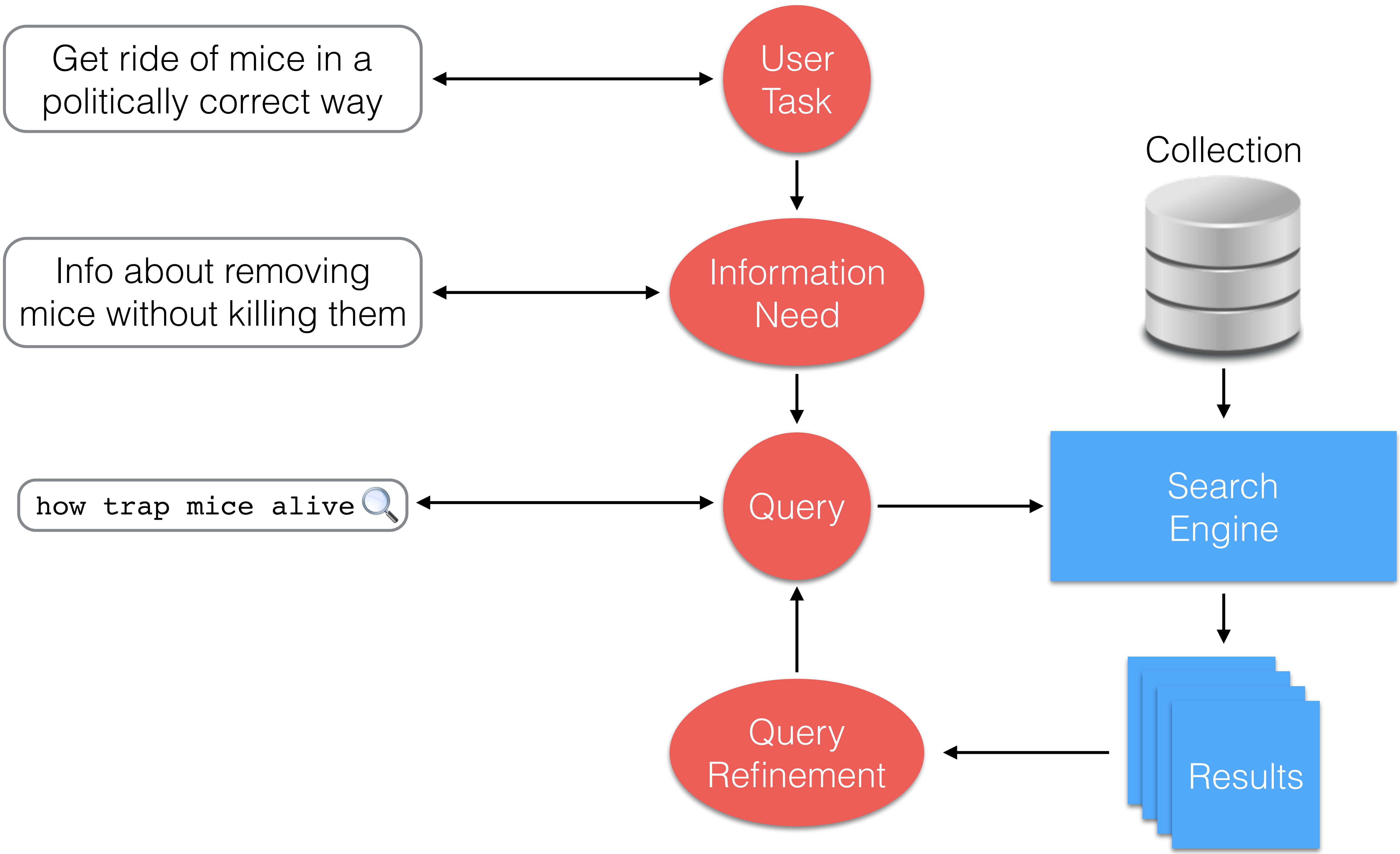
- These days we frequently think first of **Web Search**, but there are many other cases:
    - E-mail search
    - Searching your laptop
    - Corporate knowledge bases
    - Legal information retrieval
    - Patent Retrieval
    - Medical Retrieval

Basic Assumptions

- **Collection**: A set of textual documents

- **Goal**: Retrieve documents with information that is **relevant** to the user's **information need** and helps the user complete a **task**

Classical IR Model



Search in 1620

- Which plays of Shakespeare contain the words **Brutus AND Caesar** but **NOT Calpurnia**?
- One could **grep** all of Shakespeare's plays for **Brutus** and **Caesar**, then strip out lines containing **Calpurnia**?
- Why is that not the answer?
  - Slow (for large corpora)
  - NOT Calpurnia is non-trivial
  - Other operations (e.g., find the word Romans near countrymen) not feasible
  - Ranked retrieval (only best documents to return)

Term-Document Incidence Matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Brutus AND Caesar BUT NOT Calpurnia

1 if **play** contains **word**
0 otherwise

Incidence Vectors

- So we have a 0/1 vector for each term.
- To answer query: take the vectors for **Brutus**, **Caesar** and **Calpurnia** (complemented) and perform **bitwise AND**.

110100 AND
110111 AND
101111 =
100100

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Bigger Collections

- Consider N = **1 million documents**, each with about **1000 words**.
- Average **6 bytes/word** including spaces/punctuation
- **6GB of data** in the documents.
- Say there are M = **500K distinct terms** among these.
- 500K x 1M matrix has **0.5T** 0's and 1's.
- But it has no more than one **1G** 1's.
- Matrix is **extremely sparse**.
- What's a better representation?

Bigger Collections

- Consider N = **1 million documents**, each with about **1000 words**.
- Average **6 bytes/word** including spaces/punctuation
- **6GB of data** in the documents.
- Say there are M = **500K distinct terms** among these.
- 500K x 1M matrix has **0.5T** 0's and 1's.
- But it has no more than one **1G** 1's.
- Matrix is **extremely sparse**.
- What's a better representation?

★ **We only record the 1's positions.**

Inverted Index

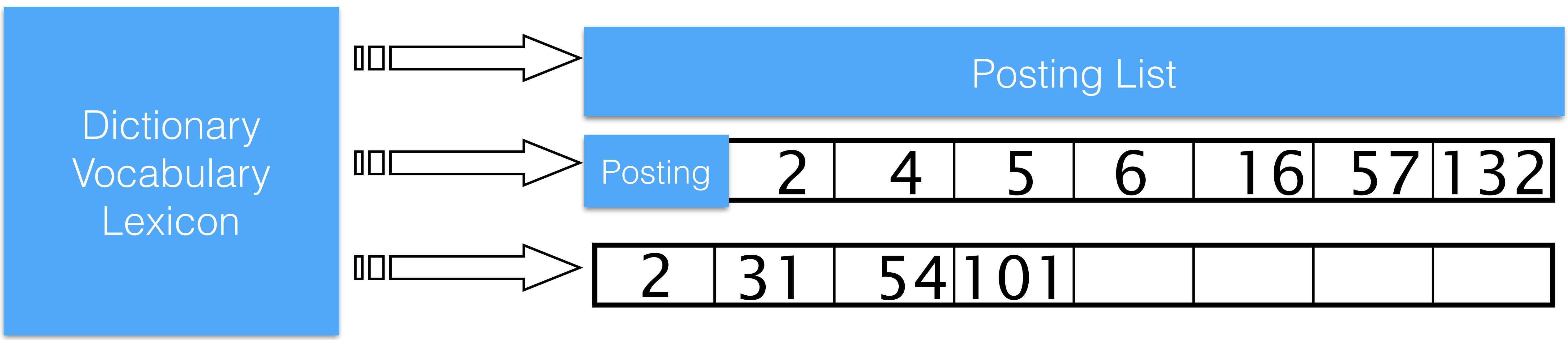
- For each **term t**, we must store a **list of all documents that contain t**.
- Identify each doc by a **docid**, a document serial number.

Brutus →	1	2	4	11	31	45	173	174
Caesar →	1	2	4	5	6	16	57	132
Calpurnia →	2	31	54	101				

- Can we used fixed-size arrays for this?
- We need variable-size **posting lists**.

Inverted Index

- For each **term t**, we must store a **list of all documents that contain t**.
- Identify each doc by a **docid**, a document serial number.



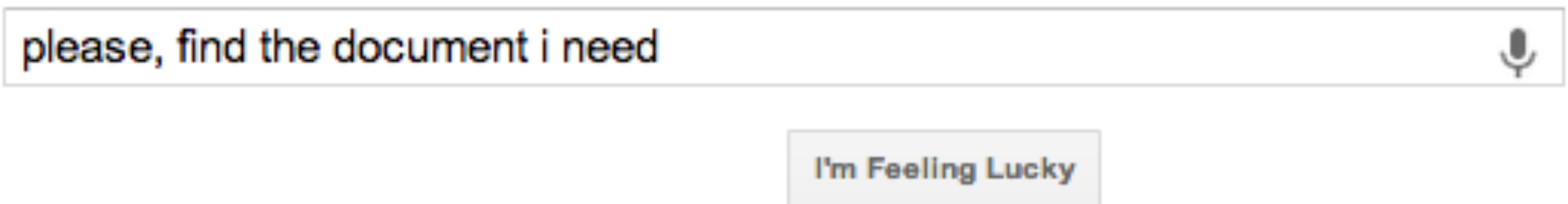
- Can we used fixed-size arrays for this?
- We need variable-size **posting lists**.

Boolean queries: exact match

- The **Boolean retrieval model** is being able to ask a query that is a Boolean expression:
  - Boolean queries are queries **using AND, OR and NOT** to join query terms
  - Views each document as a **set of words**
  - Is **precise**: document matches condition or not.
- Perhaps the **simplest model** to build an IR system on
- Primary commercial retrieval tool for **3 decades**.
- Many search systems you still use are boolean:
  - Email, library catalog, Mac OS X Spotlight

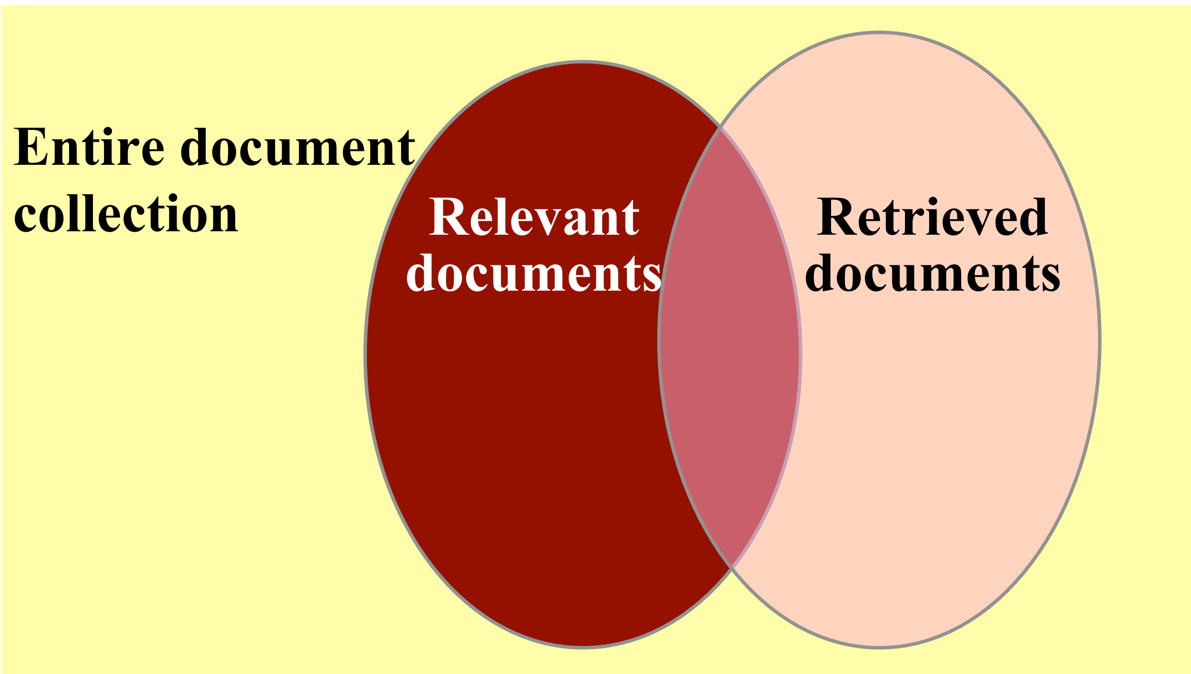
Ranked Retrieval

- **Ranking** is (one of) the most important challenges in Web Search



- We define Ranking as the problem of sorting a set of documents according to their **relevance** to the user query.

Precision and Recall



	retrieved	not retrieved
irrelevant	retrieved & irrelevant	Not retrieved & irrelevant
relevant	retrieved & relevant	not retrieved but relevant

$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

Precision and Recall

- Rather than evaluating the full list of documents, look only at the top k:

  P@k

  R@k

- There are more advanced measures:

  MAP@k

  Mean Average Precision

  NDCG@k

  Normalized Discounted Cumulative Gain

BM25

- BM25 is a **probabilistic model**: using term independence assumption to approximate the document **probability of being relevant**

$$\mathrm{BM25}(d, q) = \sum_t \mathrm{IDF}_t \, \tau(F_t)$$

- $\mathrm{IDF}_t = \log(N/n_t)$ is the **inverse document frequency**
    - N is the number of docs in the collection
    - n_t is the number of docs containing t
- Frequent terms are not very specific, and their contribution is reduced

BM25

$$\mathrm{BM25}(d,q) = \sum_t \mathrm{IDF}_t\, \tau(F_t)$$

$$F_t = \frac{f_{t,d}}{1 - b + b \cdot l_d / L} \qquad \tau(F_t) = \frac{F_t}{k + F_t}$$

- **$f_{t,d}$** is the frequency of term t in document d
- **l_d** is the length of document d
    - longer documents are less important
- **L** is the average document length in the collection
- **b** determines the importance of l_d
- $\tau()$ is a **smoothing function**, modelling non-linearity of terms contribution

Query Processing Breakdown

- Pre-process the query (e.g., tokenisation, stemming)
- Lookup the statistics for each term in the lexicon
- Process the postings for each query term, computing scores for documents to identify the final retrieved set
- Output the retrieved set with metadata (e.g., URLs)

Query Processing Breakdown

- Pre-process the query (e.g., tokenisation, stemming)
- Lookup the statistics for each term in the lexicon
- Process the postings for each query term, computing scores for documents to identify the final retrieved set
- Output the retrieved set with metadata (e.g., URLs)

Document-at-a-Time (DAAT)

Input: The index I of the collection
The query q
Output: A set D containing the top K ranked documents

Allocate an empty list L
For each term t in q
    Scan index I and grab posting list I_t
    Calculate $w_q(t)$
While all posting lists I_t have postings
    Set $d \leftarrow \infty$
    For each posting list I_t
        If current document id in I_t < d
            $d \leftarrow$ current document id in I_t
    Set $A \leftarrow 0$
    For each posting list I_t
        If current document id in I_t = d
            $A \leftarrow A + w_q(t) \cdot w_q(d, t)$
            Move I_t one step ahead to the next document id
    If $A \neq 0$
        Insert A in L
Select the K highest scores in L and put the corresponding docids in D

Dynamic Pruning

- What takes time?
  - Number of query terms
    - Longer queries have more terms with posting lists to process
  - Length of posting lists
    - More postings takes longer times

- Aim: avoid (unnecessary) scoring of posting

Safeness

- **Safe pruning**: the output ordering of the strategy is identical to the output ordering of the full processing

- **Safe up to rank K**: the first K documents are identical to the first K documents of the full processing

- **Approximate**: no guarantees on final ordering of document w.r.t. full processing

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Safeness

- **Safe pruning**: the output ordering of the strategy is identical to the output ordering of the full processing

- **Safe up to rank K**: the first K documents are identical to the first K documents of the full processing

- **Approximate**: no guarantees on final ordering of document w.r.t. full processing

DAAT Pruning

- **MaxScore** (Turtle & Flood, IPM 31(6), 1995)
  - *Early termination*: does not compute scores for documents that won't be retrieved
  - By comparing upper bounds with threshold
  - Suitable for TAAT as well
- **WAND** (Broder et al., CIKM 2003)
  - *Approximate evaluation*: does not consider documents with approximate scores (sum of upper bounds) lower than threshold
  - Exploit skipping
- **BlockMaxWAND** (Ding & Suel, SIGIR 2011)
  - Two levels: initially on blocks (128 postings), then on postings
  - *Approximate evaluation*: does not consider documents with approximate scores (sum of upper bounds) lower than threshold
  - Exploit skipping
- All three use docids sorted posting lists

Some (Unpublished) Results (50 M docs)

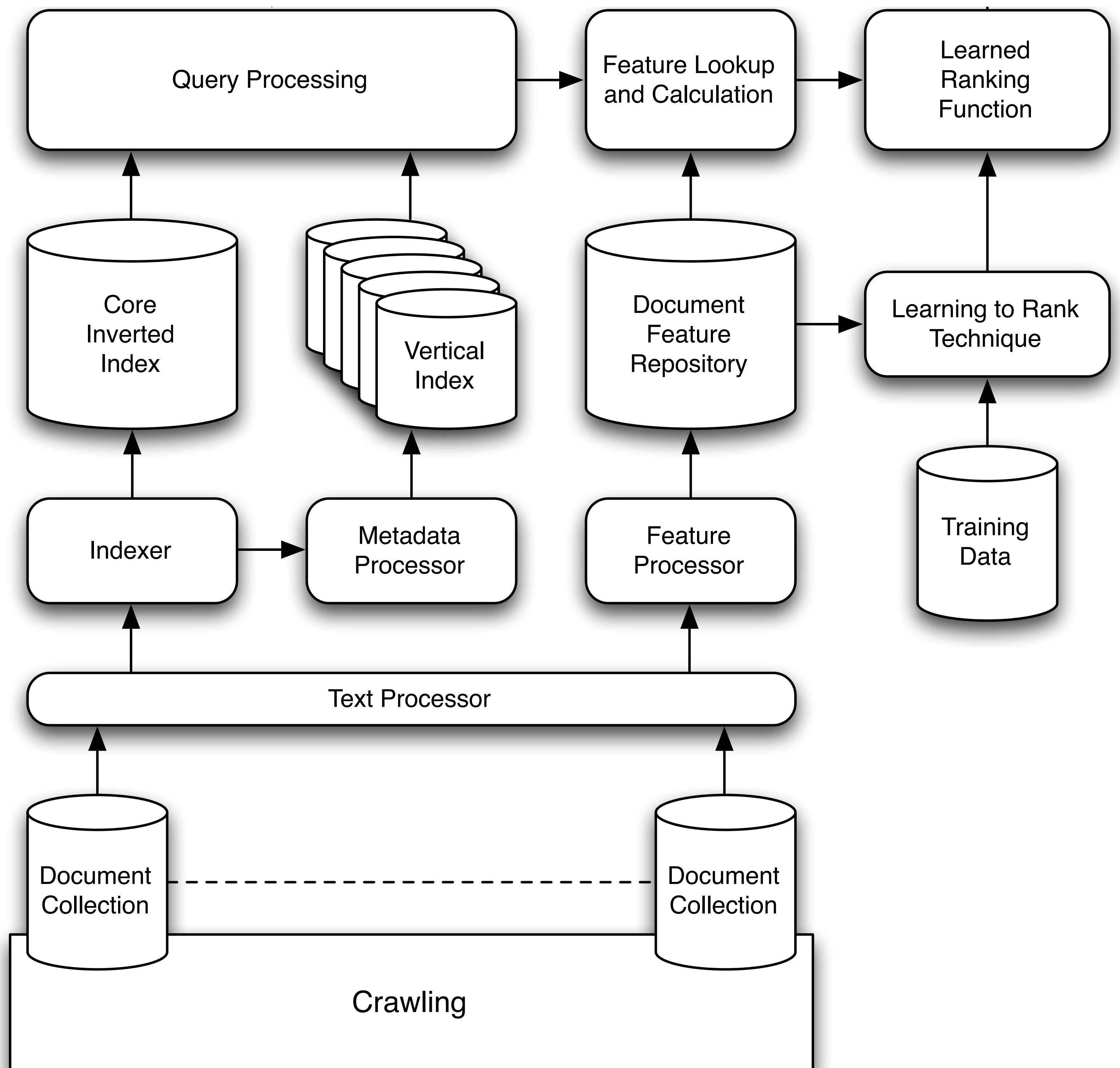
Ranking Algorithm	Num Terms						
	1	2	3	4	5	6	7
Ranked And	43.59	38.08	**32.68**	**25.23**	**29.26**	**17.57**	**15.78**
Ranked Or	43.05	261.59	536.06	807.05	1,107.93	1,402.26	1,756.52
MaxScore	45.01	48.21	51.06	57.28	75.66	**95.06**	**117.61**
Wand	62.62	44.98	48.24	**55.27**	**69.39**	98.47	120.70
BlockMaxWand	**0.71**	**13.11**	**40.69**	64.62	99.95	149.94	192.65

Average Response Times (msec)

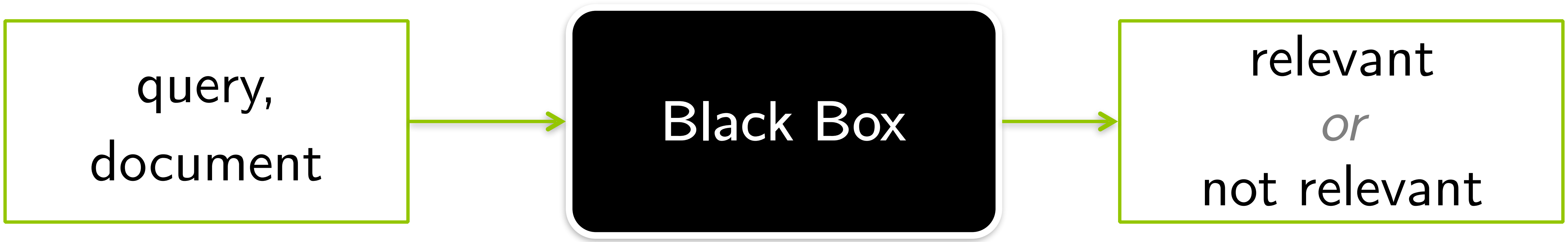
Ranking Algorithm	Num Terms						
	1	2	3	4	5	6	7
Ranked And	265.25	182.84	**136.33**	**101.75**	**94.99**	**66.70**	**61.26**
Ranked Or	260.74	838.04	1,296.98	1,759.49	2,209.58	2,663.02	3,130.37
MaxScore	245.03	189.39	174.92	**175.44**	215.57	**253.29**	**313.25**
Wand	387.55	210.37	184.10	182.05	**210.05**	289.27	337.07
BlockMaxWand	**1.64**	**43.05**	**140.12**	201.39	280.05	394.48	500.16

95% Response Time (msec)

Web Search Engine



Learning to Rank is not classification

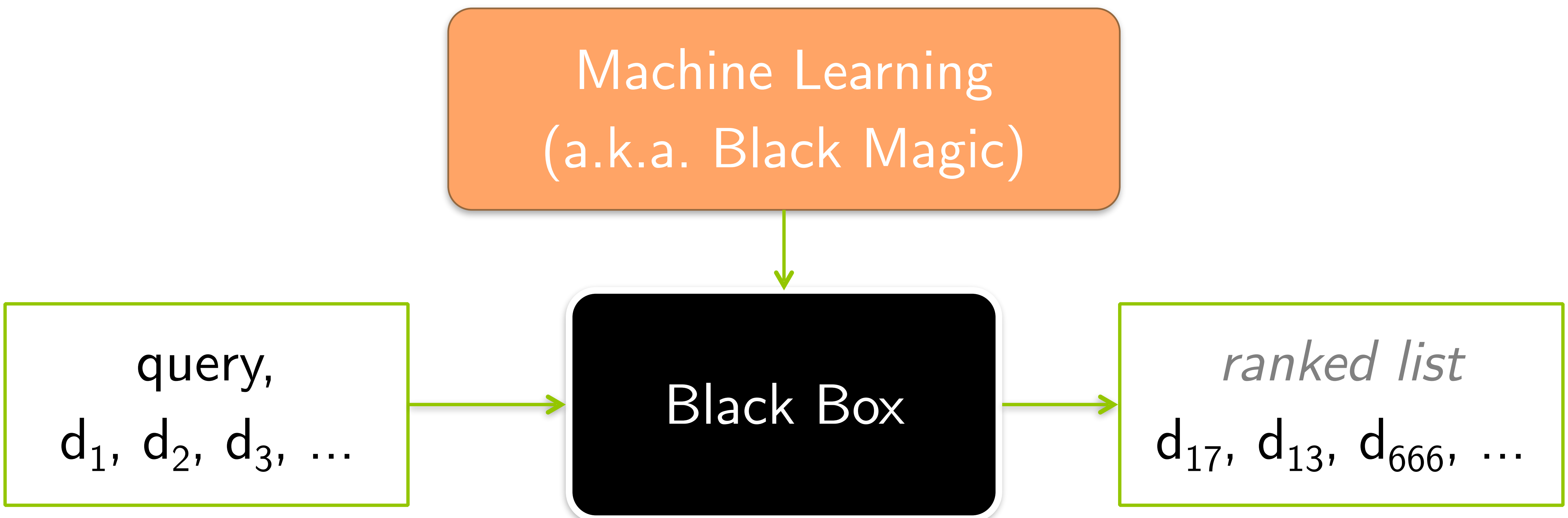


ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Learning to Rank is:

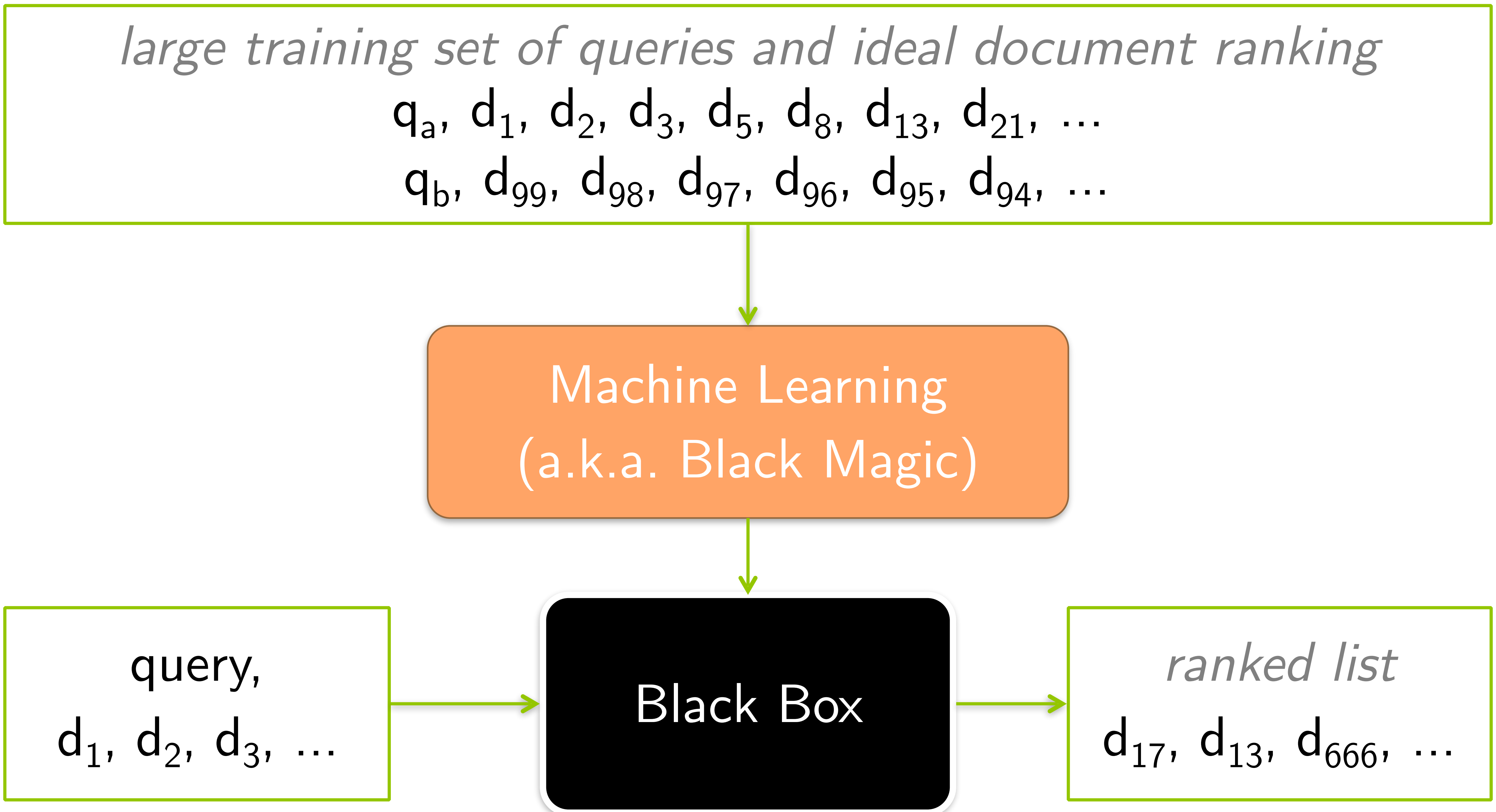
The goal is to learn the ranking, not the label !

Learning to Rank is:



The goal is to learn the ranking, not the label !

Learning to Rank is:



The goal is to learn the ranking, not the label !

Normalized Discounted Cumulative Gain
NDCG@K

- Consider only the top-K ranked documents, and sum up (cumulate) their contribution

- The contribution (gain) of a result depends on its relevance label

- Contribution is diminished (discounted) if the result is in the "bottom" positions

- Normalize between 0 and 1

$$\mathrm{DCG@k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log(i+1)} \quad \mathrm{NDCG@k} = \frac{\mathrm{DCG@k}}{\mathrm{IDCG@k}}$$

- rel_i is the relevance label of the *i-th* result (e.g., 1..5)
- IDCG@k is the score of the ideal ranking

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Learning to Rank Approaches

- Pointwise
  - Each query-document pair is associated with a score
  - The objective is to predict such score
    - Can be considered a regression problem
  - Does not consider the position of a document into the result list
- Pairwise
  - We are given pairwise preferences, d1 is better than d2 for query q
  - The objective is to predict a score that preserves such preferences
    - Can be considered a classification problem
  - It partially considers the position of a document into the result list
- Listwise
  - We are given the ideal ranking of results for each query
    - NB: It might not be trivial to produce such training set
  - Objective maximize the quality of the resulting ranked list
    - We need some improved approach...

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Decision Tree

- Tree-like structure similar to a flow chart.
- Every internal node denotes a test over an attribute/feature
  - Outgoing edges correspond to the test possible outcomes
- Every leaf node is associated to a class label (if it is a classification task) or class distribution or predicted value (if it is a regression task)

- It is used to label a new data instance on the basis of its attributes
- Runs tests on the data instance attributes and traverses the tree according to the tests results
  - Starting form the root, the data instances follows a path to a leaf
  - The label associated to the leaf is the prediction of the decision tree.

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

(Basic) Boosted Decision Trees

(Basic) Boosted Decision Trees

- We want to learn a predictor incrementally:

$$F^*(x) = \sum_{m=0}^{M} f_m(x)$$

- Input: a learning sample $\{ (x_i, y_i) : i = 1, \ldots, N \}$

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

(Basic) Boosted Decision Trees

- We want to learn a predictor incrementally:

$$F^*(x) = \sum_{m=0}^{M} f_m(x)$$

- Input: a learning sample $\{ (x_i, y_i) : i = 1, \ldots, N \}$
- Initialize
  - Baseline tree predicts the average label value

$$\hat{y}_0(x) = 1/N \textstyle\sum_i y_i$$
$$r_i = y_i,\ i = 1, \ldots, N$$

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

(Basic) Boosted Decision Trees

- We want to learn a predictor incrementally:

$$F^*(x) = \sum_{m=0}^{M} f_m(x)$$

- Input: a learning sample $\{ (x_i, y_i): i = 1, \ldots, N \}$
- Initialize
  - Baseline tree predicts the average label value

$$\hat{y}_0(x) = 1/N \textstyle\sum_i y_i$$
$$r_i = y_i,\ i = 1, \ldots, N$$

- For $t = 1$ to M:
  - Regression tree predicts the residual error
  - For $i = 1$ to N, compute the residuals

$$r_i \leftarrow r_{i-1} - \hat{y}_{t-1}(x_i)$$

  - Build a regression tree from the learning sample $\{ (x_i, y_i): i = 1, \ldots, N \}$
  - The prediction of the new regression tree is denoted with $\hat{y}_t$

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

(Basic) Boosted Decision Trees

- We want to learn a predictor incrementally:

$$F^*(x) = \sum_{m=0}^{M} f_m(x)$$

- Input: a learning sample $\{ (x_i, y_i) : i = 1, \ldots, N \}$
- Initialize
  - Baseline tree predicts the average label value

$$\hat{y}_0(x) = 1/N \sum_i y_i$$
$$r_i = y_i,\ i = 1, \ldots, N$$

- For t = 1 to M:
  - Regression tree predicts the residual error
  - For i = 1 to N, compute the residuals

$$r_i \leftarrow r_{i-1} - \hat{y}_{t-1}(x_i)$$

  - Build a regression tree from the learning sample $\{ (x_i, y_i) : i = 1, \ldots, N \}$
  - The prediction of the new regression tree is denoted with $\hat{y}_t$

- Return the model $\hat{y}(x) = \hat{y}_0(x) + \hat{y}_1(x) + \ldots + \hat{y}_M(x)$

(Basic) Boosted Decision Trees

- We want to learn a predictor incrementally:

$$F^*(x) = \sum_{m=0}^{M} f_m(x)$$

- Input: a learning sample $\{ (x_i,y_i): i = 1, \ldots, N \}$
- Initialize
  - Baseline tree predicts the average label value

$$\hat{y}_0(x) = 1/N \textstyle\sum_i y_i$$
$$r_i = y_i,\ i = 1, \ldots, N$$

- For t = 1 to M:
  - Regression tree predicts the residual error
  - For i = 1 to N, compute the residuals

$$r_i \leftarrow r_{i-1} - \hat{y}_{t-1}(x_i)$$

  - Build a regression tree from the learning sample $\{ (x_i,y_i): i = 1, \ldots, N \}$
  - The prediction of the new regression tree is denoted with $\hat{y}_t$

- Return the model $\hat{y}(x) = \hat{y}_0(x) + \hat{y}_1(x) + \ldots + \hat{y}_M(x)$

- Function f_m should be easy to be learnt:
  - Decision stump: trees with one node and two leaves

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

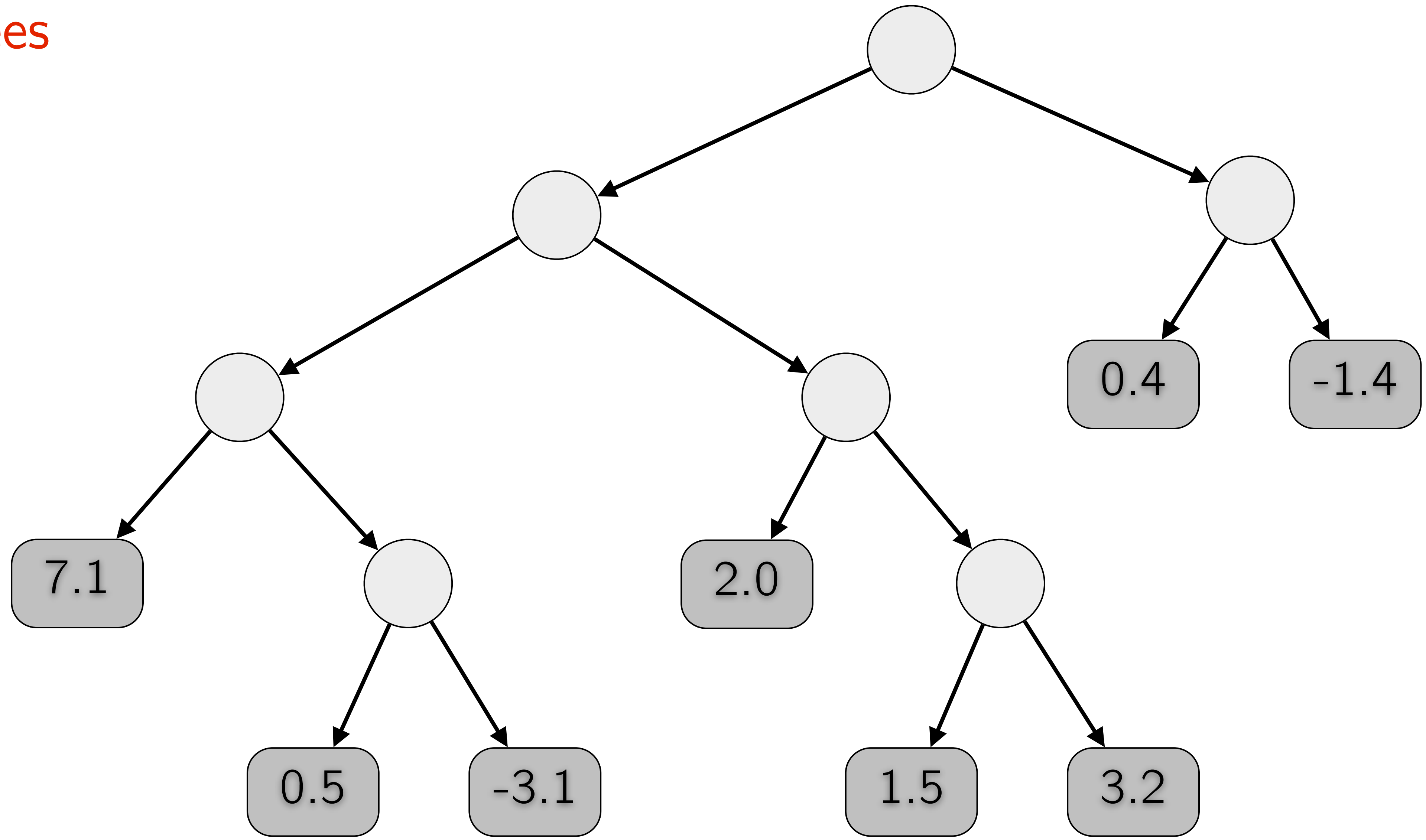
ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Documents

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

...

Trees



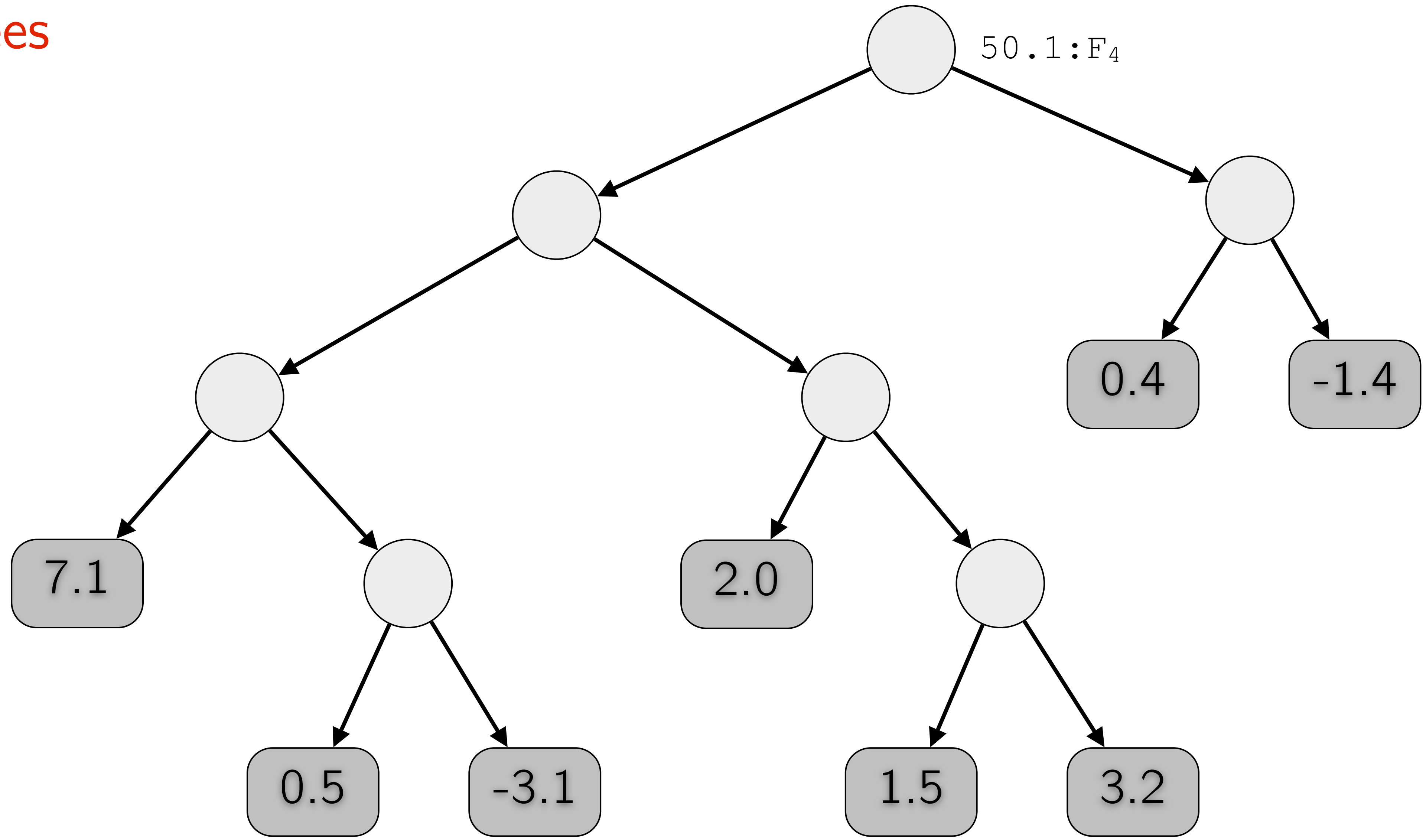
Documents

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

...

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Trees

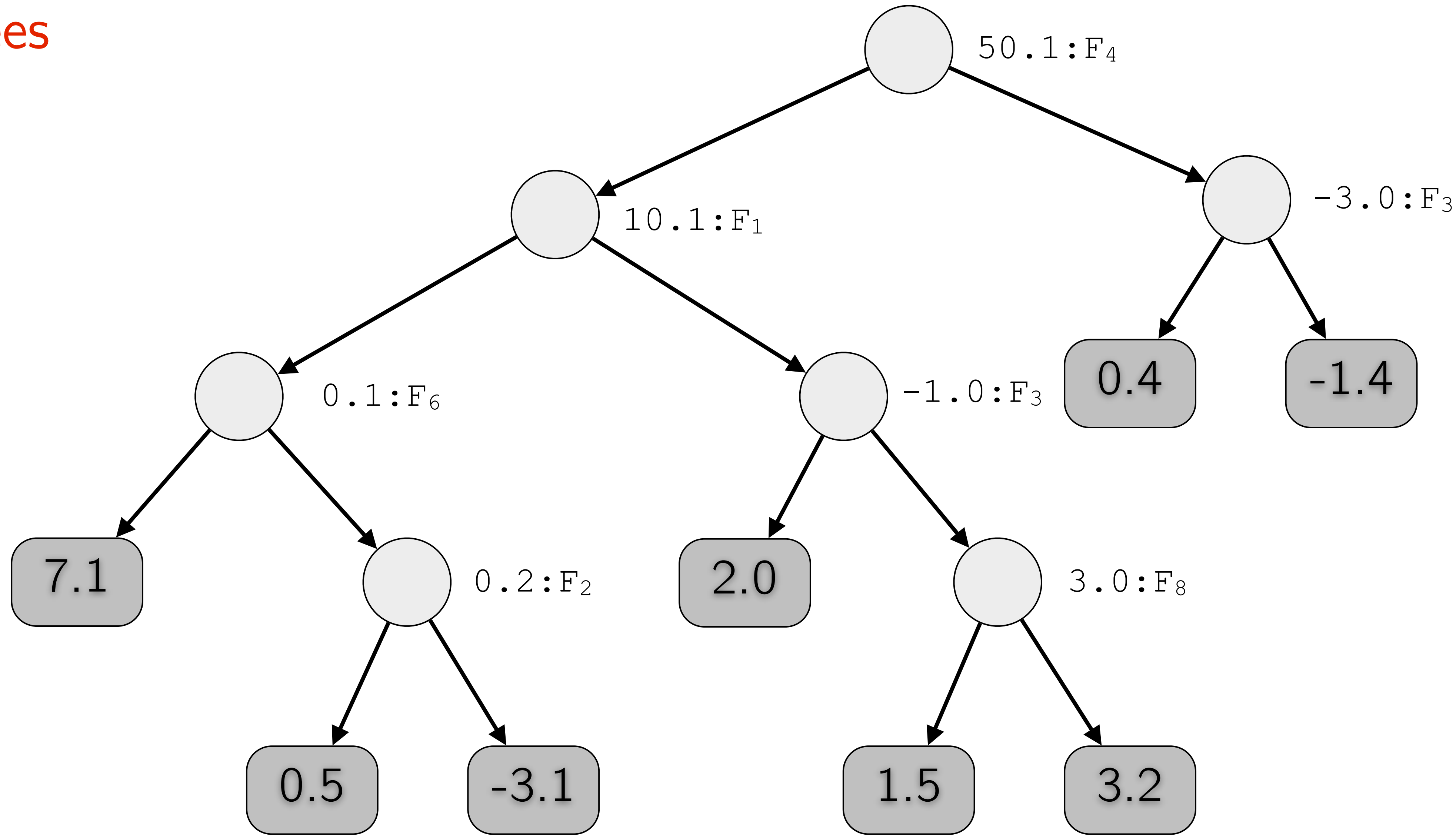


Documents

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

...

Trees

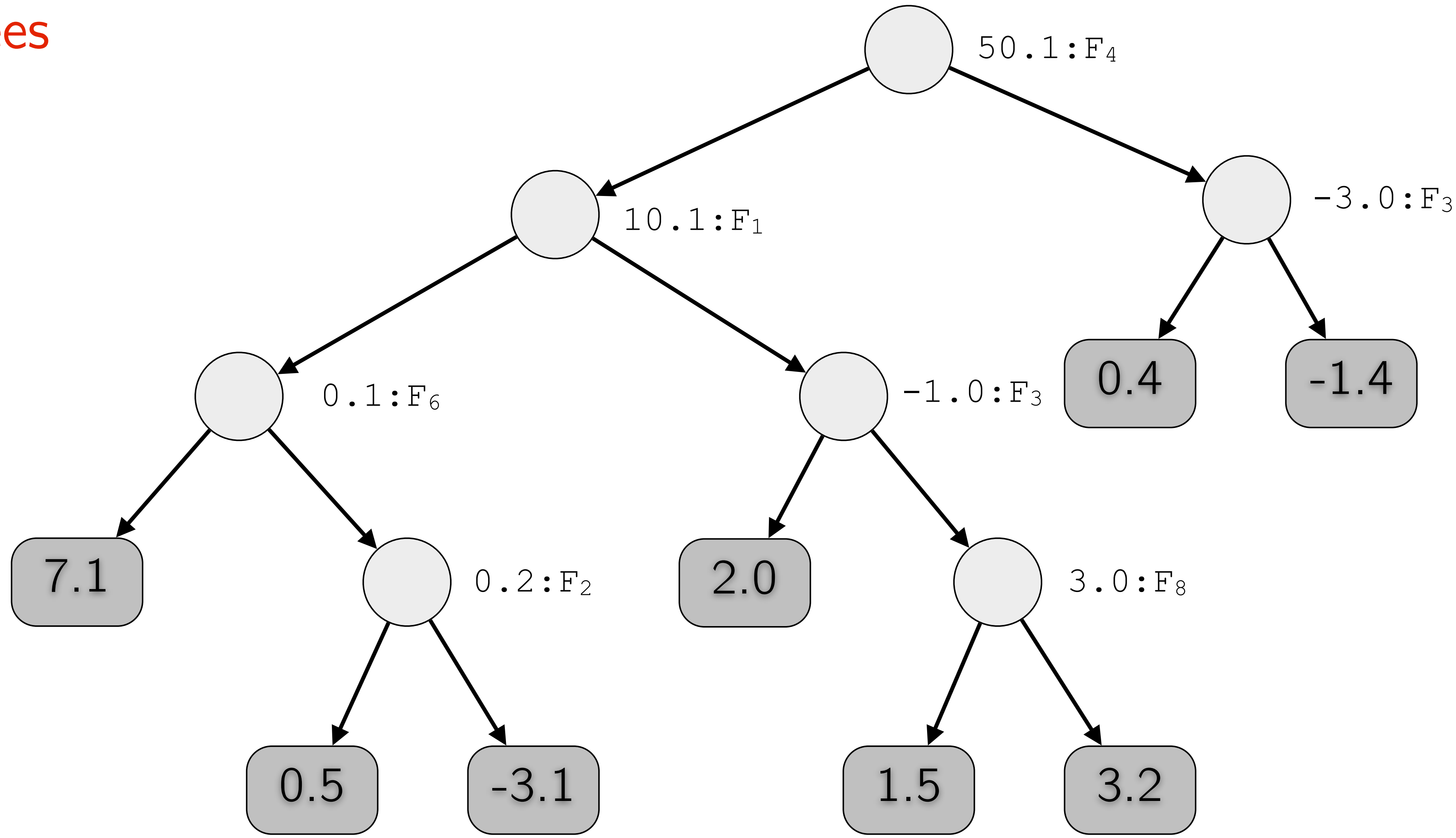


Documents

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

...

Trees

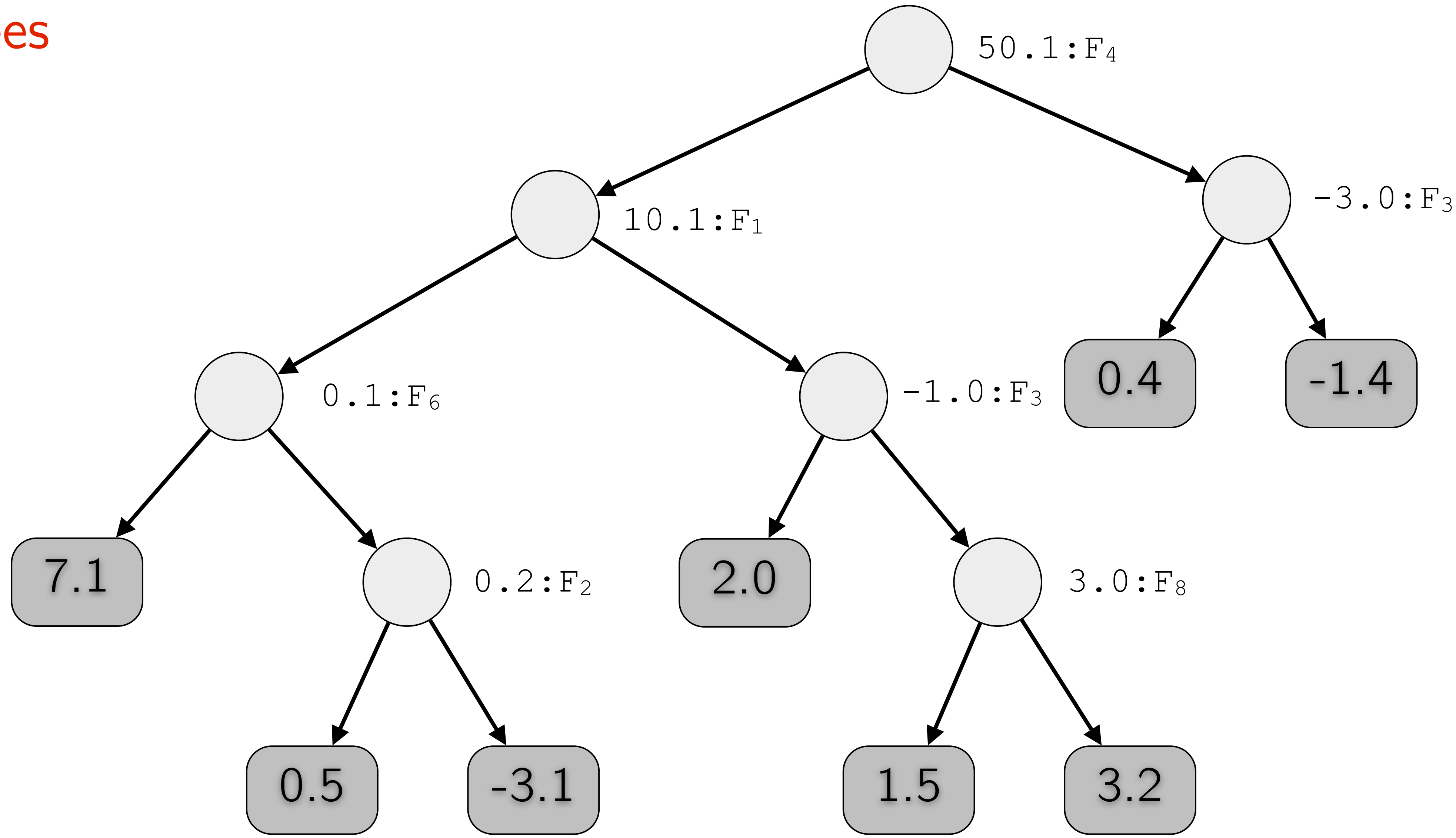


Documents

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

...

Trees



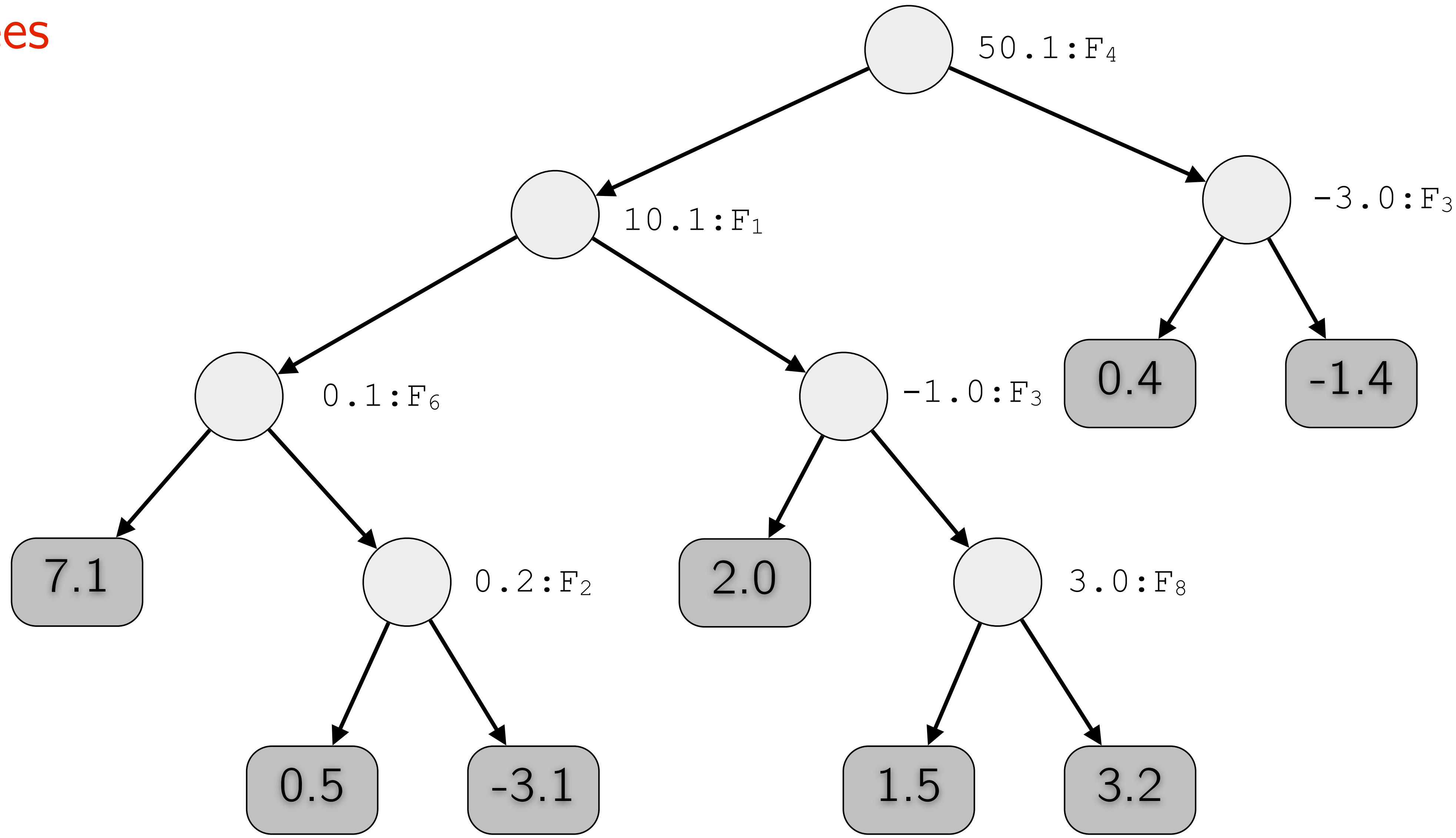
Documents

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

...

\# docs = >100K

Trees



Documents

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

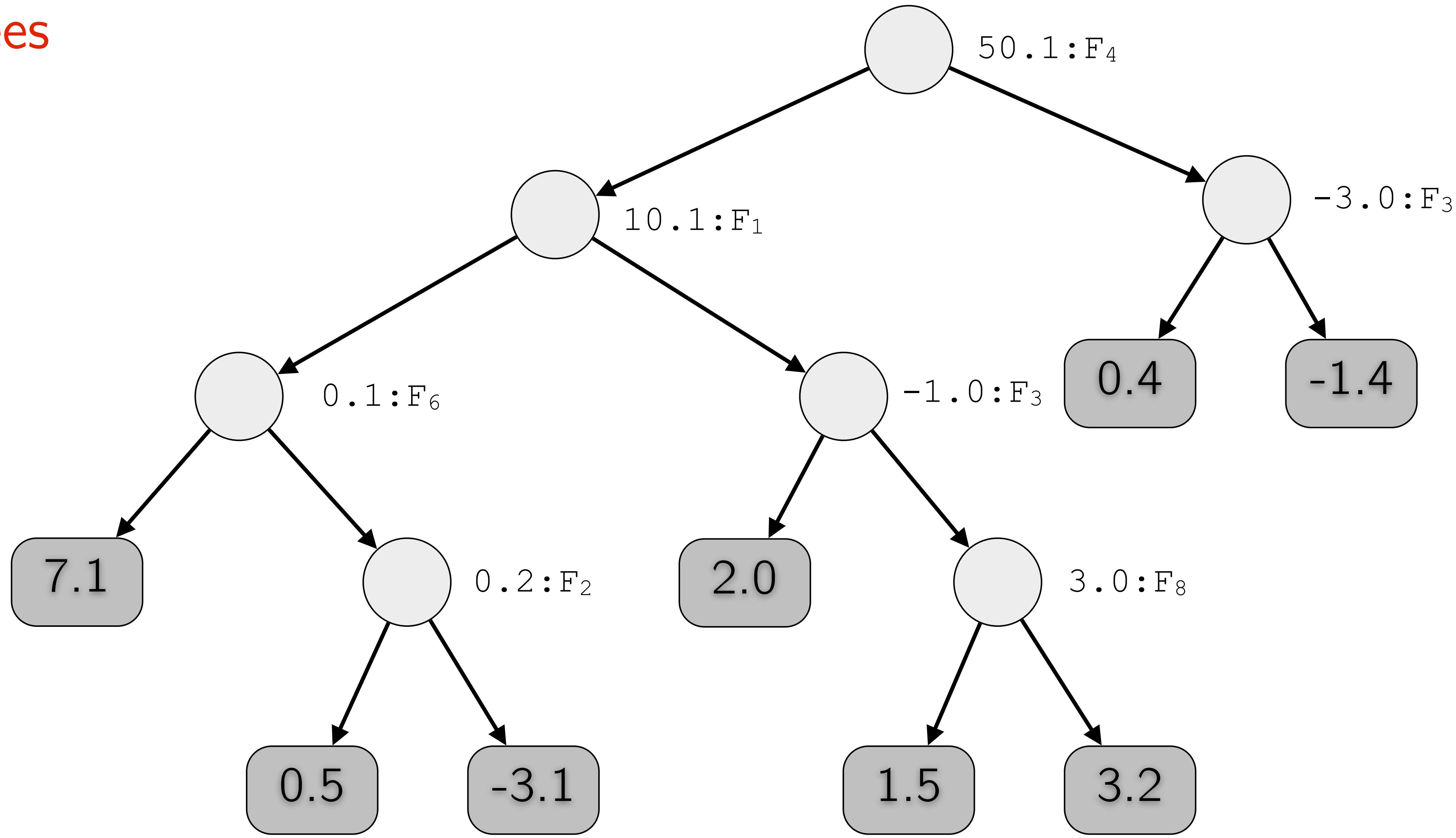
...

\# docs = >100K

\# trees = 1K–20K

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Trees



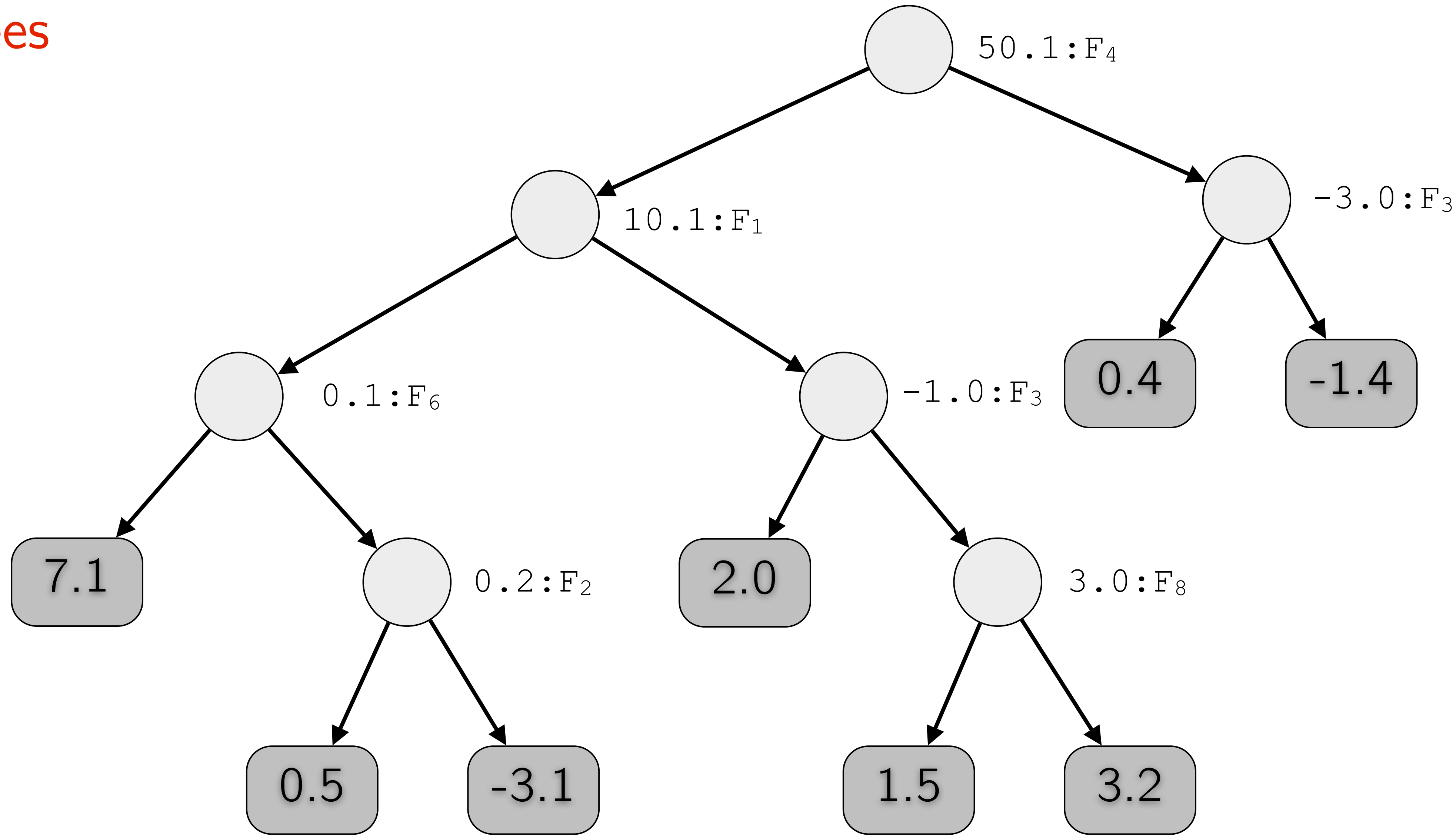
Documents

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

…

docs = >100K
trees = 1K–20K
features = 100–1000

Trees



Documents

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

...

docs = >100K

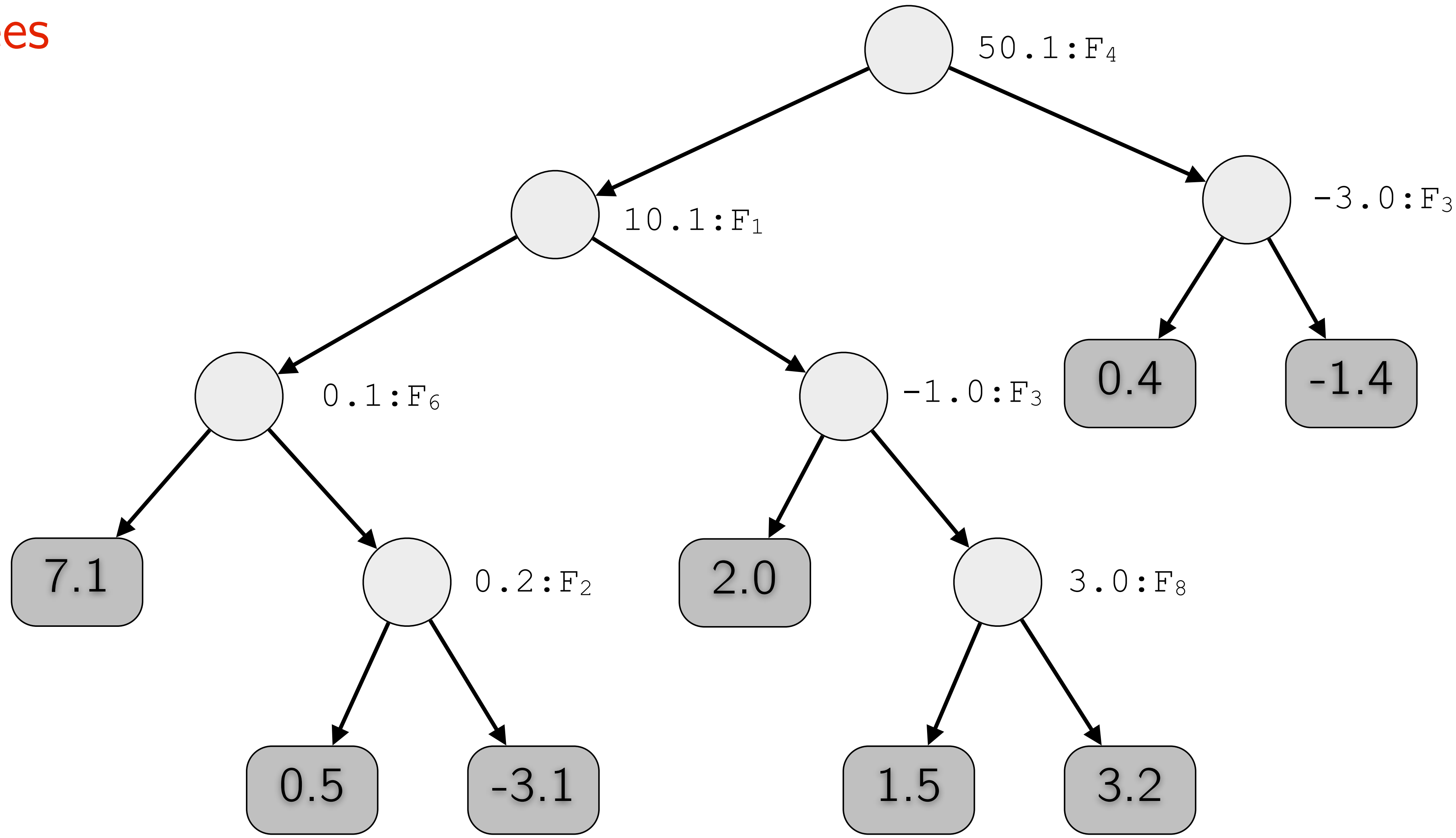
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

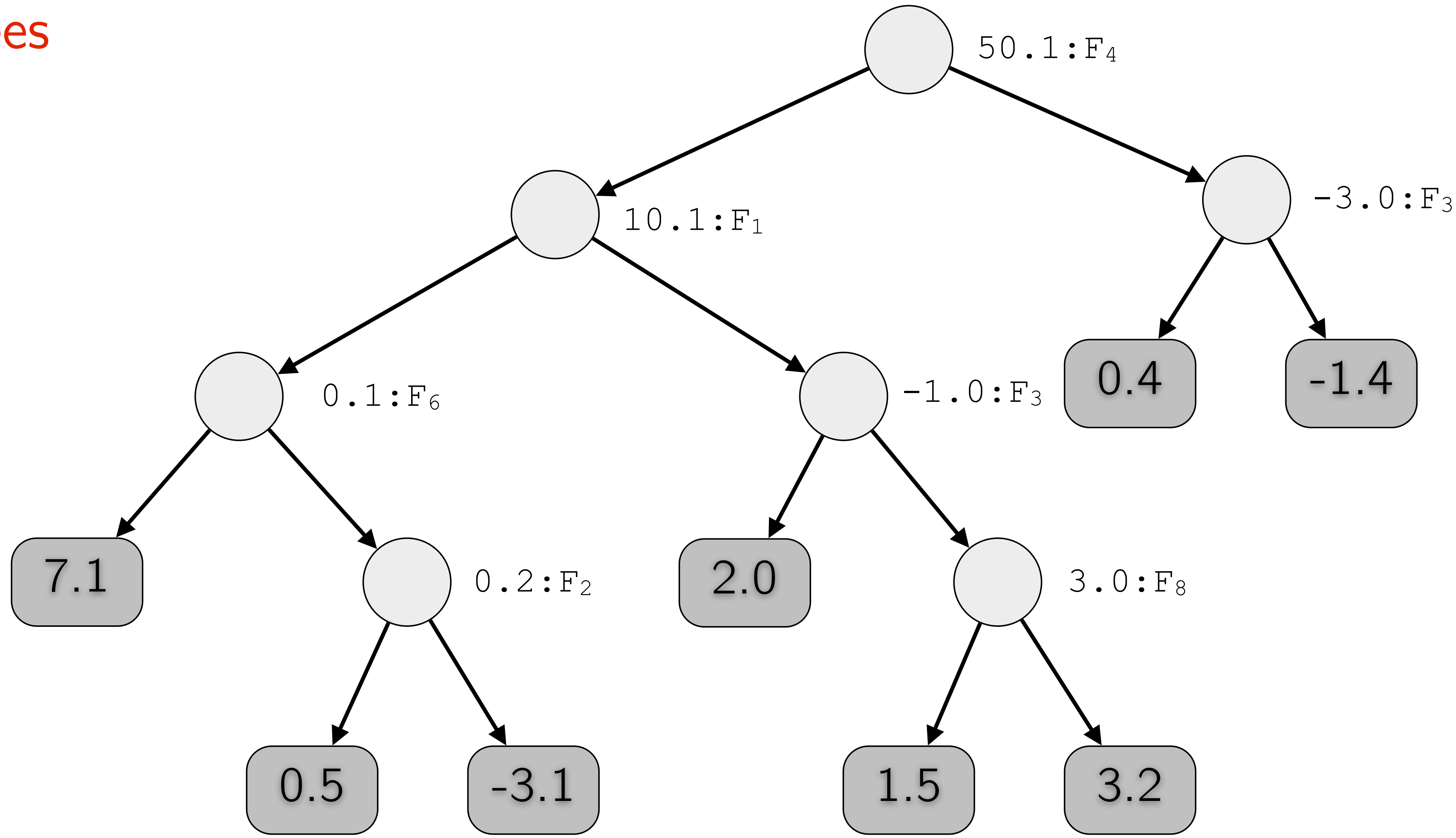
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

\# docs = >100K

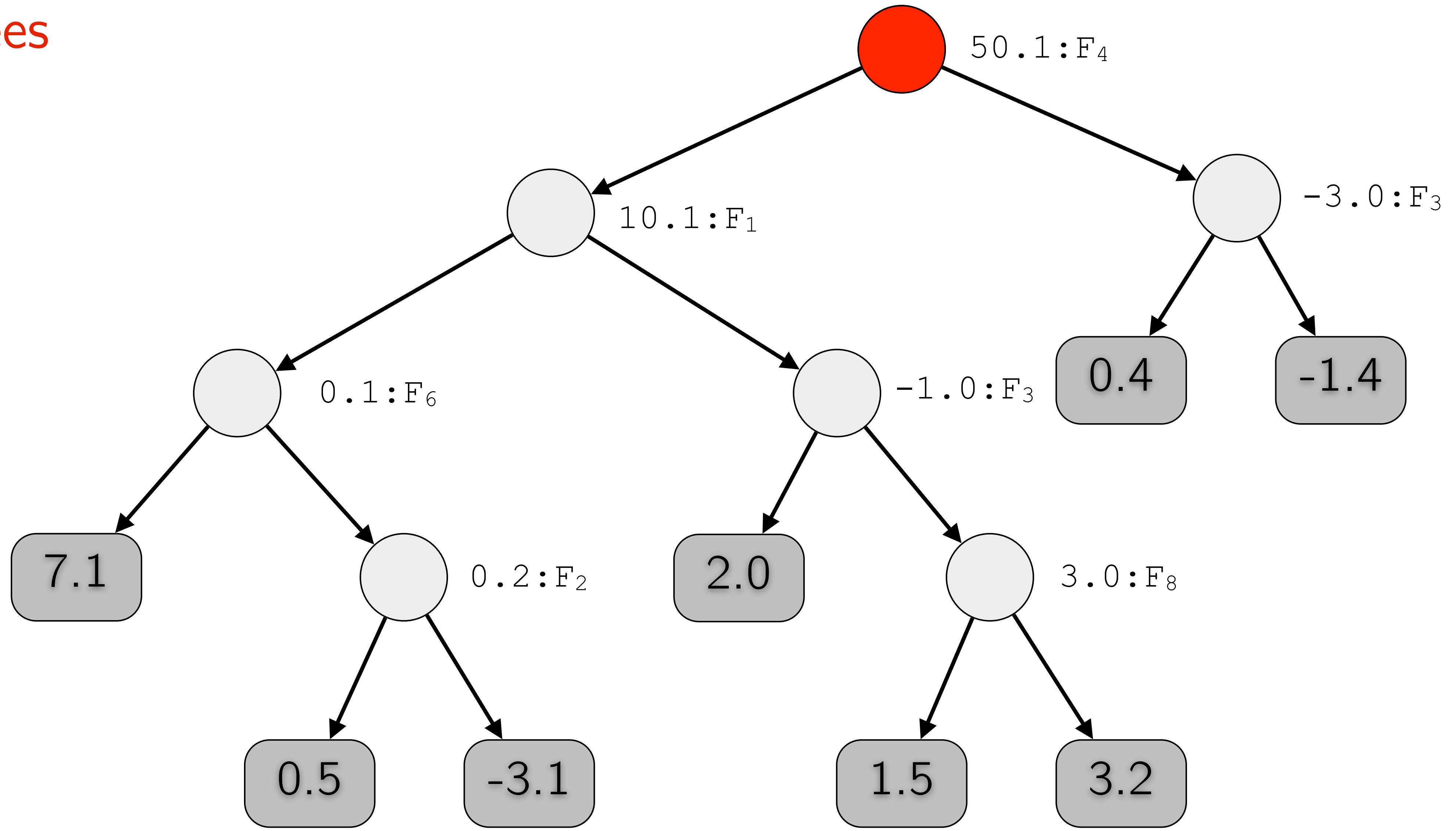
\# trees = 1K–20K

\# features = 100–1000

\# leaves = 4–64

Struct+

Trees



Documents

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

...

\# docs = >100K

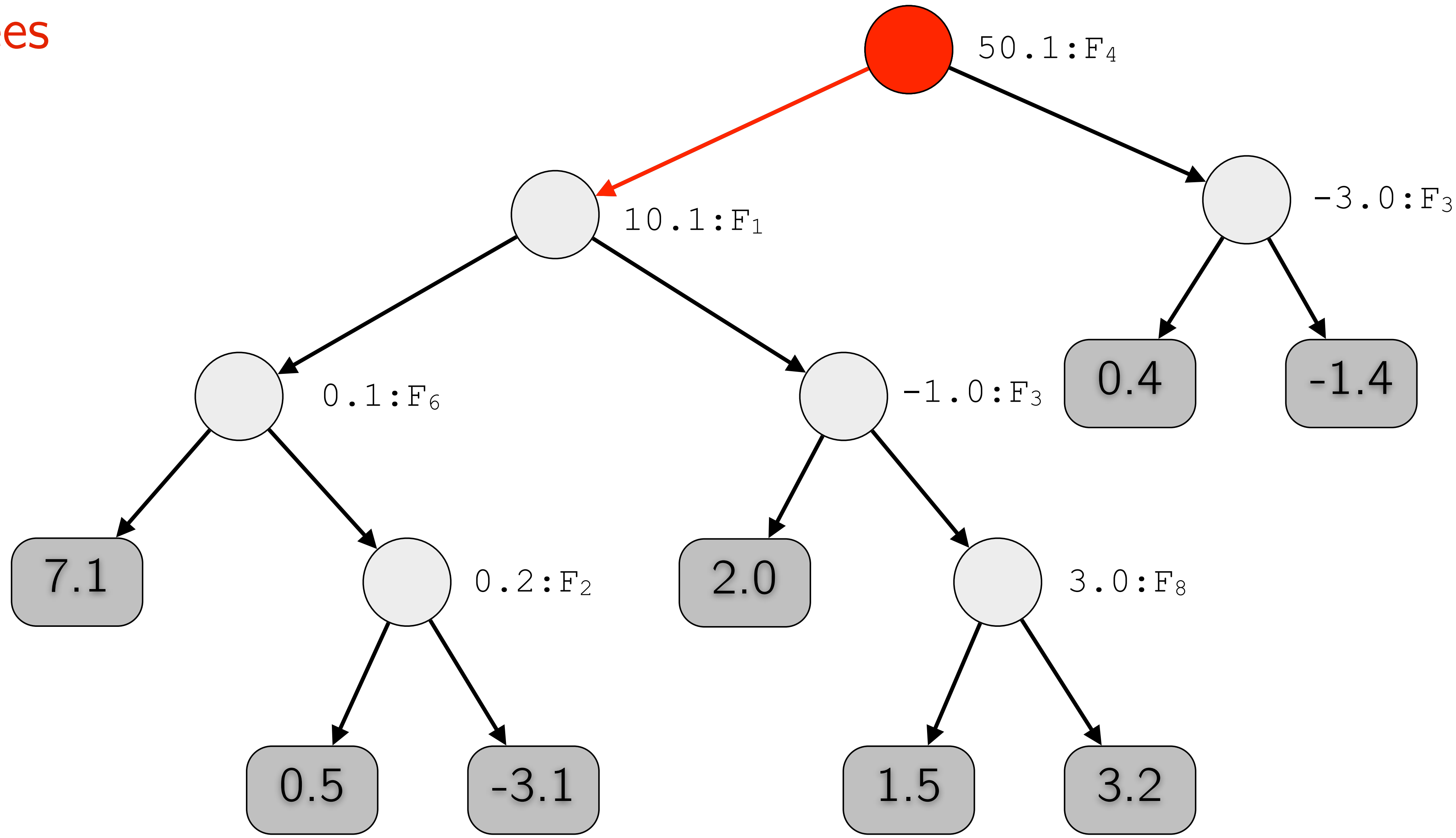
\# trees = 1K–20K

\# features = 100–1000

\# leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

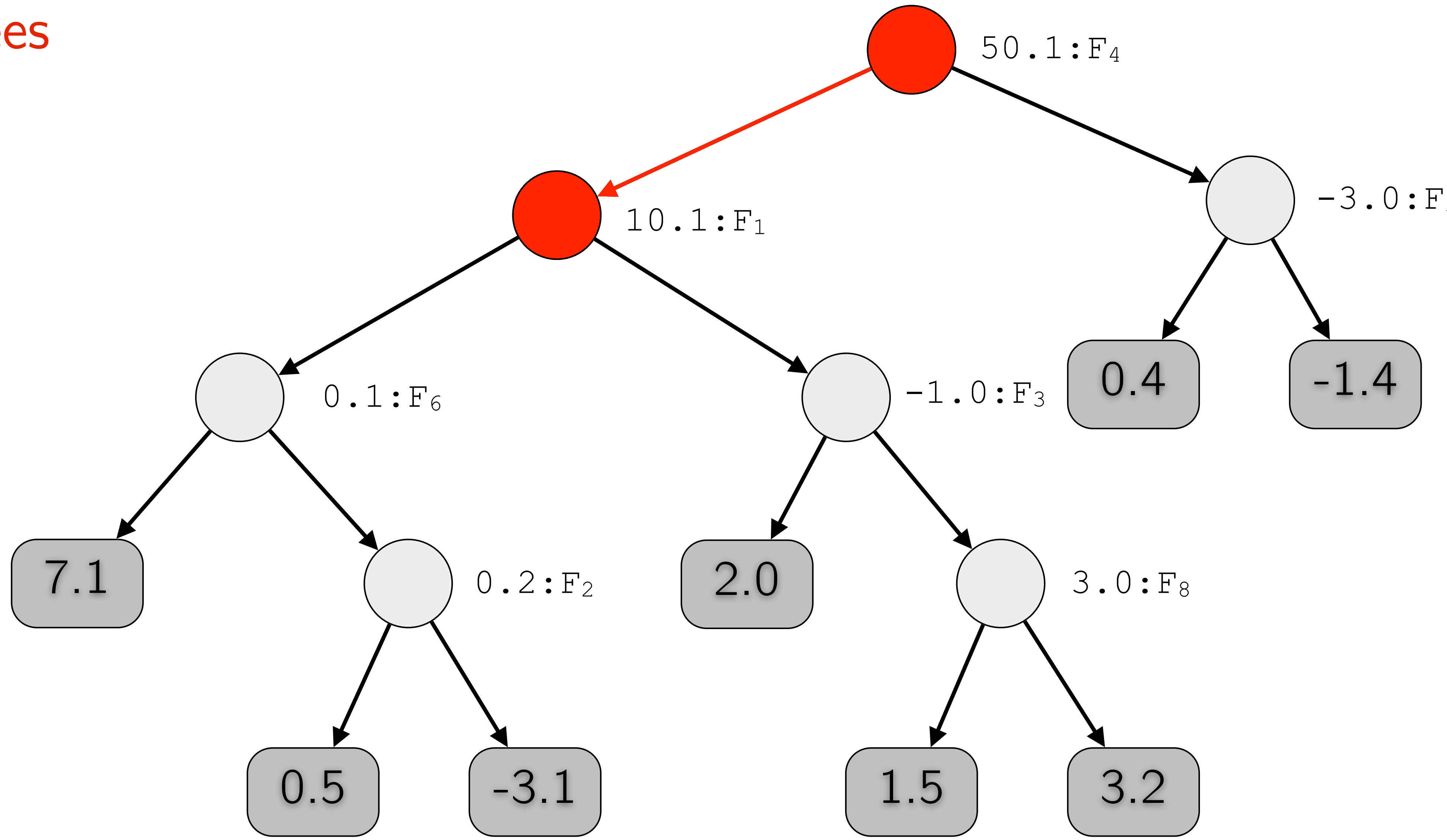
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

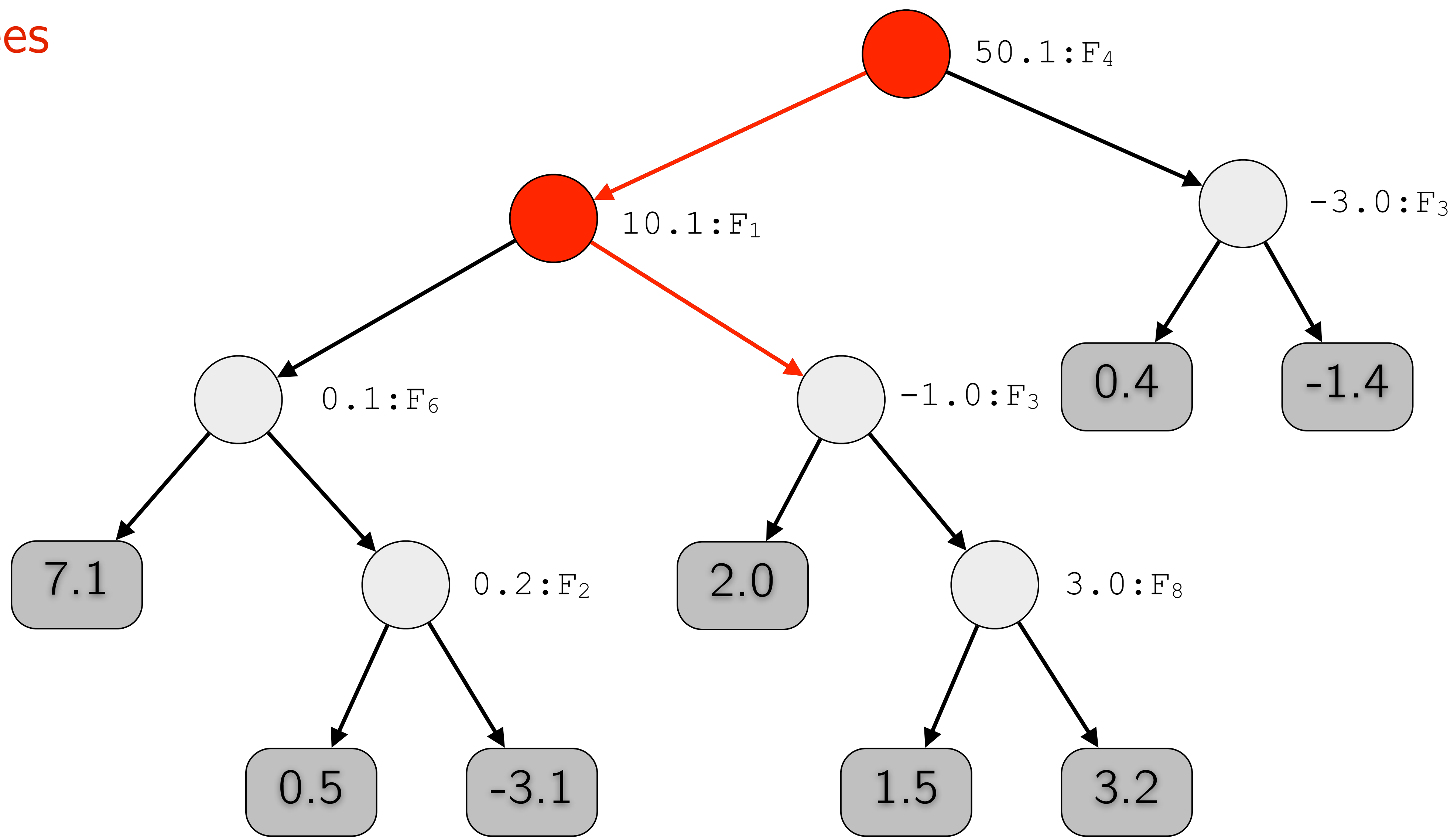
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

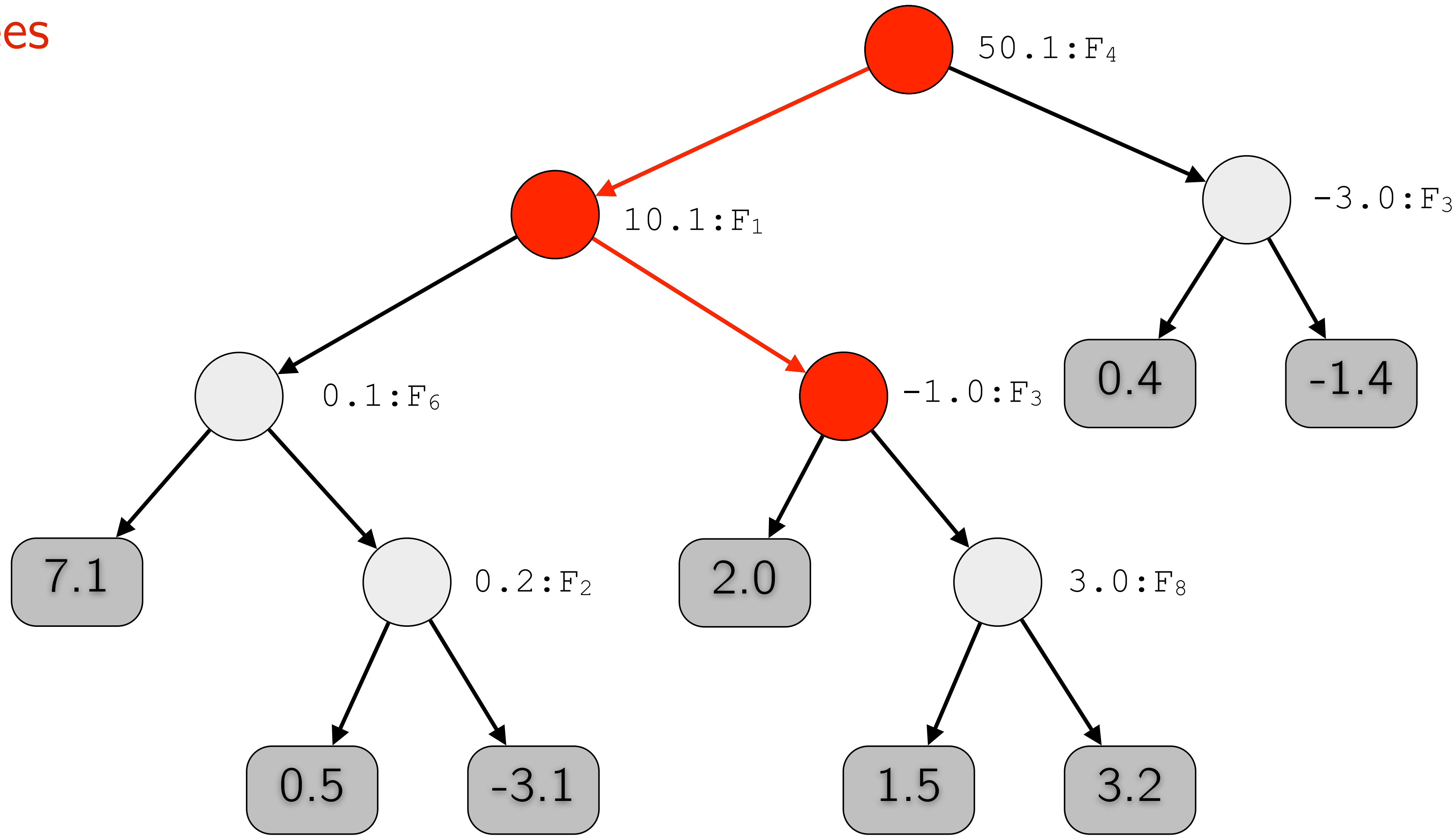
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

\# docs = >100K

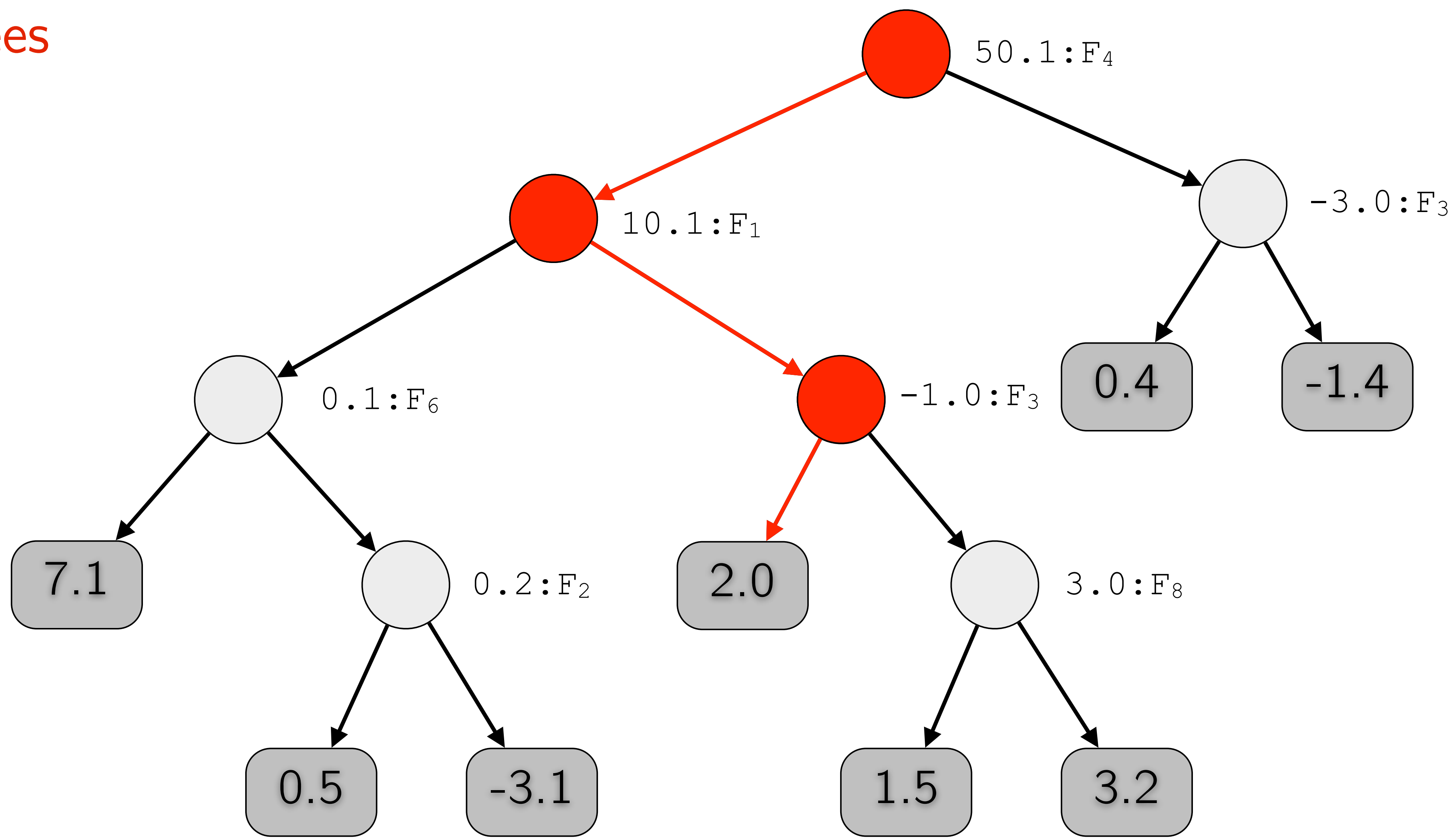
\# trees = 1K–20K

\# features = 100–1000

\# leaves = 4–64

Struct+

Trees



Documents

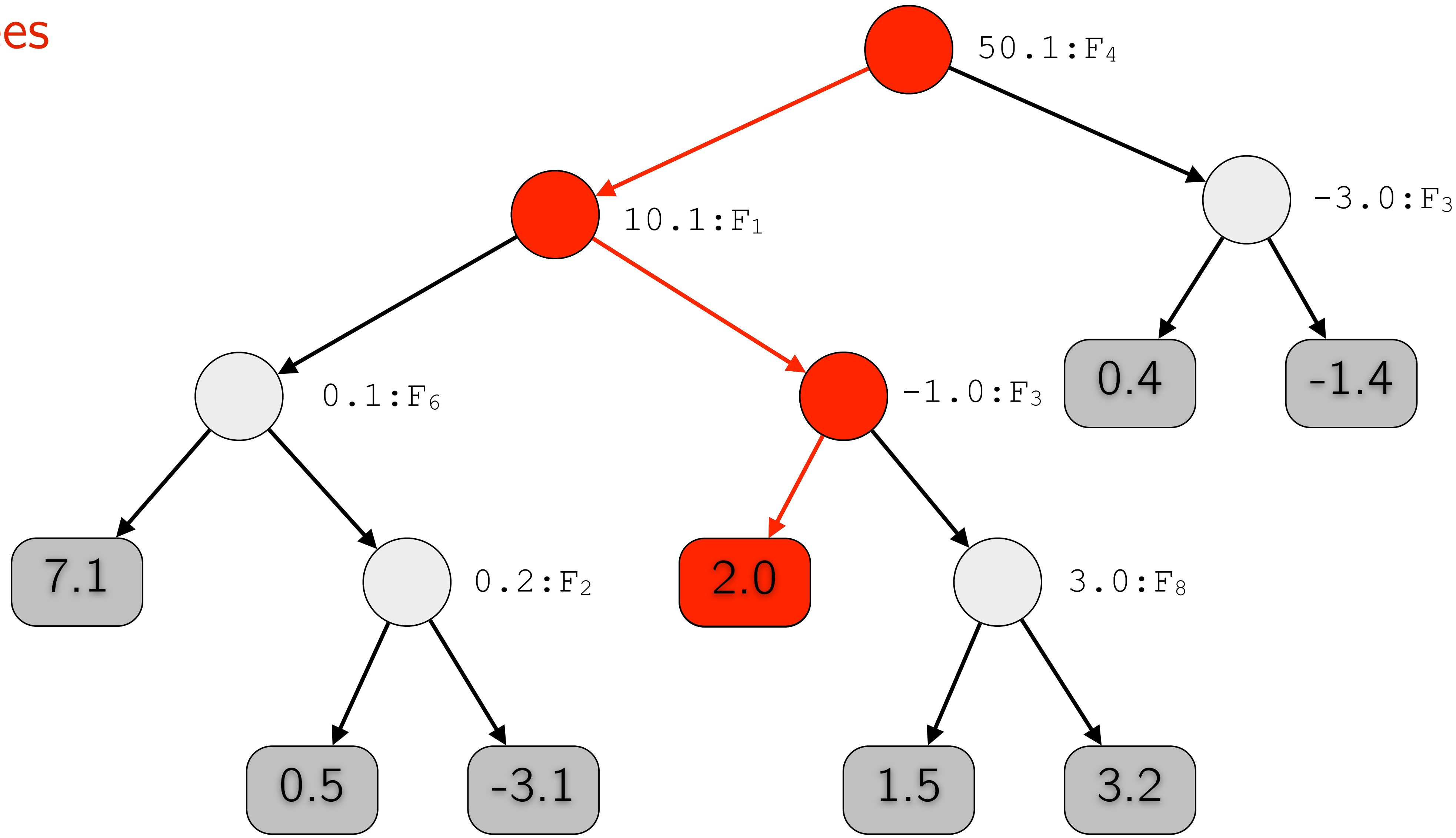
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

docs = >100K
trees = 1K–20K
features = 100–1000
leaves = 4–64

Struct+

Trees



Documents

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

…

\# docs = >100K

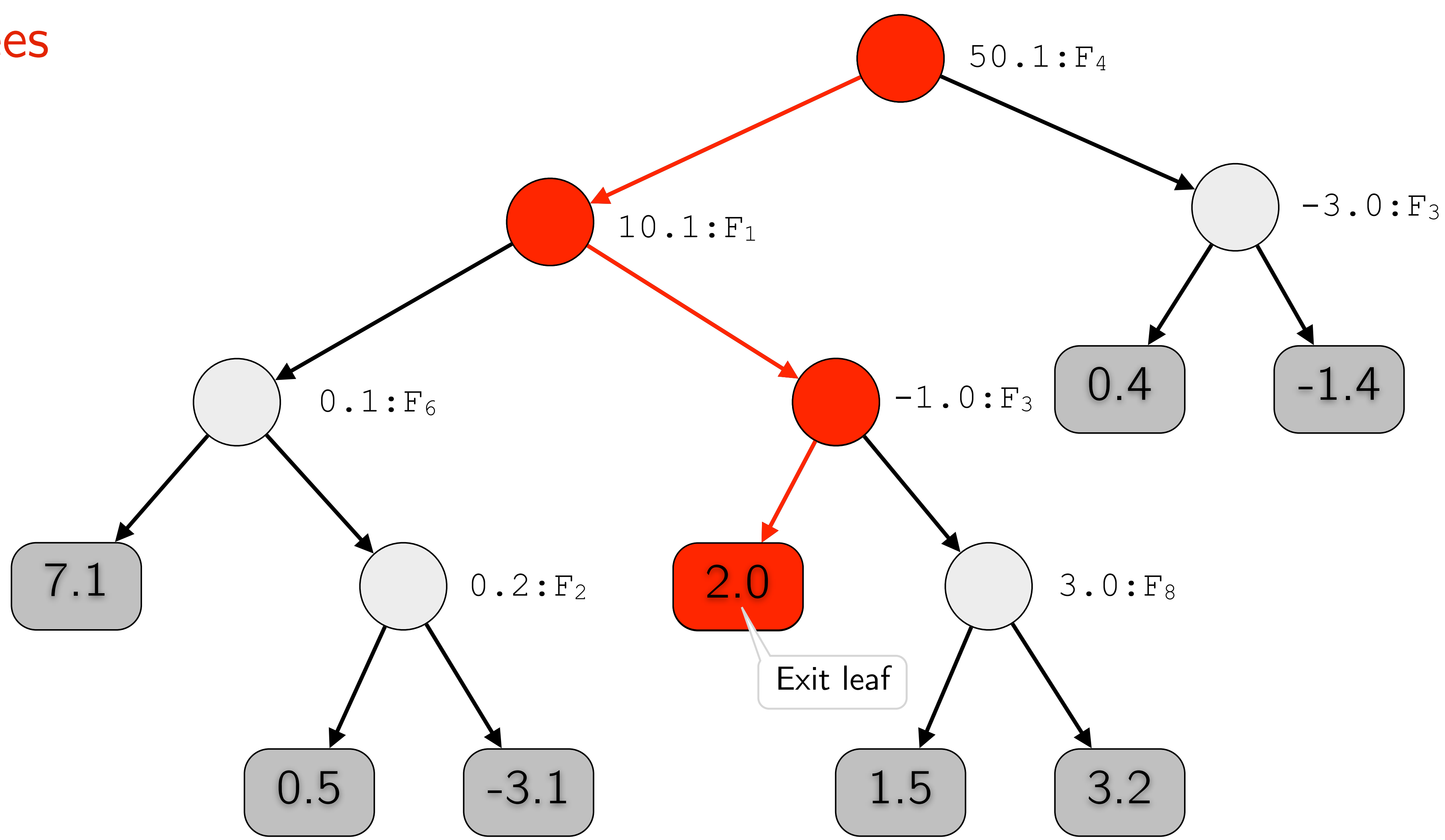
\# trees = 1K–20K

\# features = 100–1000

\# leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

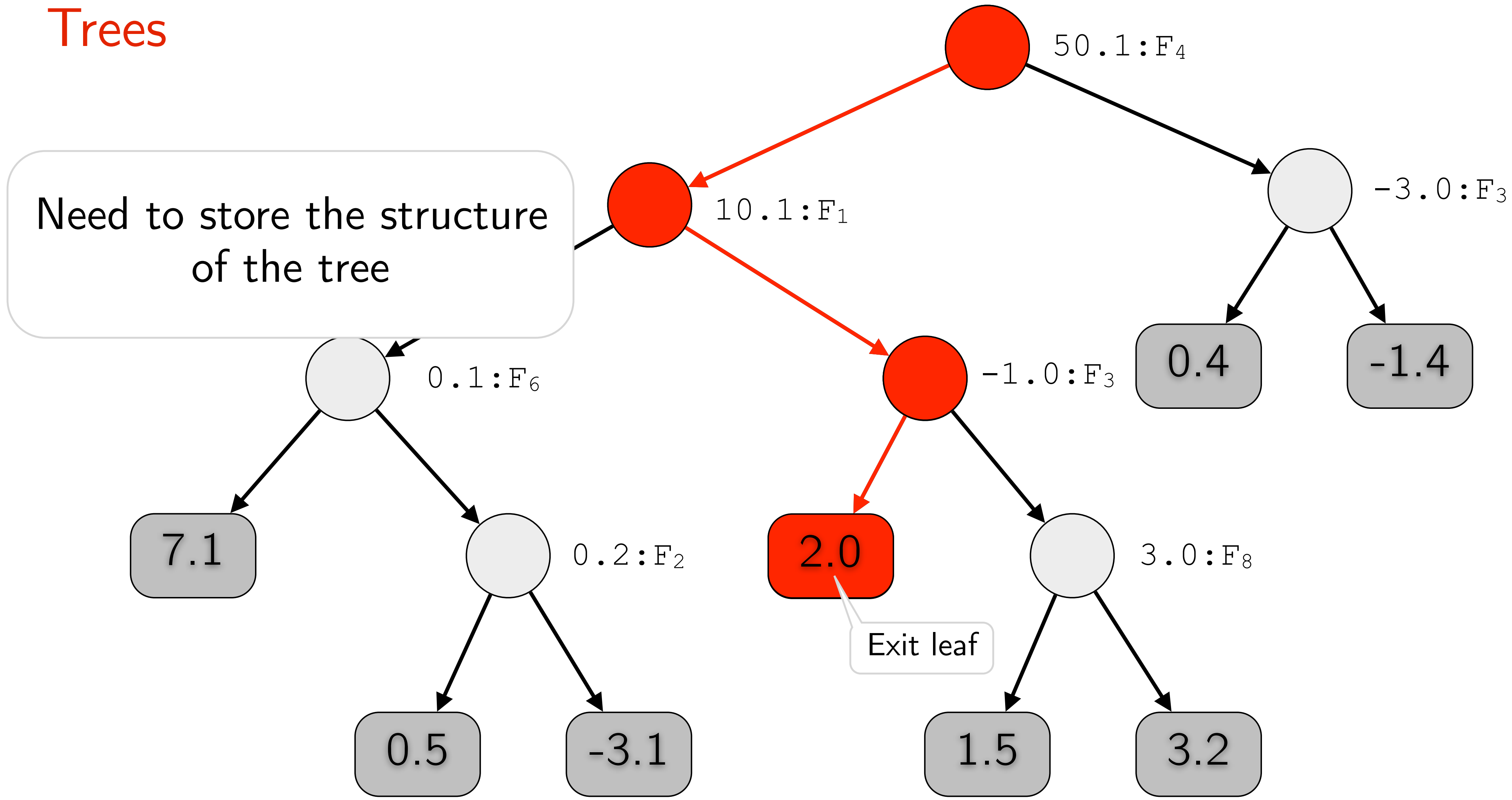
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

...

docs = >100K

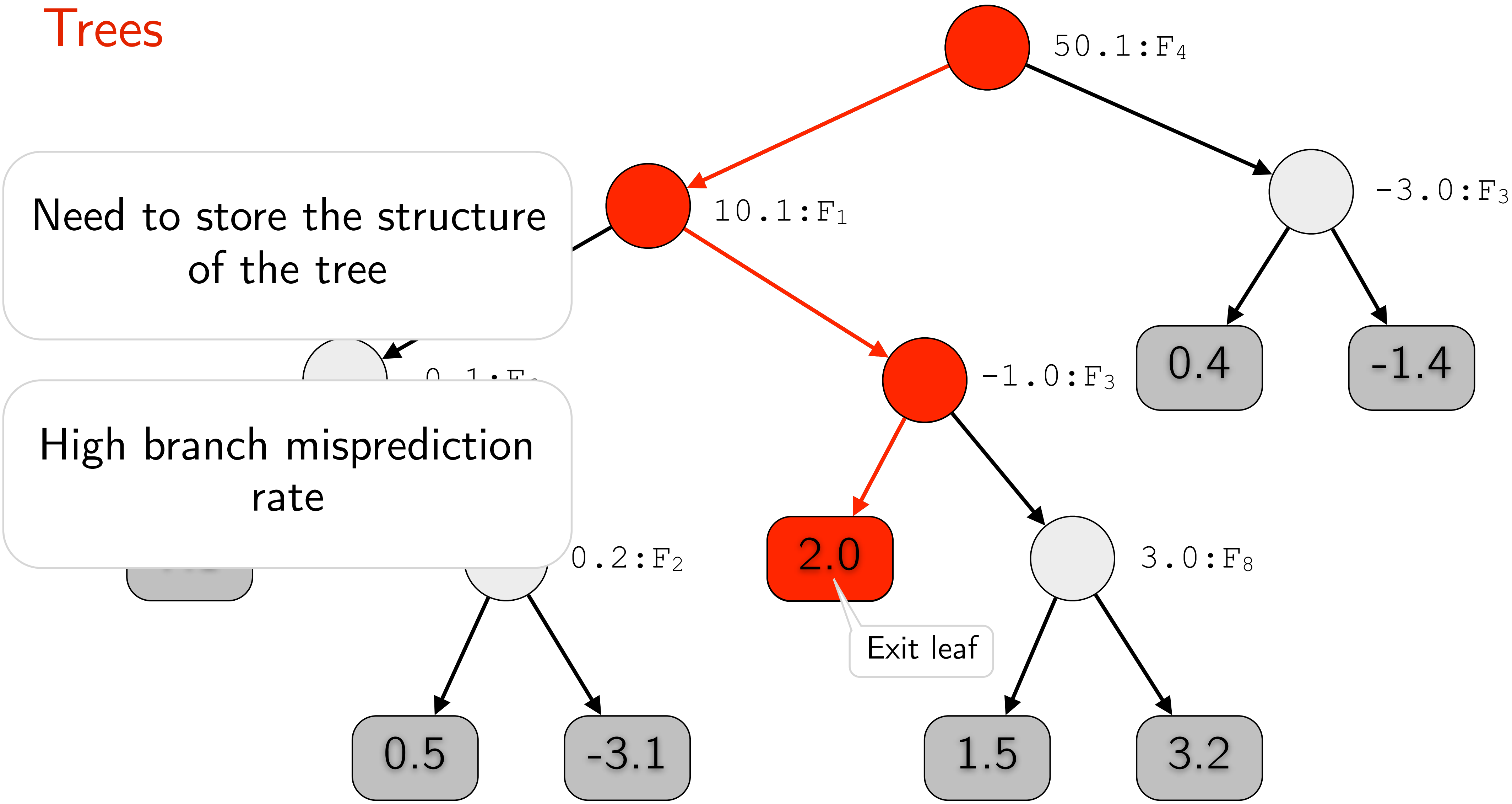
trees = 1K–20K

features = 100–1000

leaves = 4–64

Struct+

Trees



Documents

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

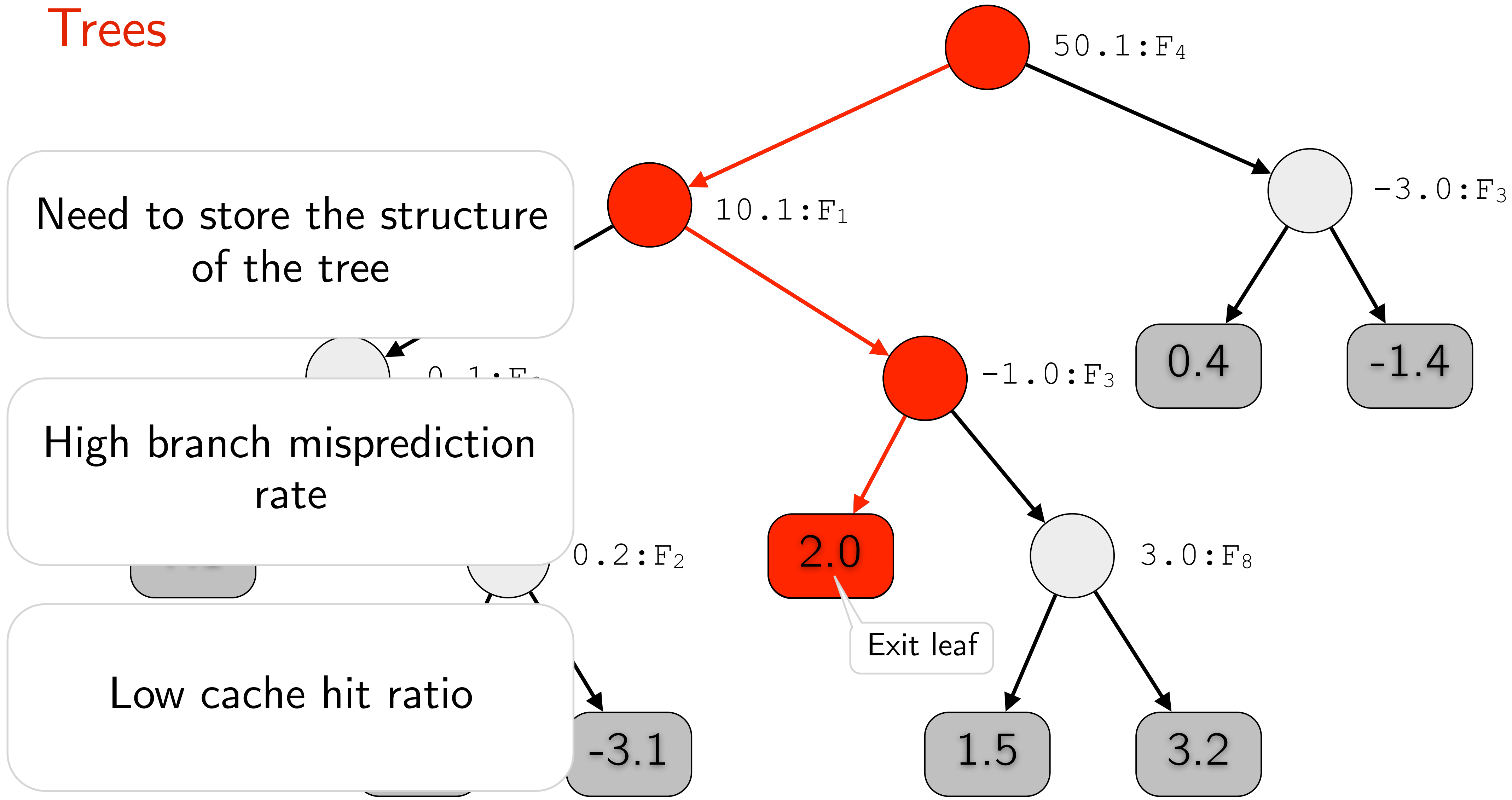
...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Struct+

Trees



Documents

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

...

docs = >100K

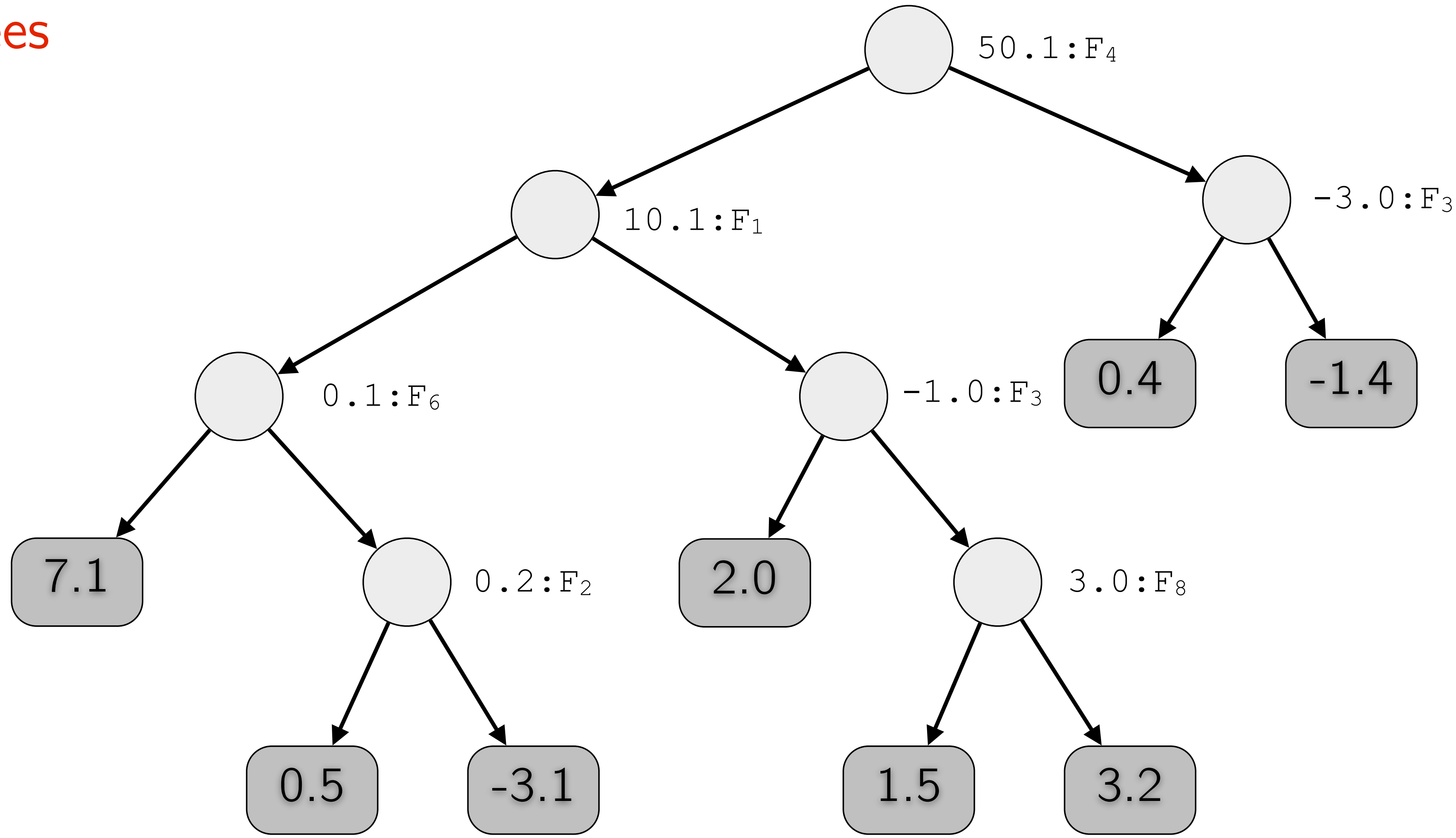
trees = 1K–20K

features = 100–1000

leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Trees



Documents

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

...

docs = >100K

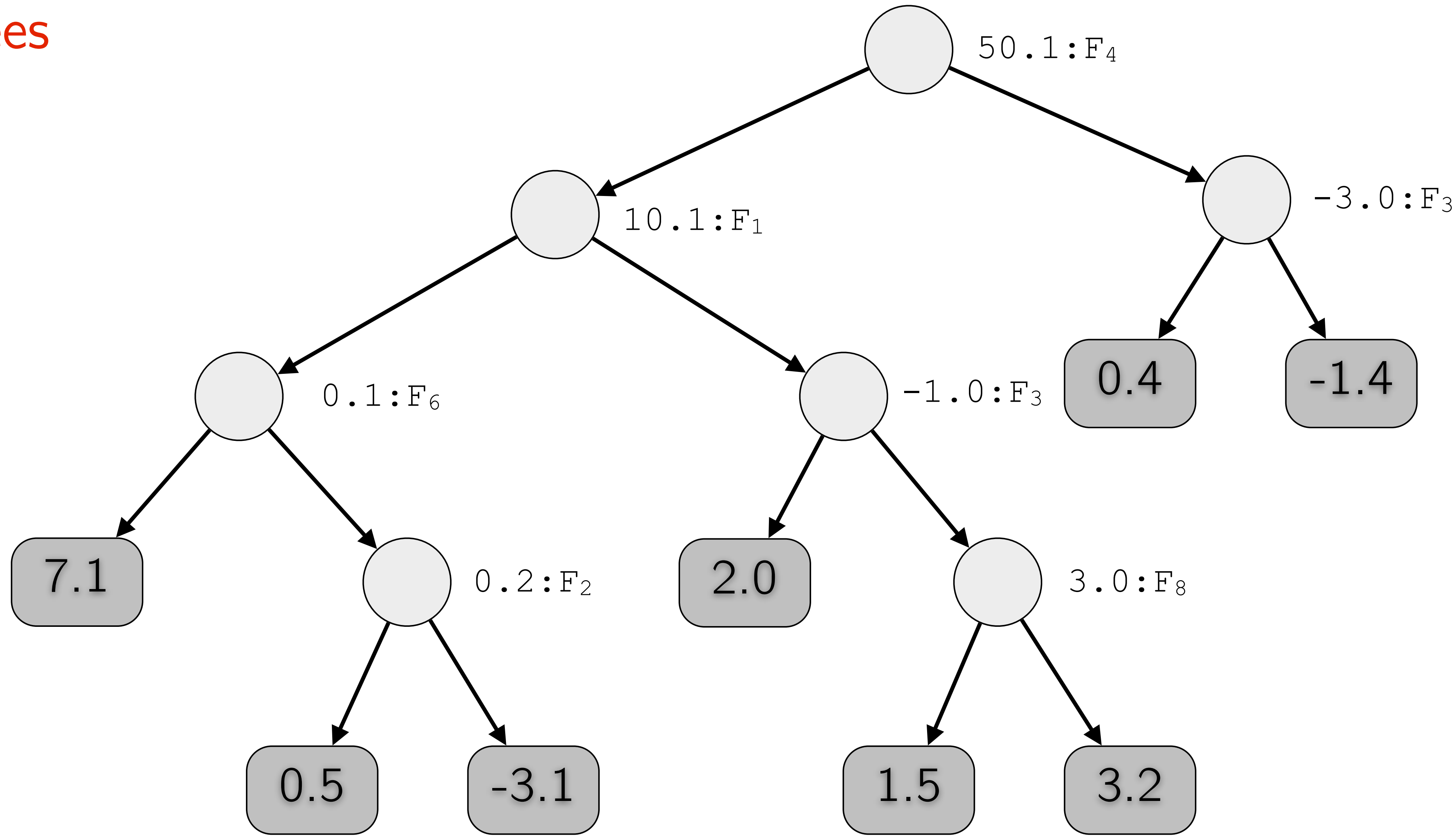
trees = 1K–20K

features = 100–1000

leaves = 4–64

If-then-else

Trees



Documents

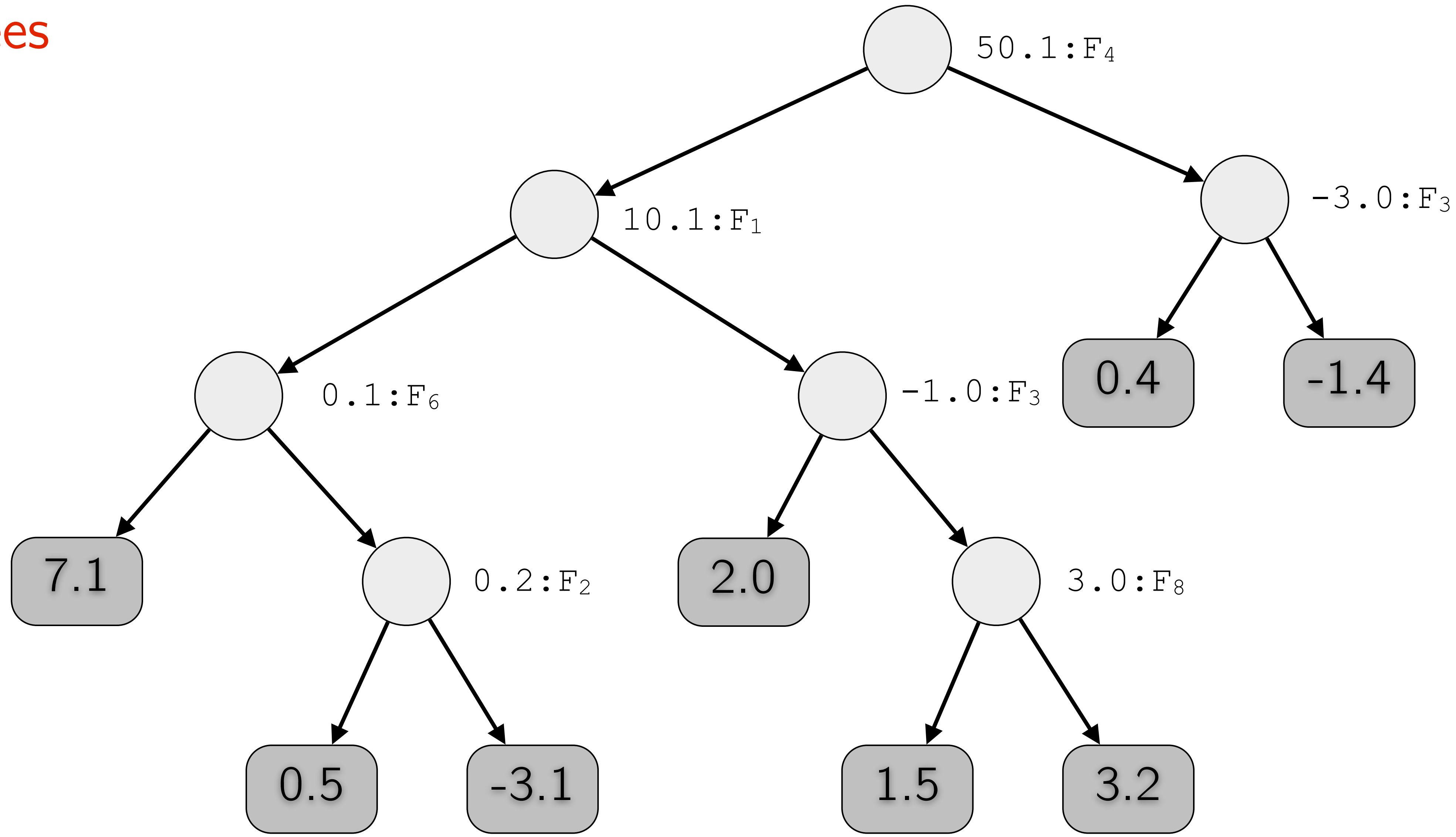
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

docs = >100K
trees = 1K–20K
features = 100–1000
leaves = 4–64

If-then-else

Trees



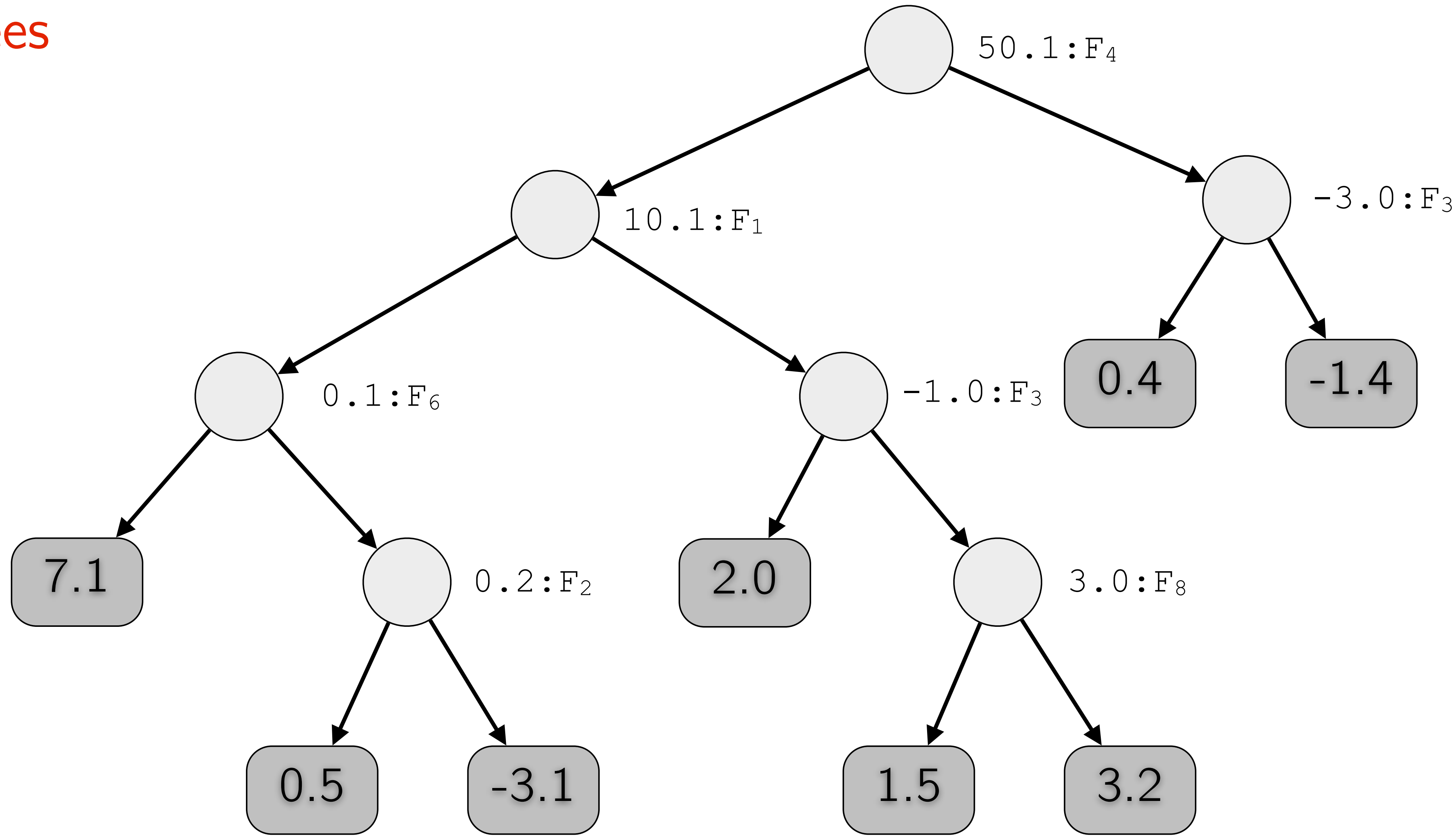
Documents

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-then-else

Trees



Documents

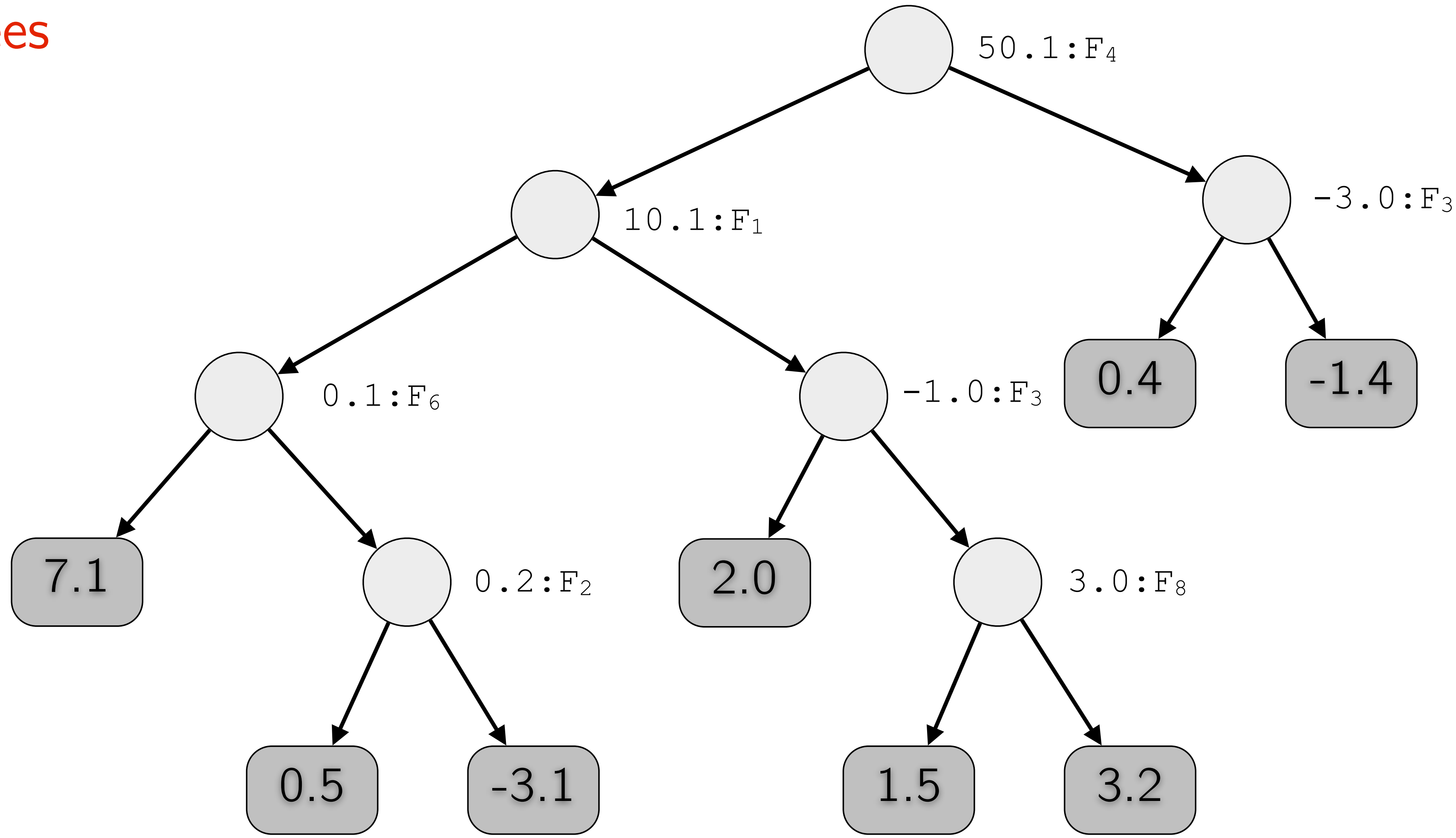
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
if (x[4] <= 50.1) {
    // recurses on the left subtree
    ...
```

If-then-else

Trees



Documents

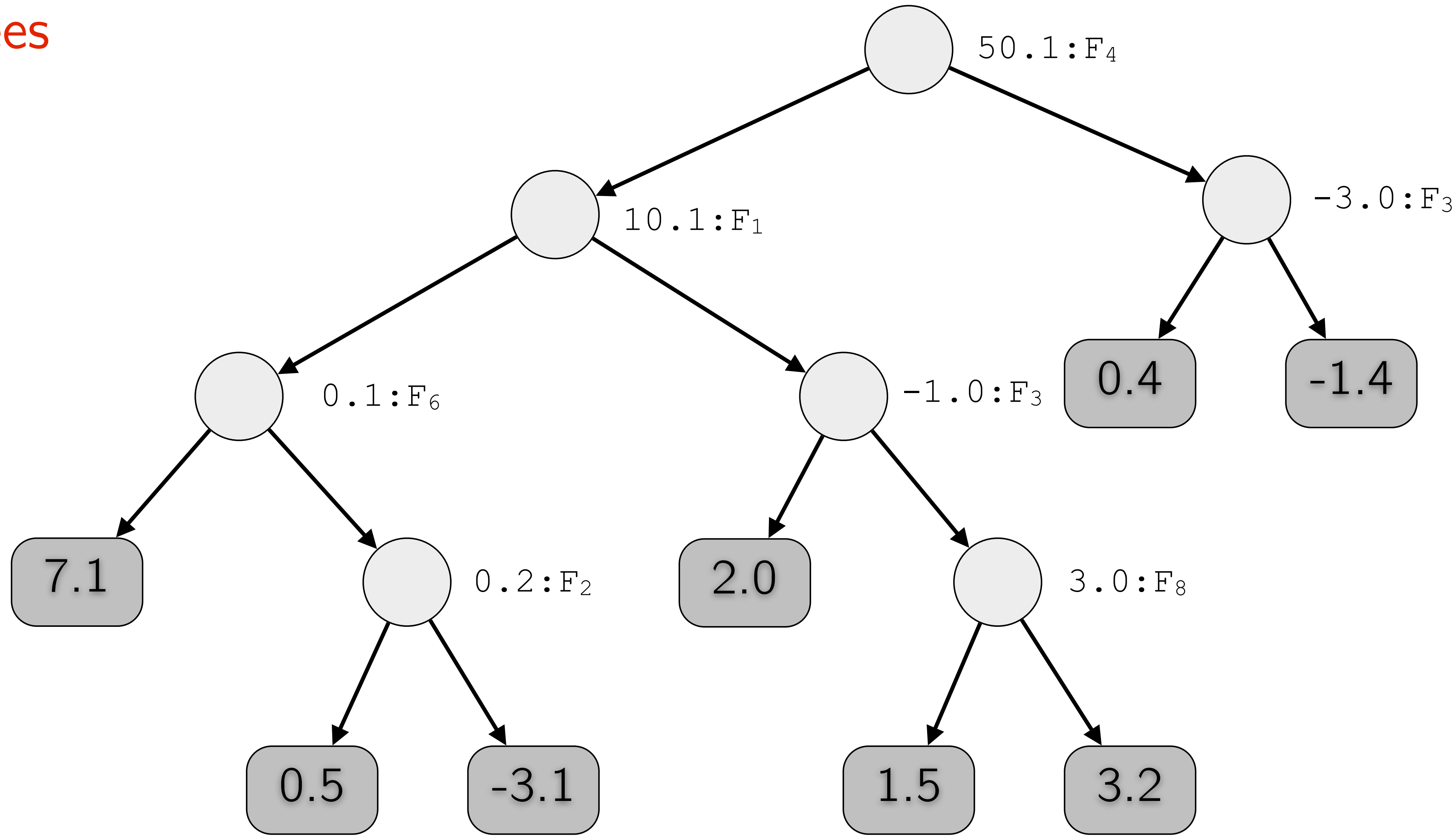
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
if (x[4] <= 50.1) {
    // recurses on the left subtree
    ...
} else {
    // recurses on the right subtree
```

If-then-else

Trees



Documents

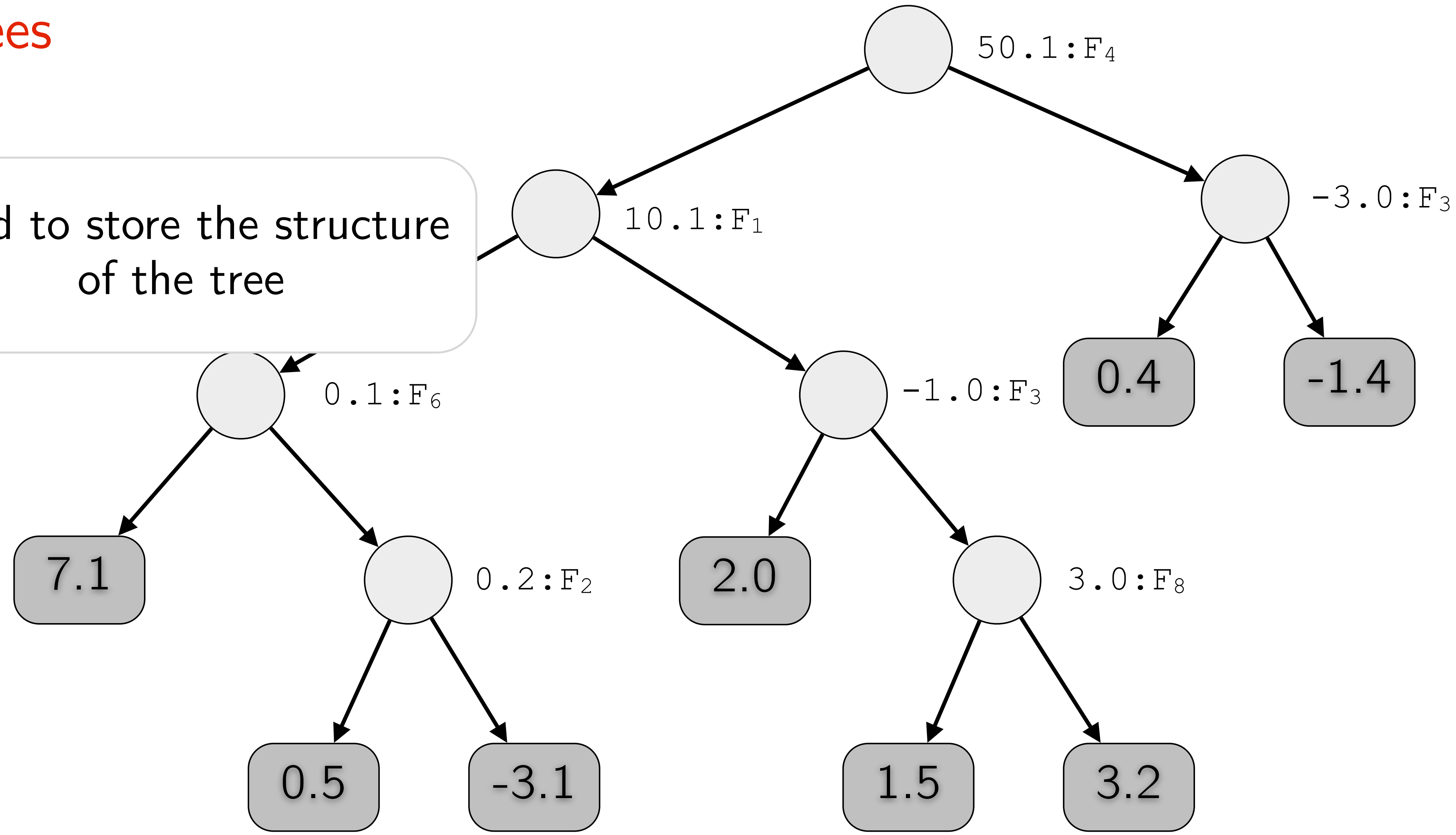
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
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;
}
```

If-then-else

Trees



Need to store the structure of the tree

Documents

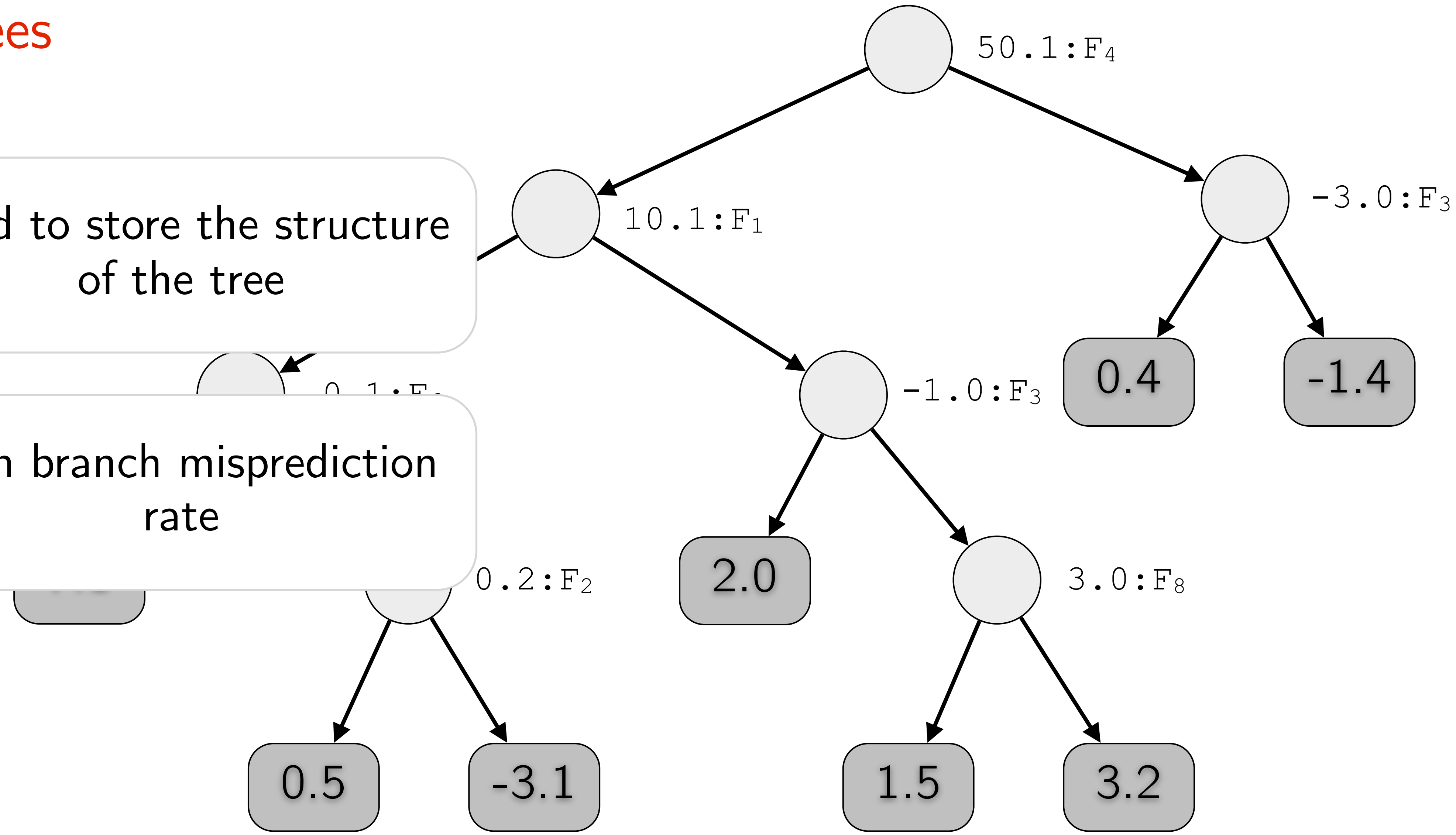
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

```
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;
}
```

If-then-else

Trees



Need to store the structure of the tree

High branch misprediction rate

Documents

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

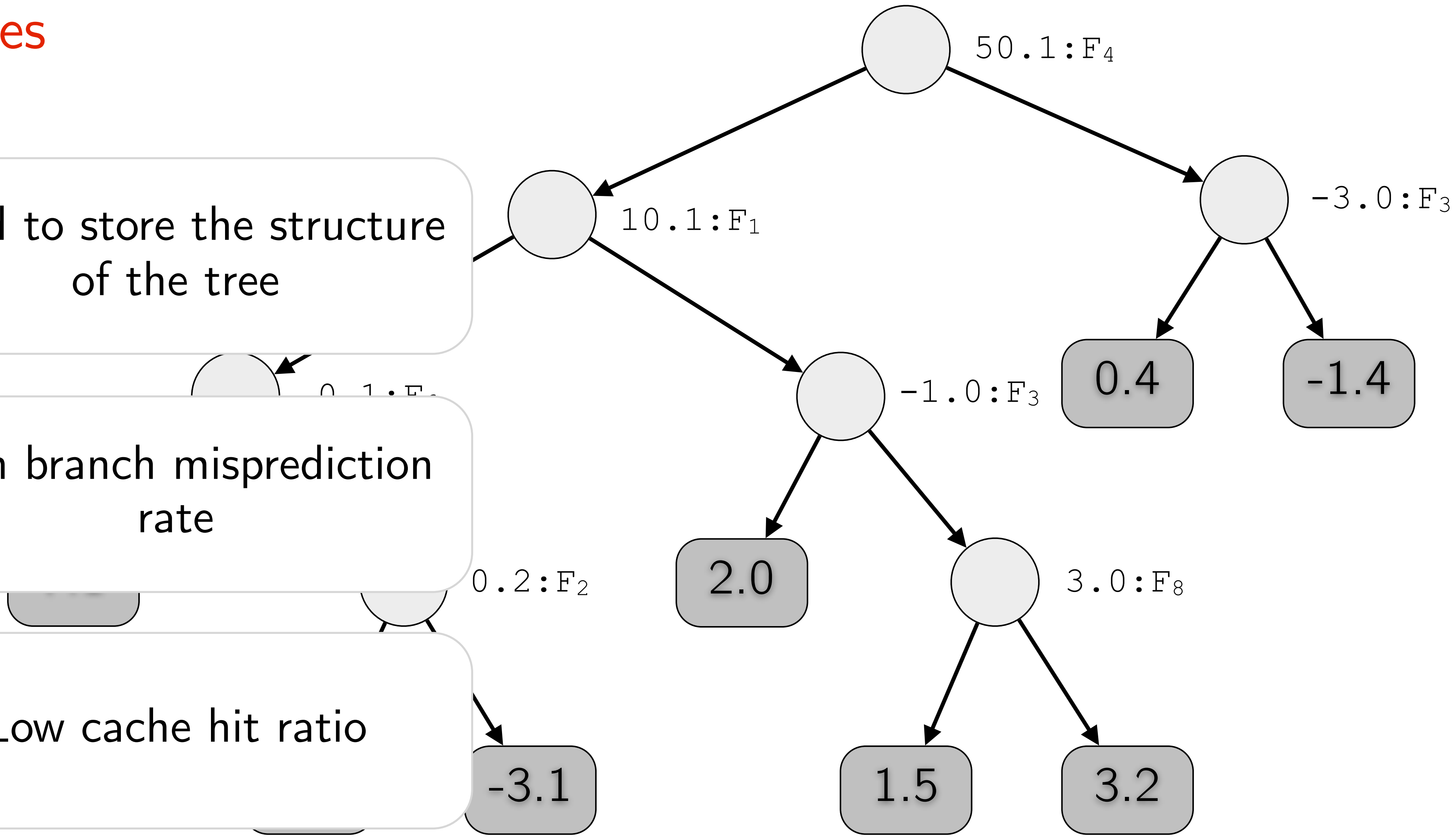
...

```
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;
}
```

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

If-then-else

Trees



Need to store the structure of the tree

High branch misprediction rate

Low cache hit ratio

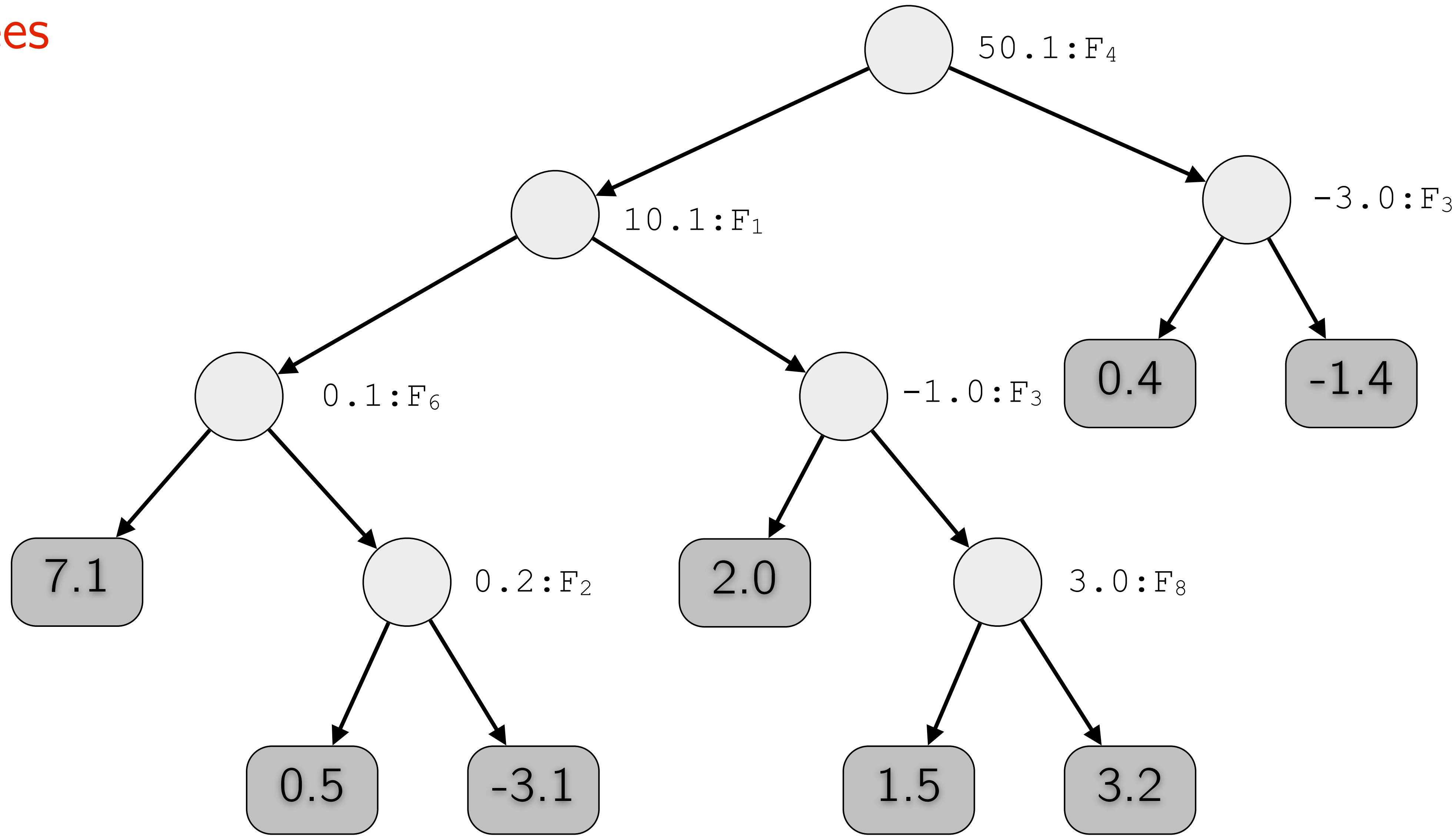
Documents

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;
}
```

Trees



Documents

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

…

docs = >100K

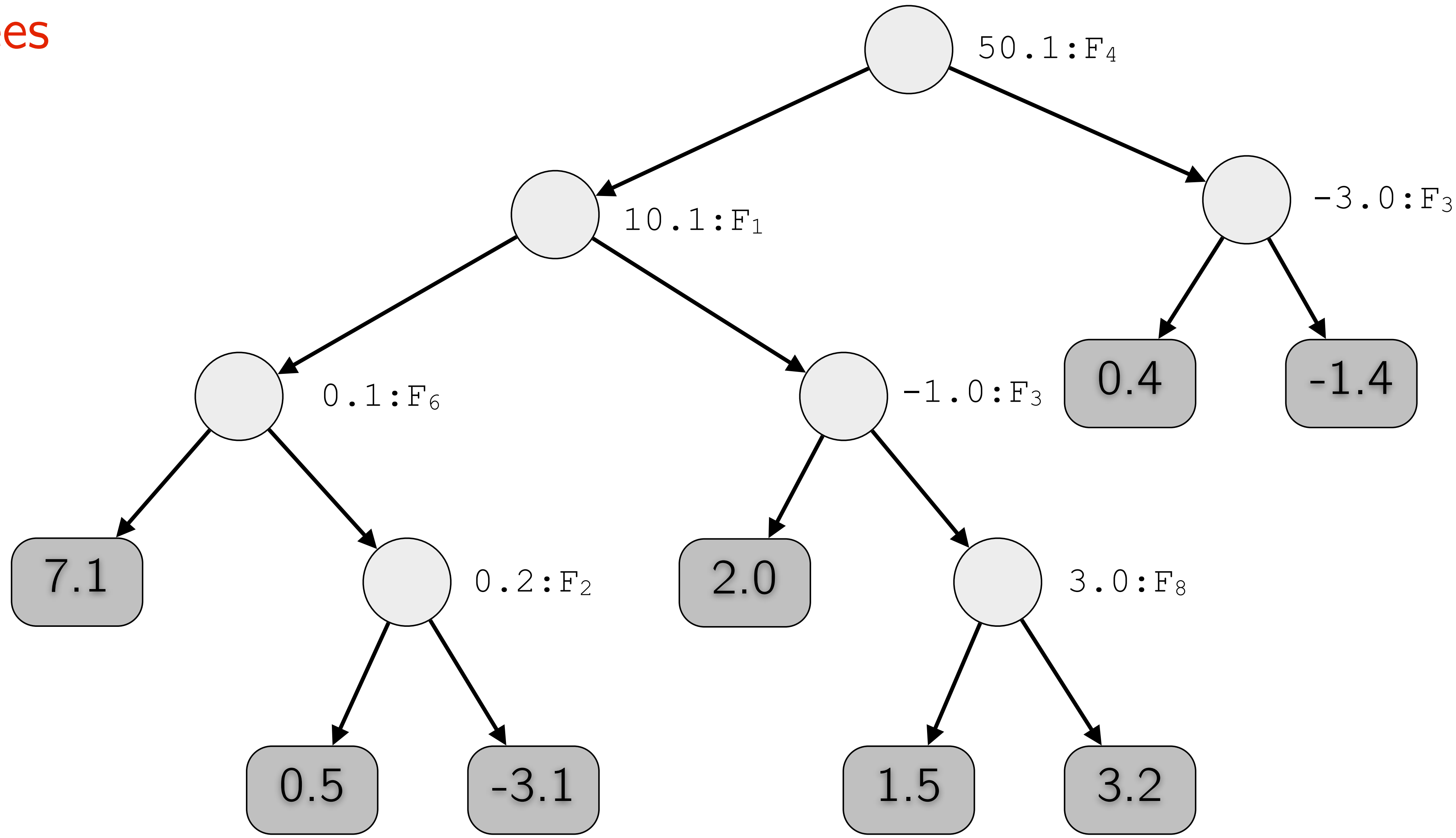
trees = 1K–20K

features = 100–1000

leaves = 4–64

Vpred

Trees



Documents

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

…

docs = >100K

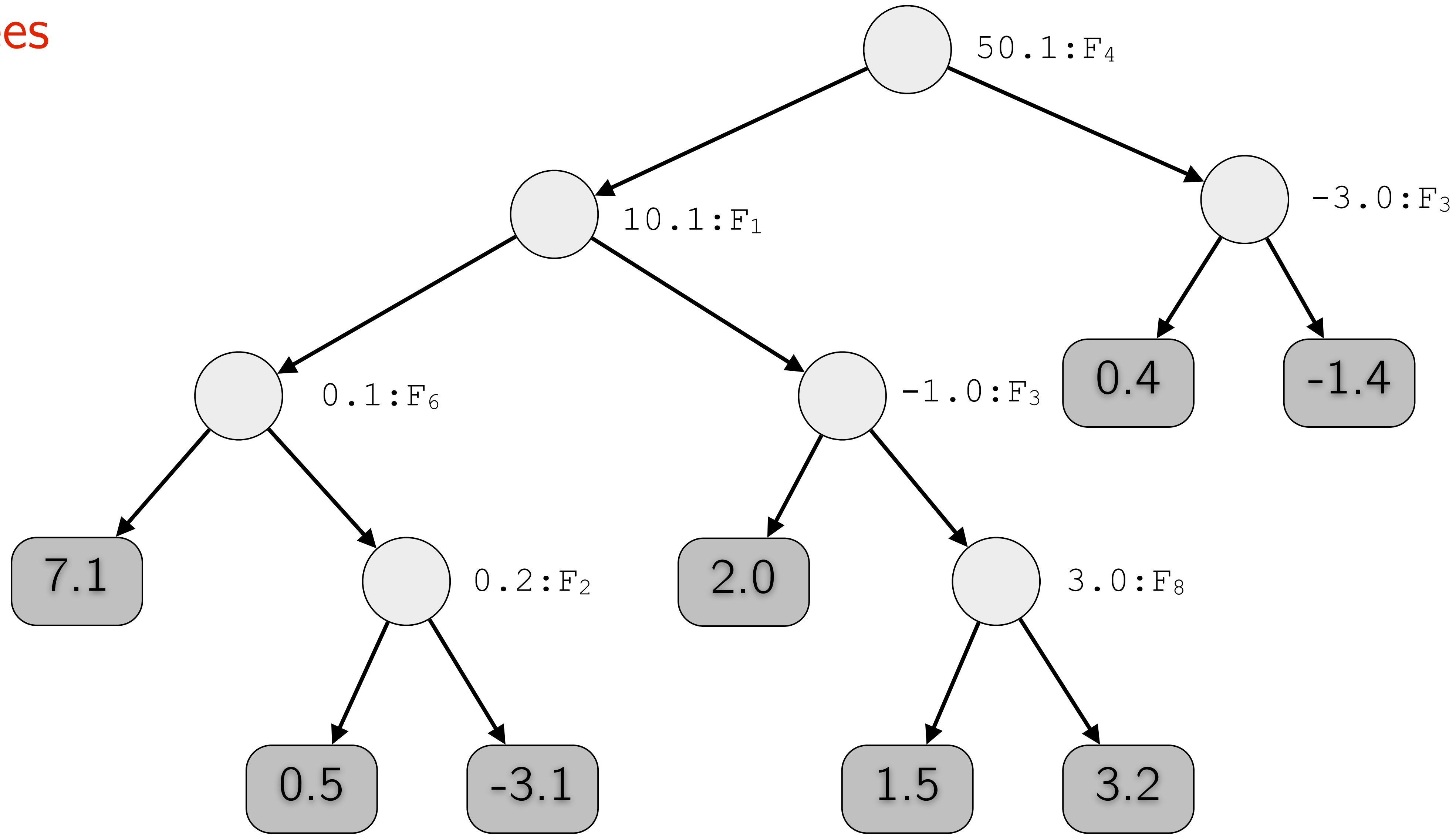
trees = 1K–20K

features = 100–1000

leaves = 4–64

Vpred

Trees



Documents

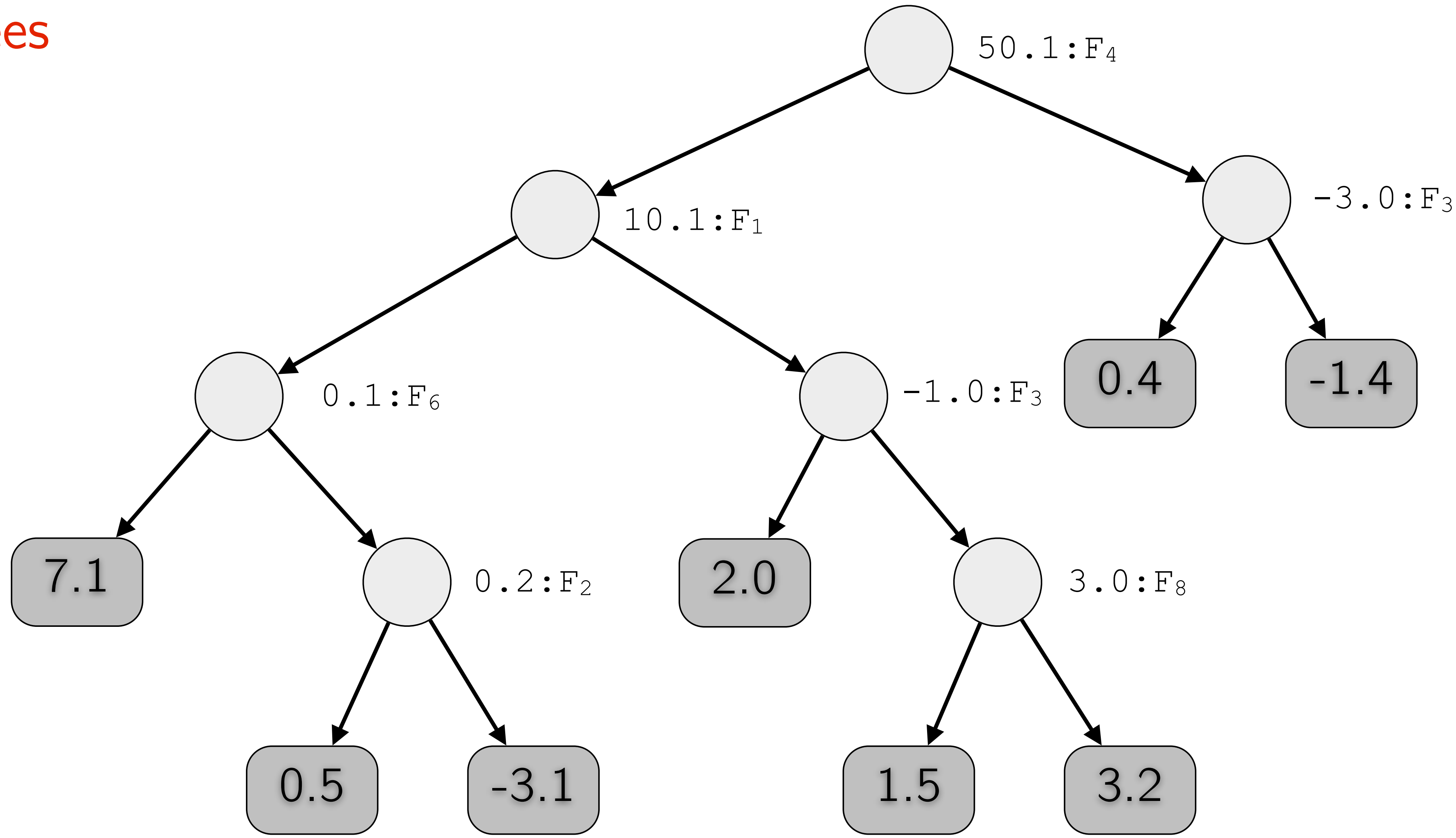
F_1	F_2	F_3	F_4	F_5	F
13.3	0.12	-1.2	43.9	11	-0
10.9	0.08	-1.1	42.9	15	-0
11.2	0.6	-0.2	54.1	13	-0.5

...

```
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];
}
```

Vpred

Trees



Documents

16 docs

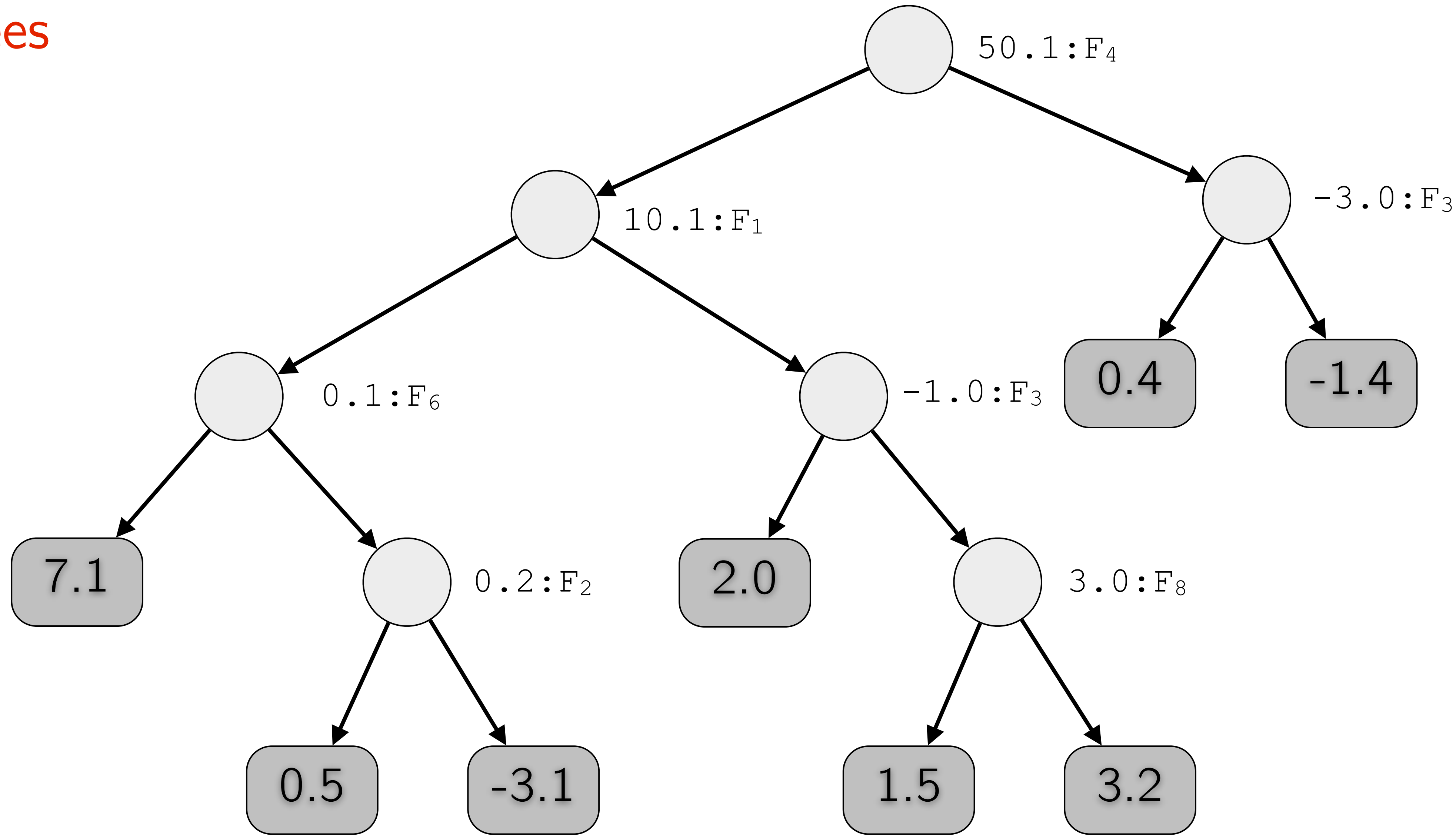
F_1	F_2	F_3	F_4	F_5	F
13.3	0.12	-1.2	43.9	11	-0
10.9	0.08	-1.1	42.9	15	-0
11.2	0.6	-0.2	54.1	13	-0.5

...

```
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];
}
```

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Trees



Documents

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

...

docs = >100K

trees = 1K–20K

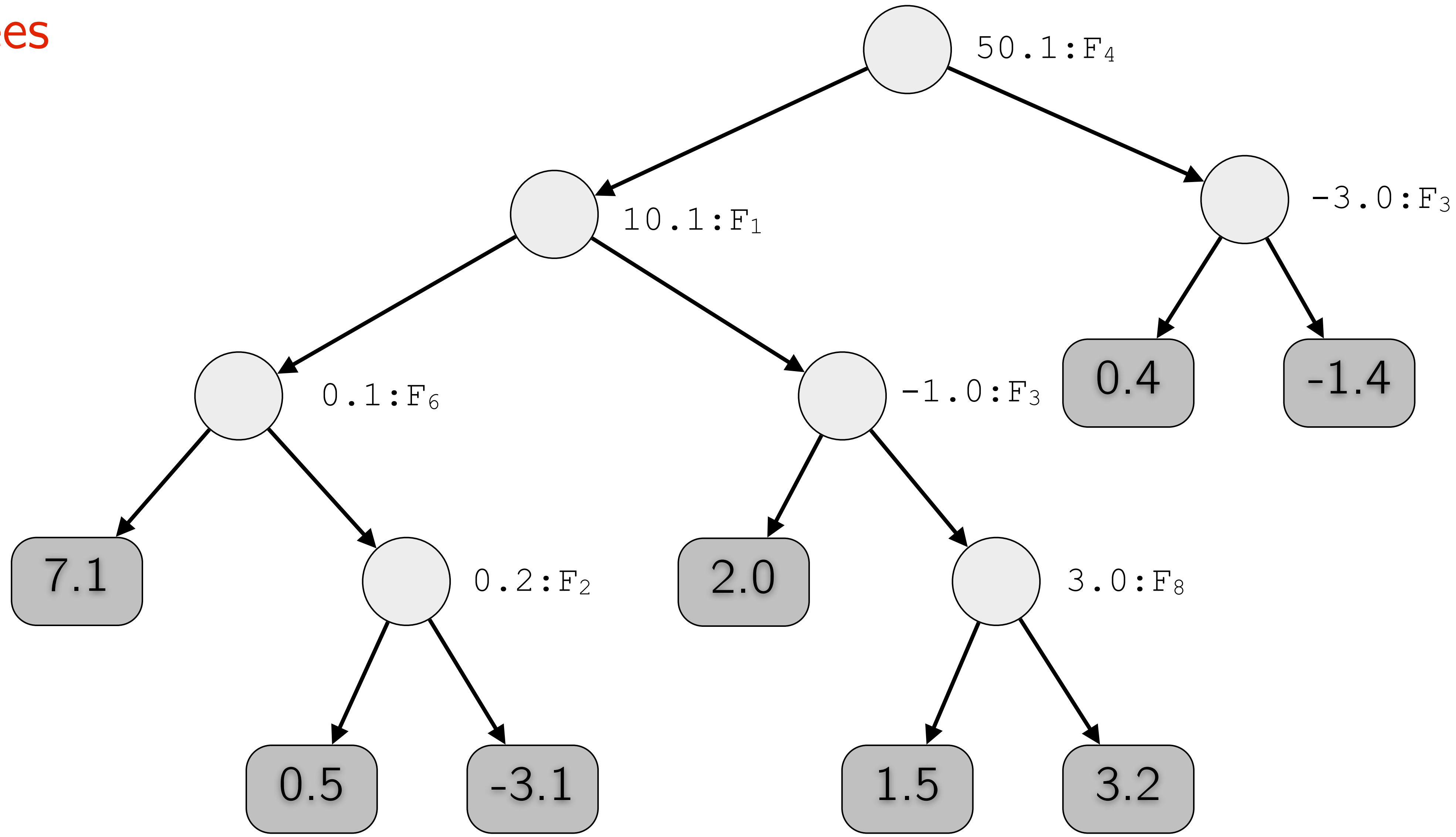
features = 100–1000

leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees



Documents

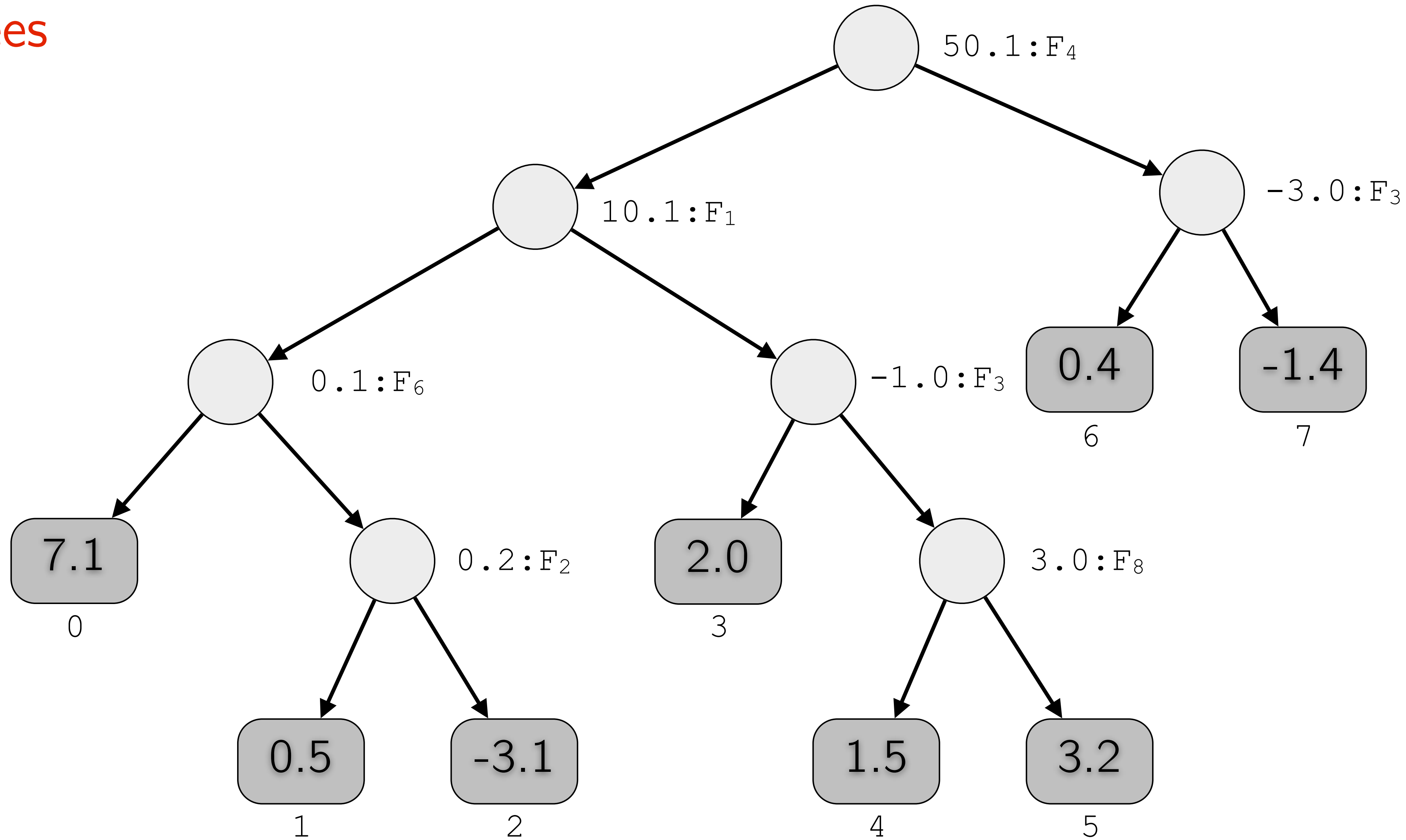
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

docs = >100K
trees = 1K–20K
features = 100–1000
leaves = 4–64

An alternative traversing algorithm

Trees



Documents

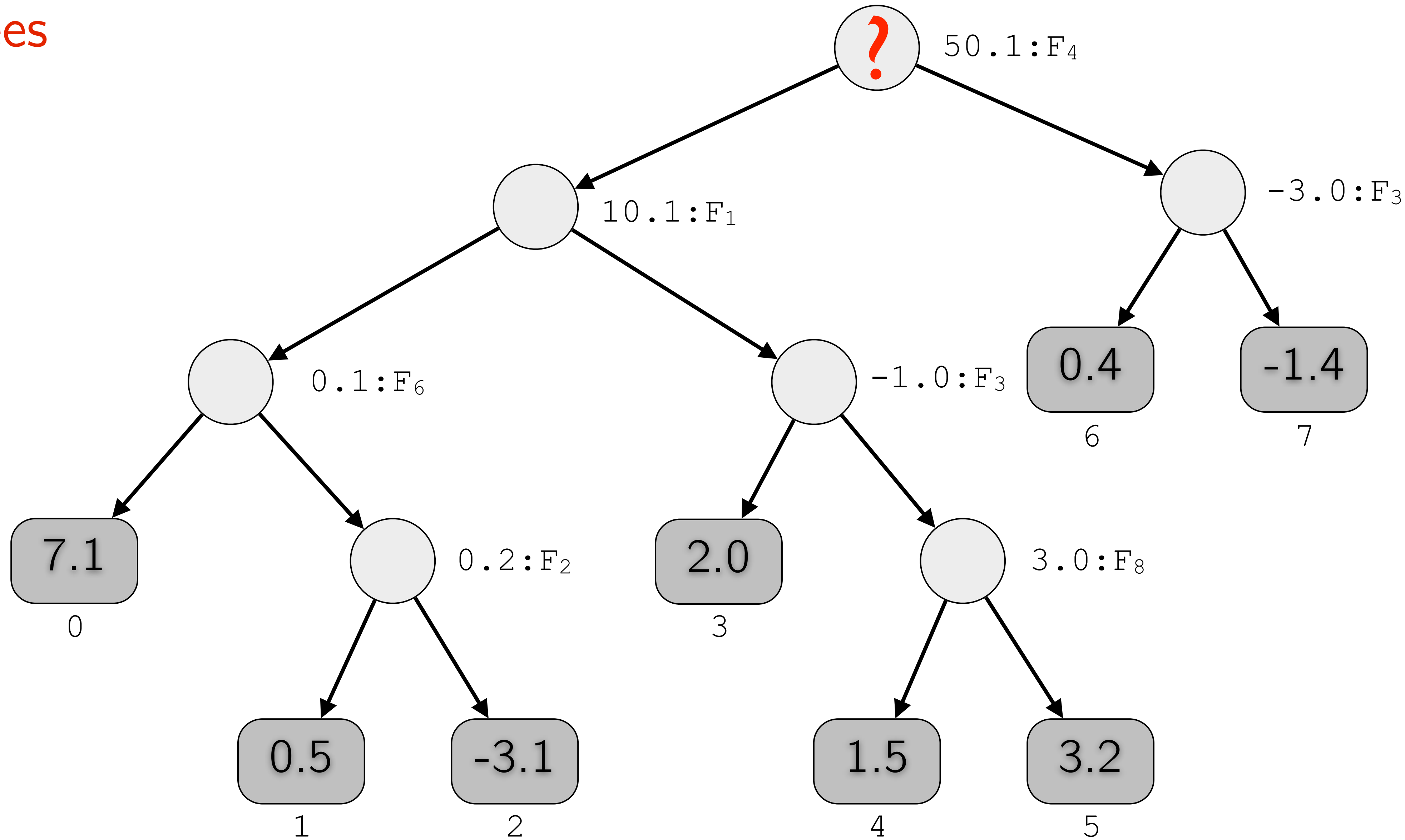
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees



Documents

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

...

docs = >100K

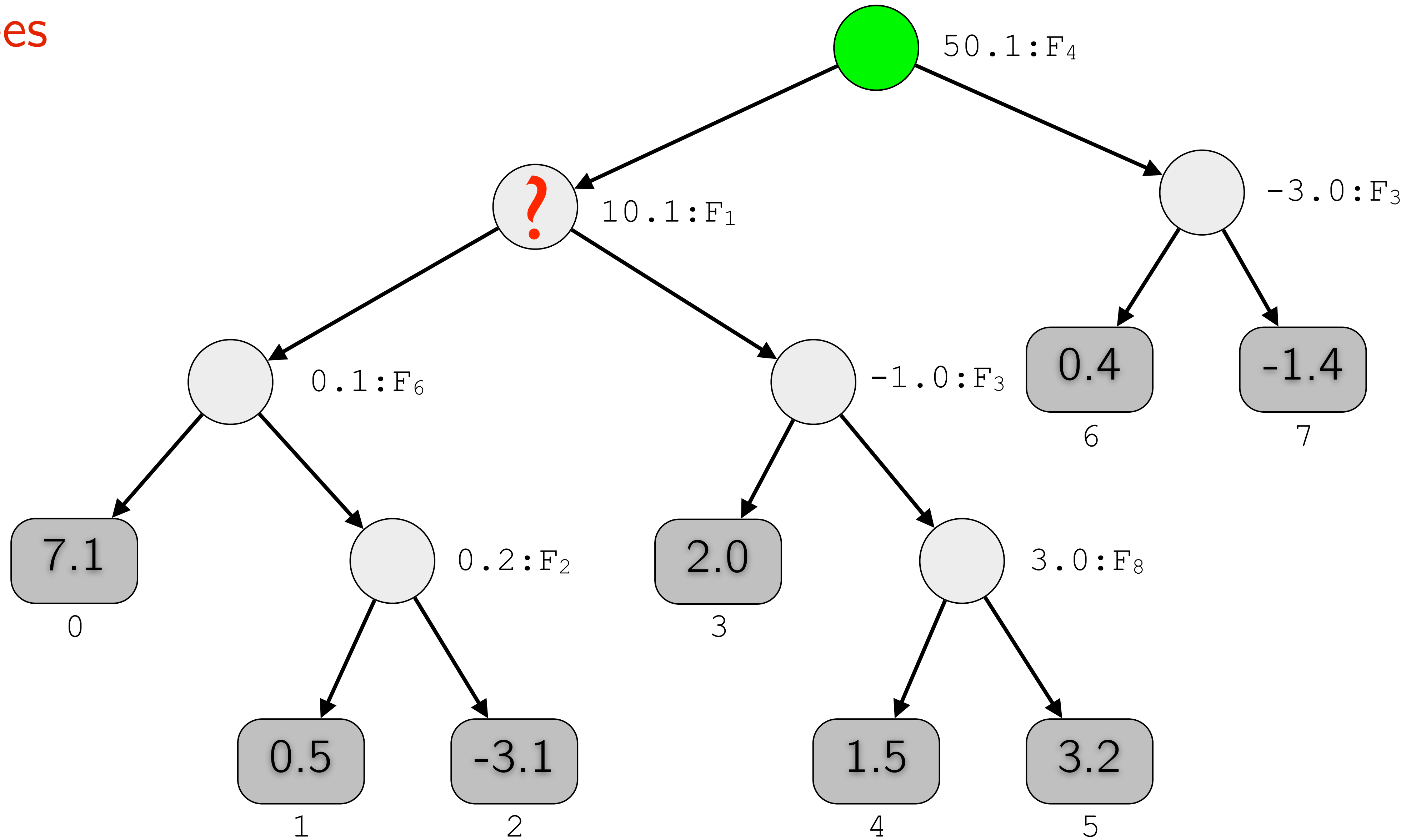
trees = 1K–20K

features = 100–1000

leaves = 4–64

An alternative traversing algorithm

Trees



Documents

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

...

\# docs = >100K

\# trees = 1K–20K

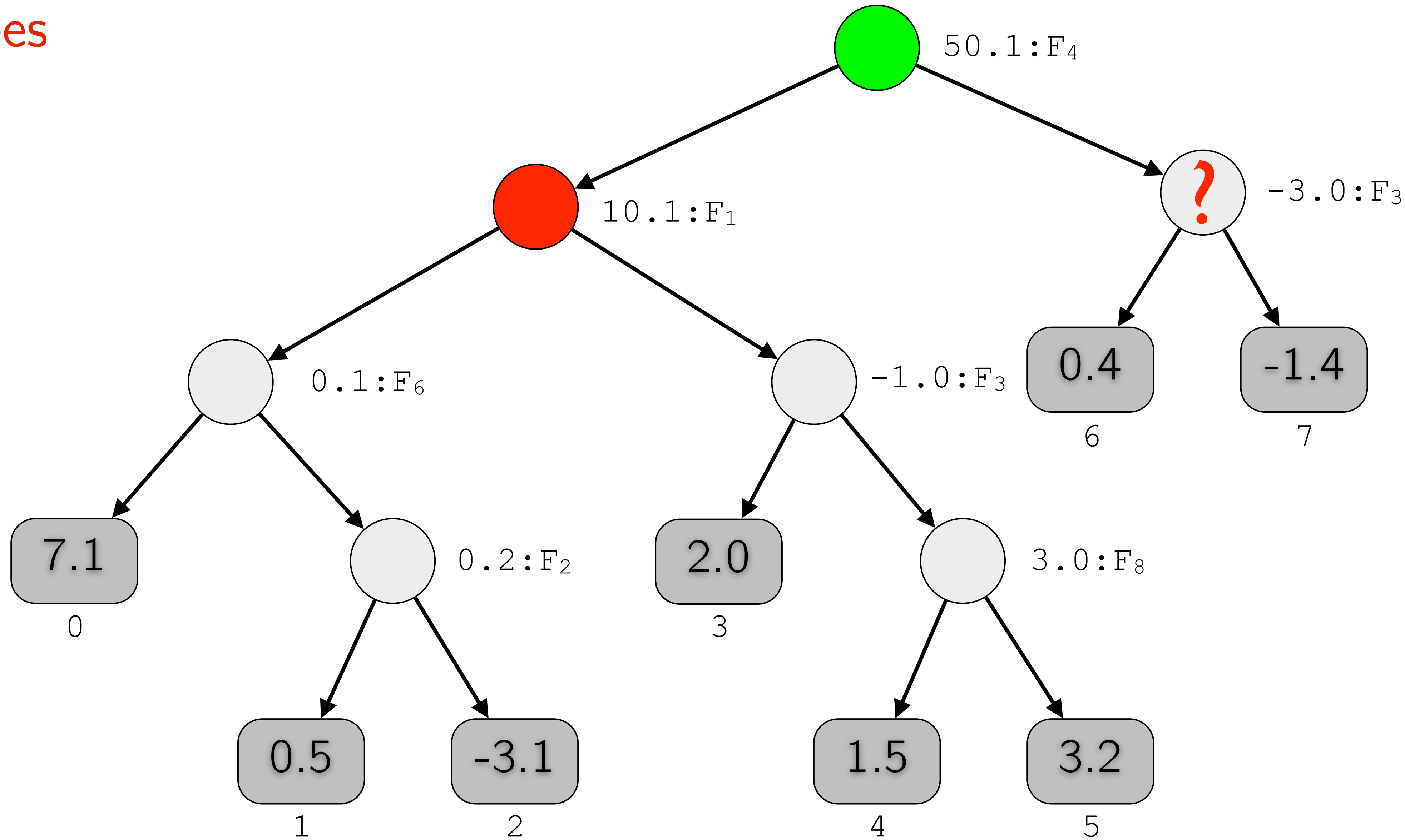
\# features = 100–1000

\# leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees



Documents

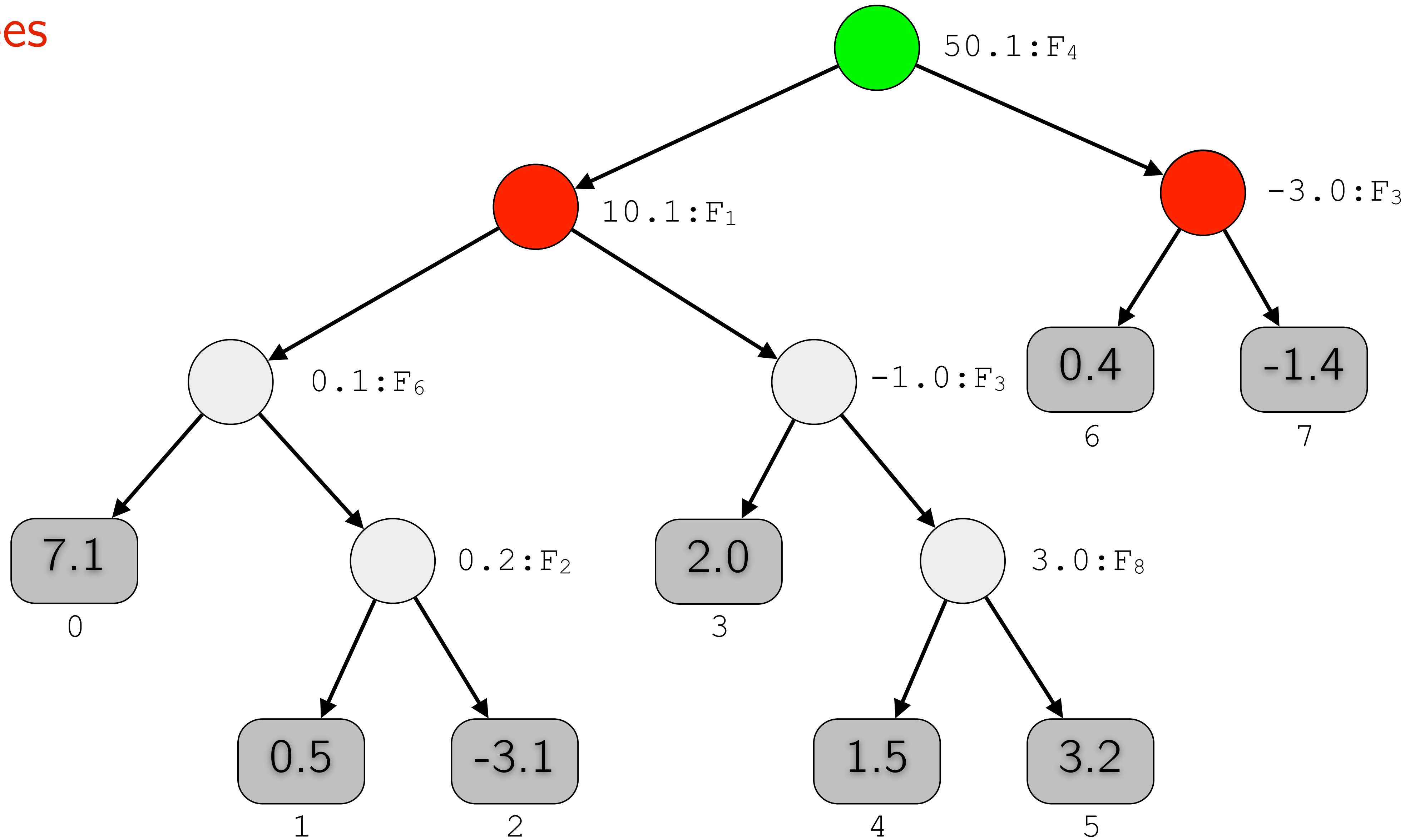
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees



Documents

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

...

docs = >100K

trees = 1K–20K

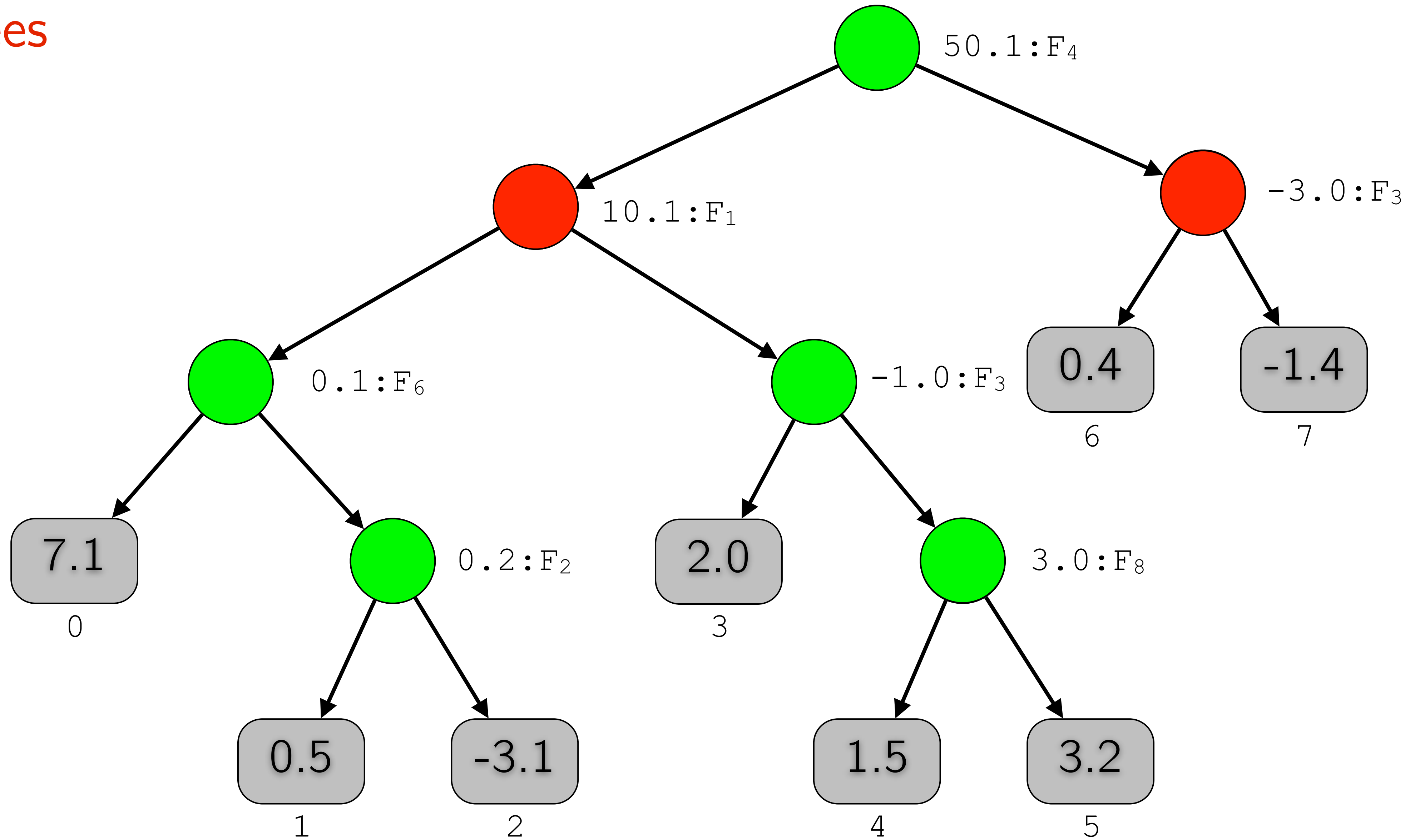
features = 100–1000

leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees



Documents

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

…

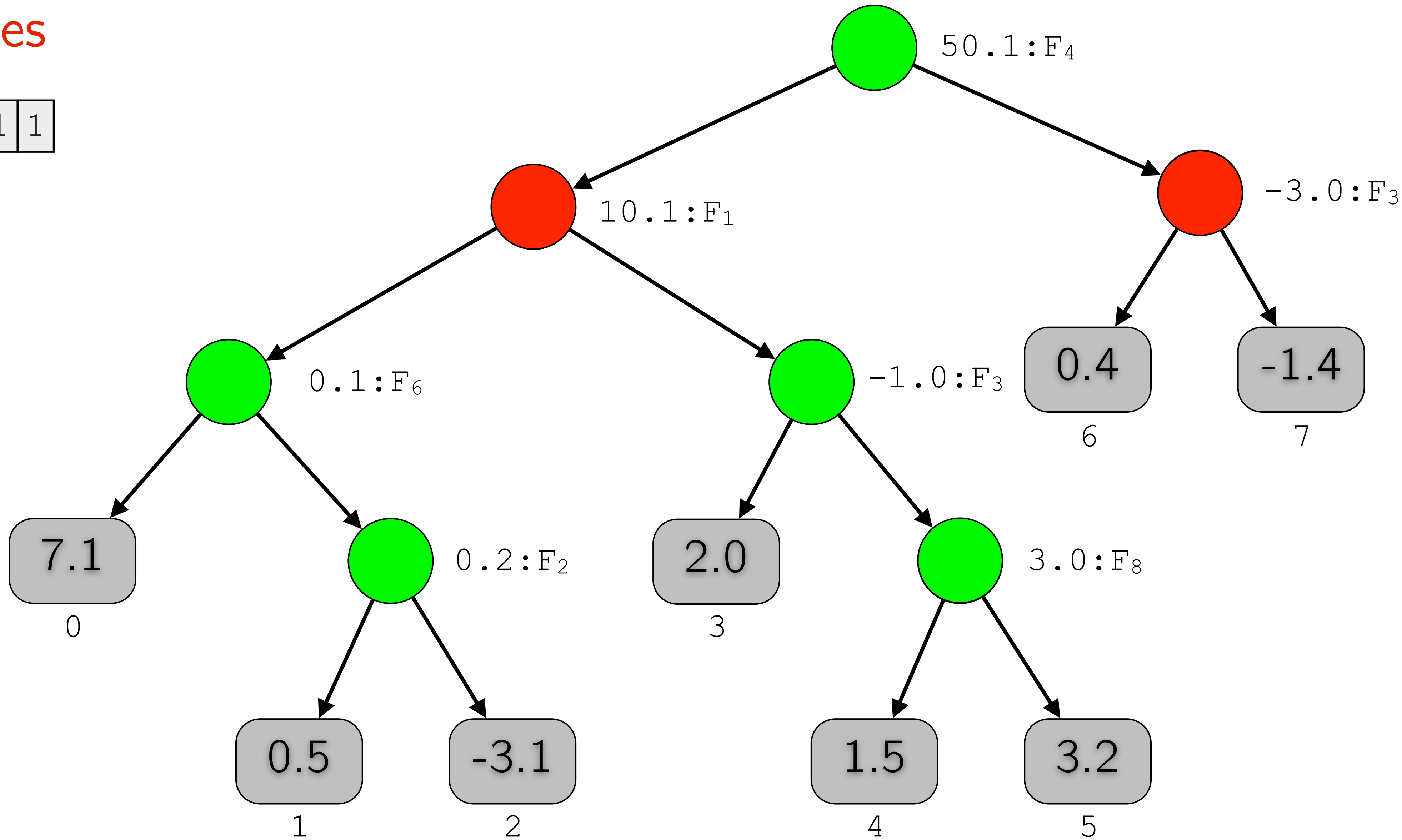
docs = >100K
trees = 1K–20K
features = 100–1000
leaves = 4–64

An alternative traversing algorithm

Trees

Result

1	1	1	1	1	1	1	1



Documents

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

...

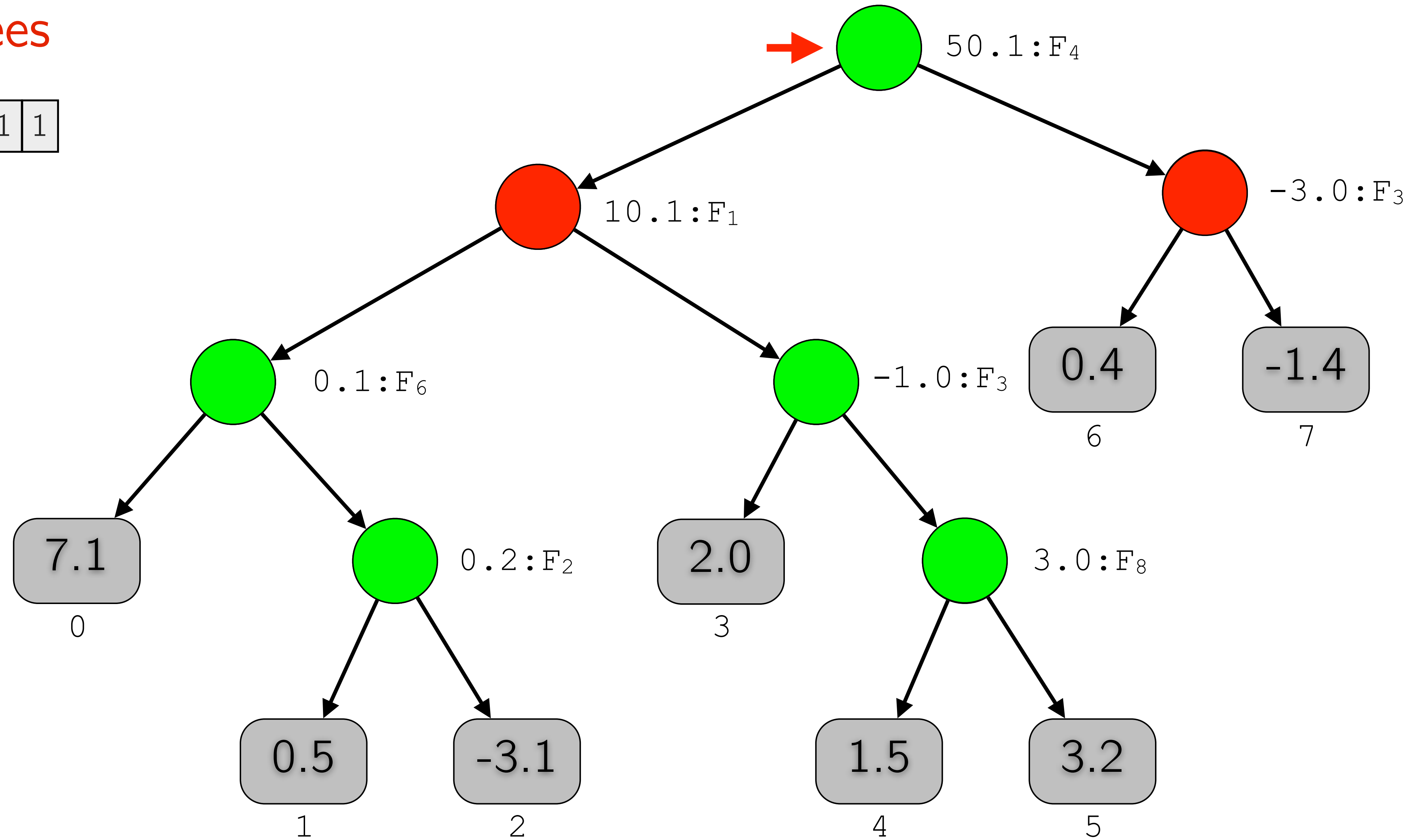
- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees

Result

1	1	1	1	1	1	1	1



Documents

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

...

docs = >100K

trees = 1K–20K

features = 100–1000

leaves = 4–64

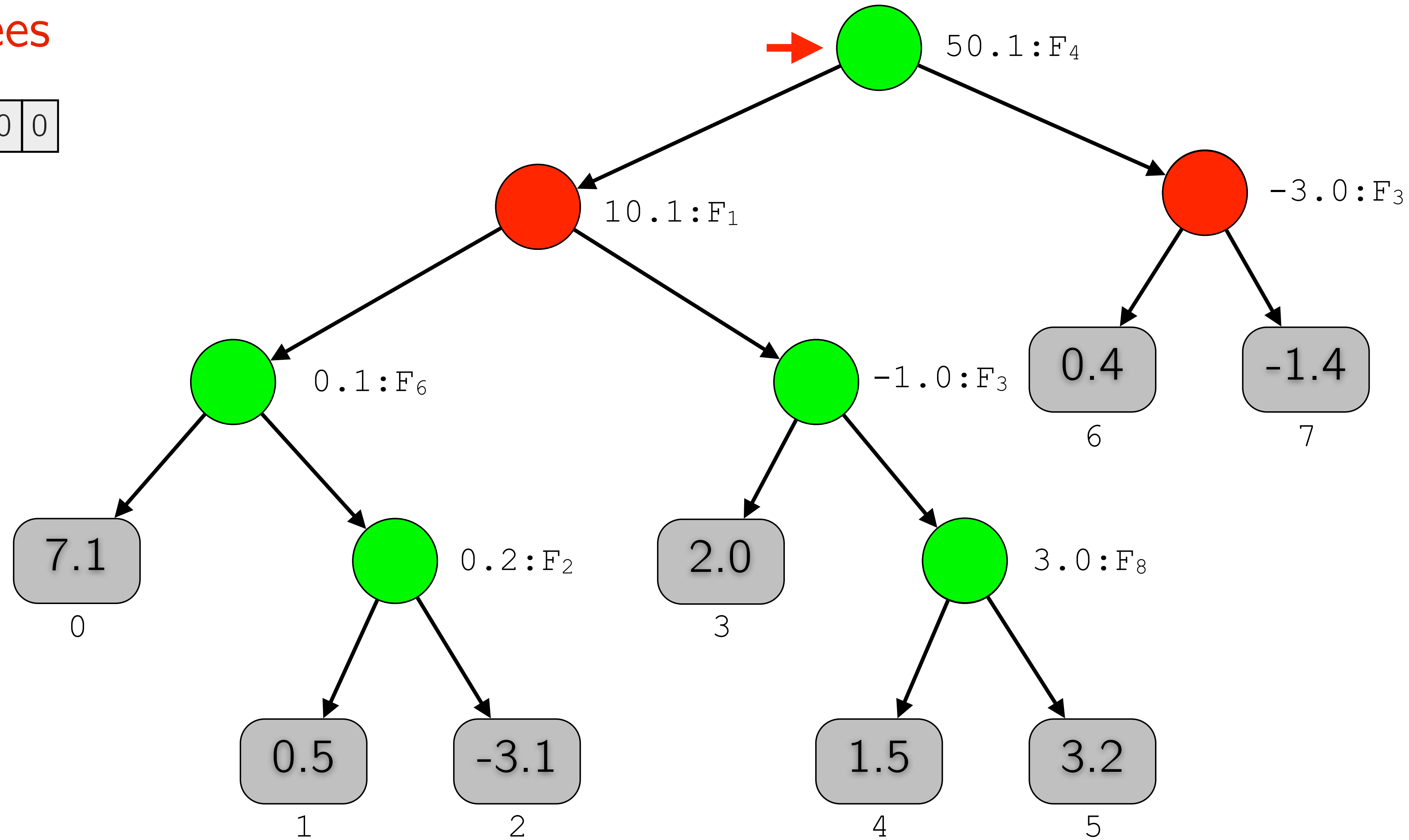
ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees

Result

1	1	1	1	1	1	0	0



Documents

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

...

docs = >100K

trees = 1K–20K

features = 100–1000

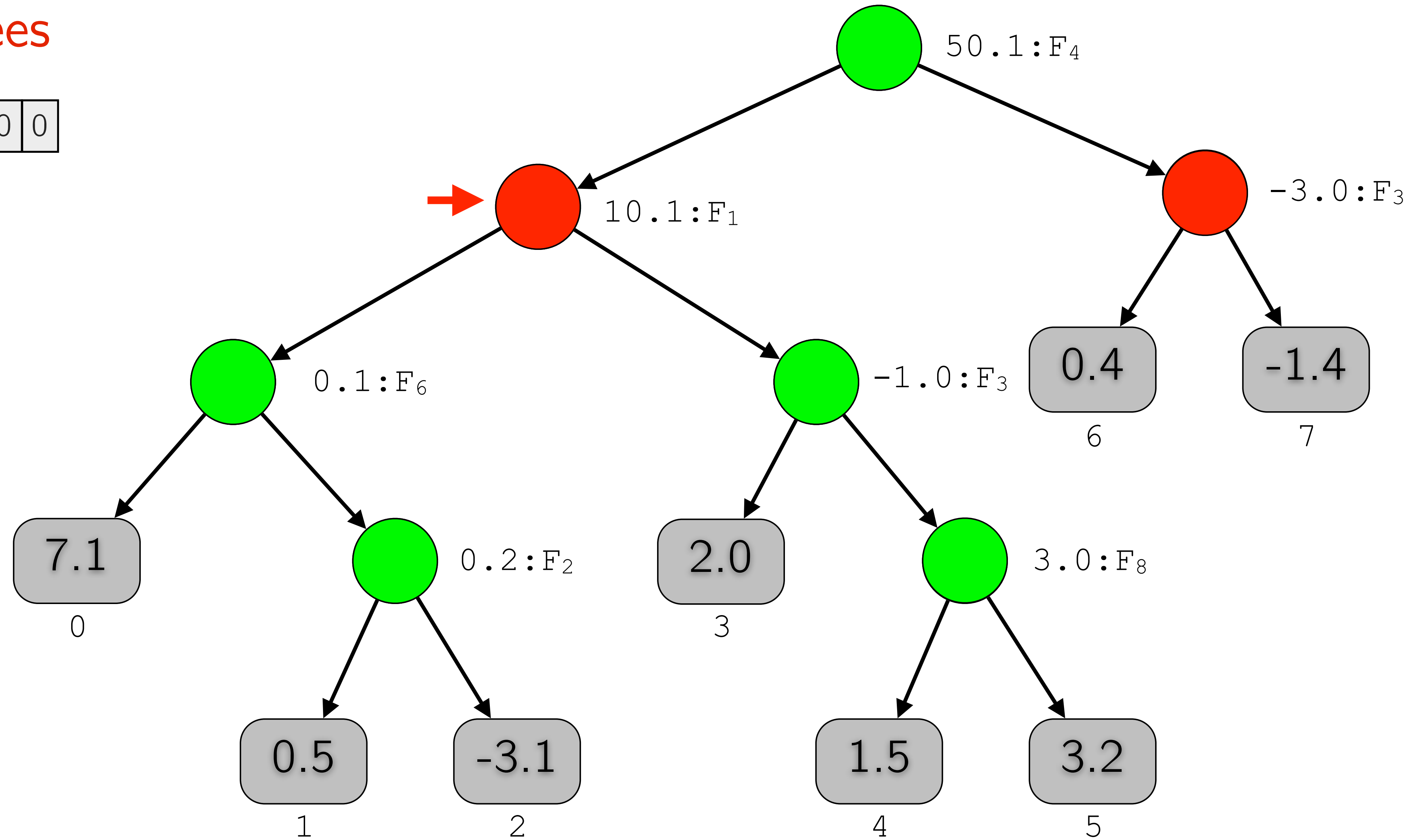
leaves = 4–64

An alternative traversing algorithm

Trees

Result

1	1	1	1	1	1	0	0



Documents

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

...

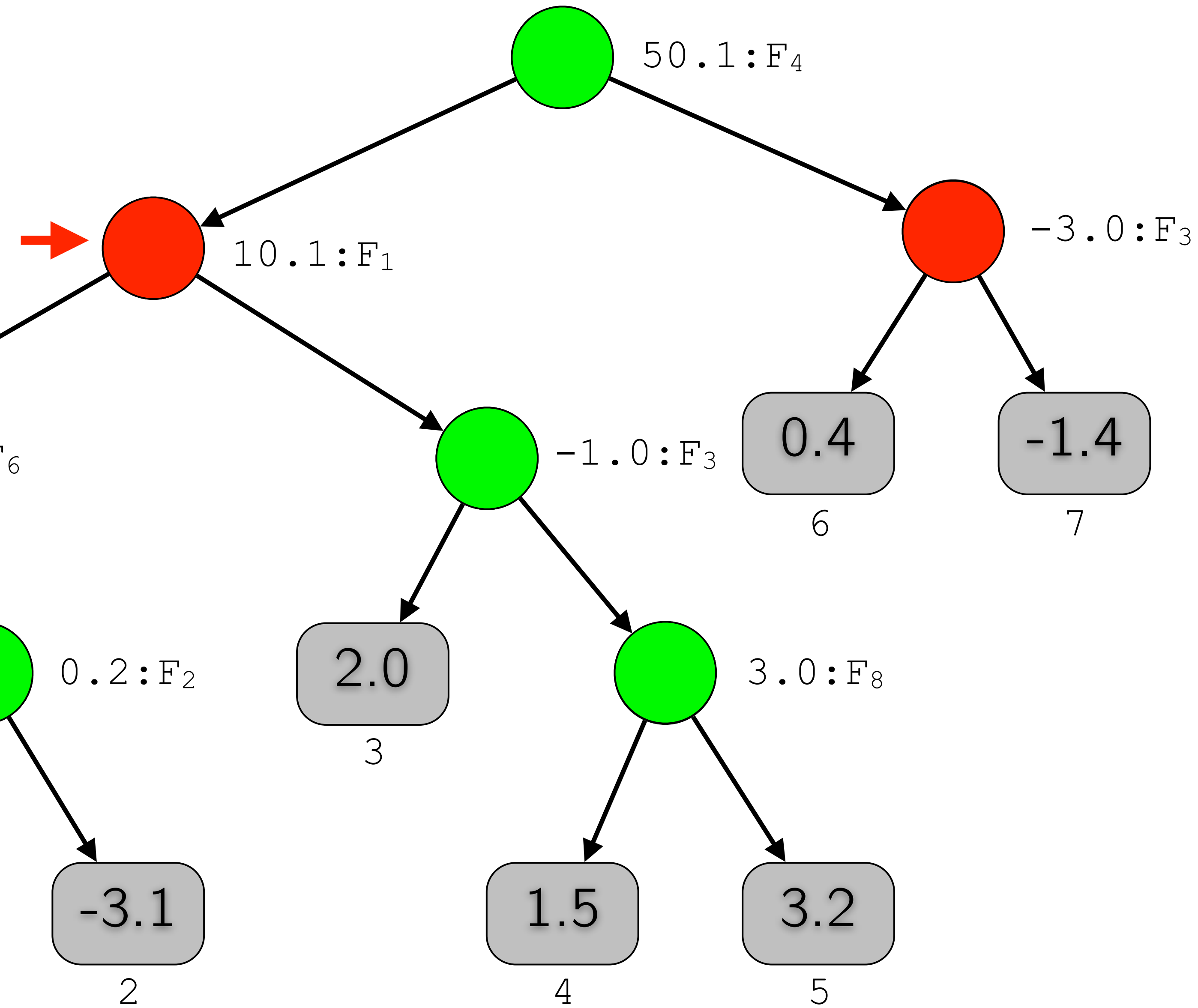
- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees

Result

0	0	0	1	1	1	0	0



Documents

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

…

docs = >100K
trees = 1K–20K
features = 100–1000
leaves = 4–64

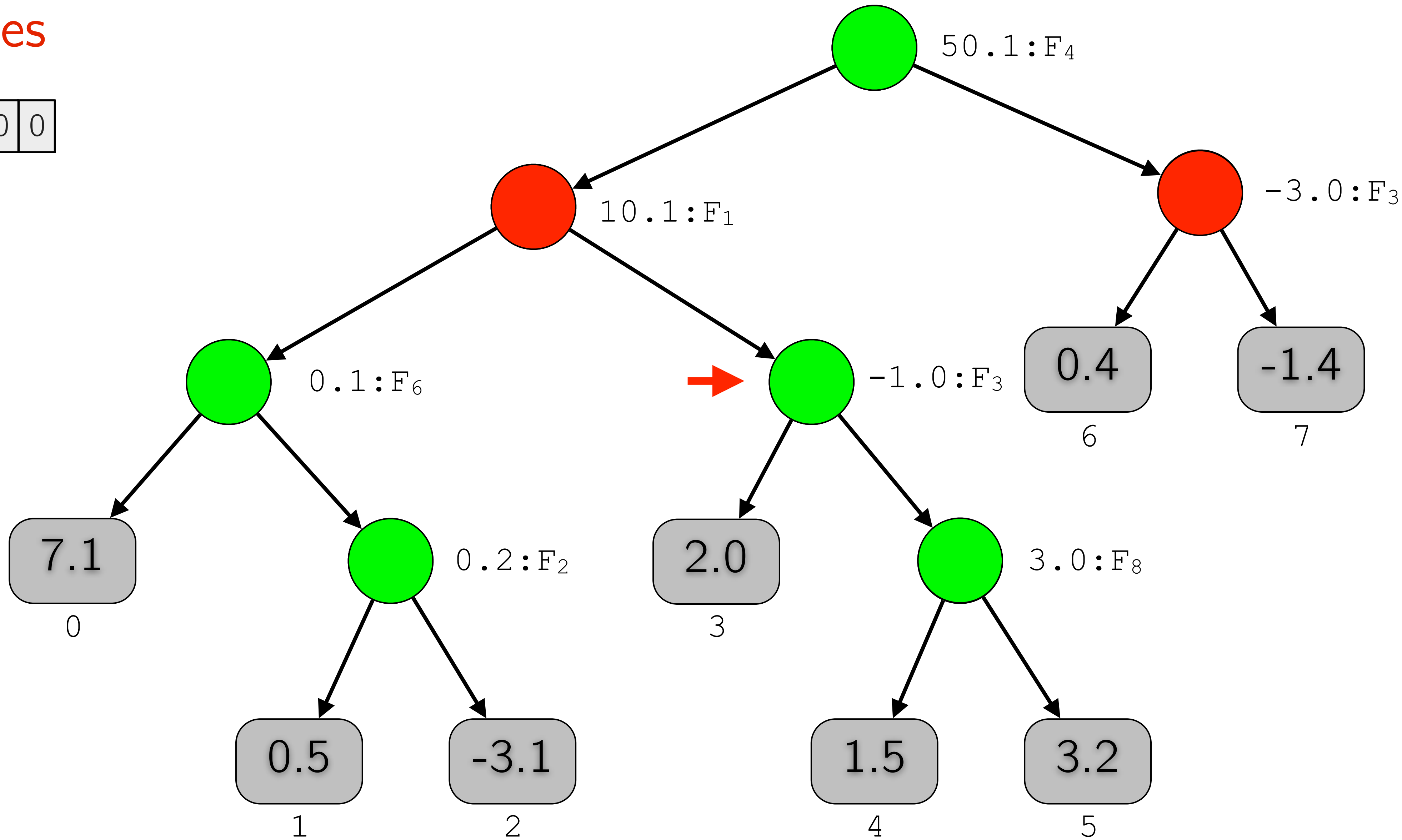
ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees

Result

0	0	0	1	1	1	0	0



Documents

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

...

docs = >100K

trees = 1K–20K

features = 100–1000

leaves = 4–64

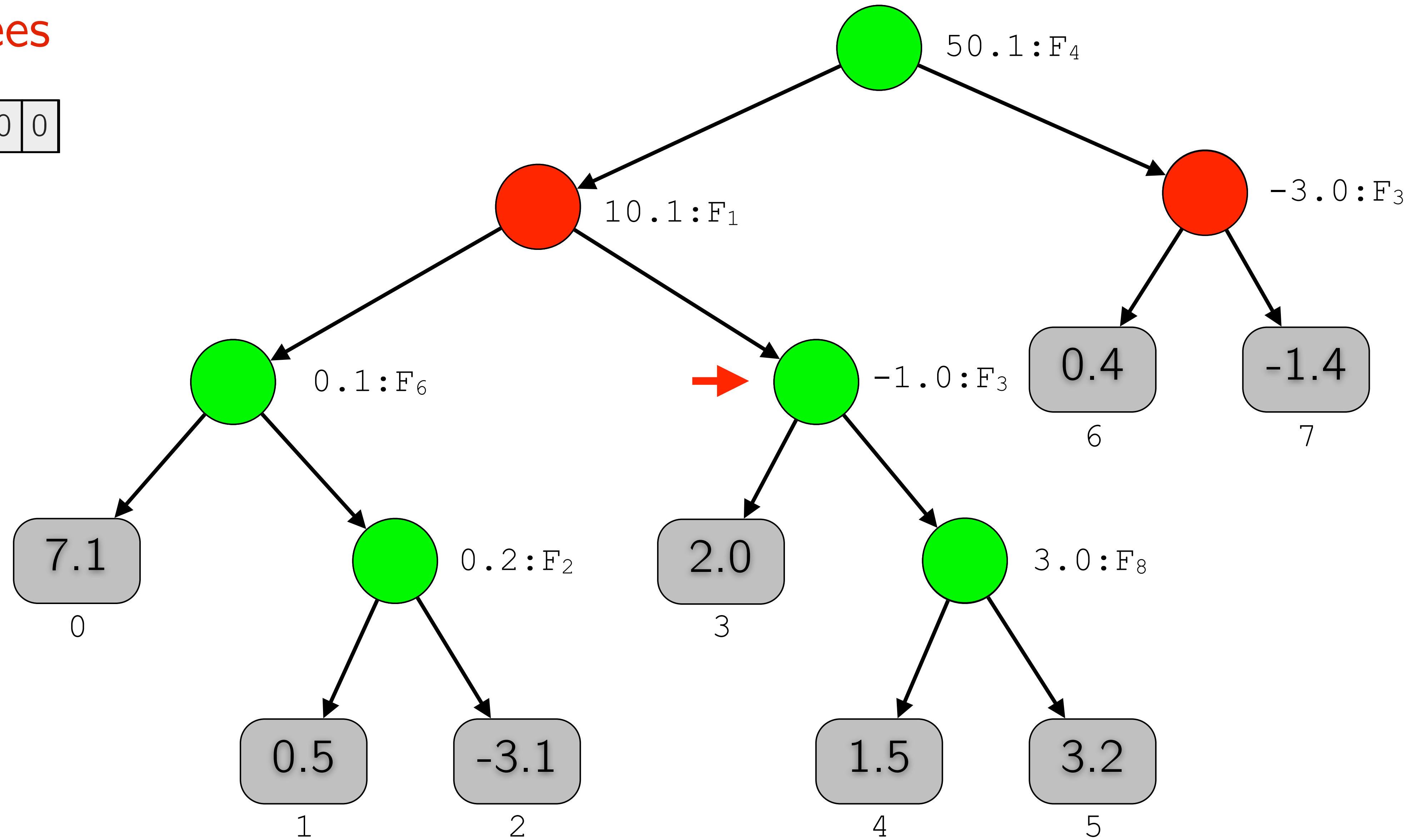
ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm

Trees

Result

0	0	0	1	0	0	0	0



Documents

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

...

docs = >100K

trees = 1K–20K

features = 100–1000

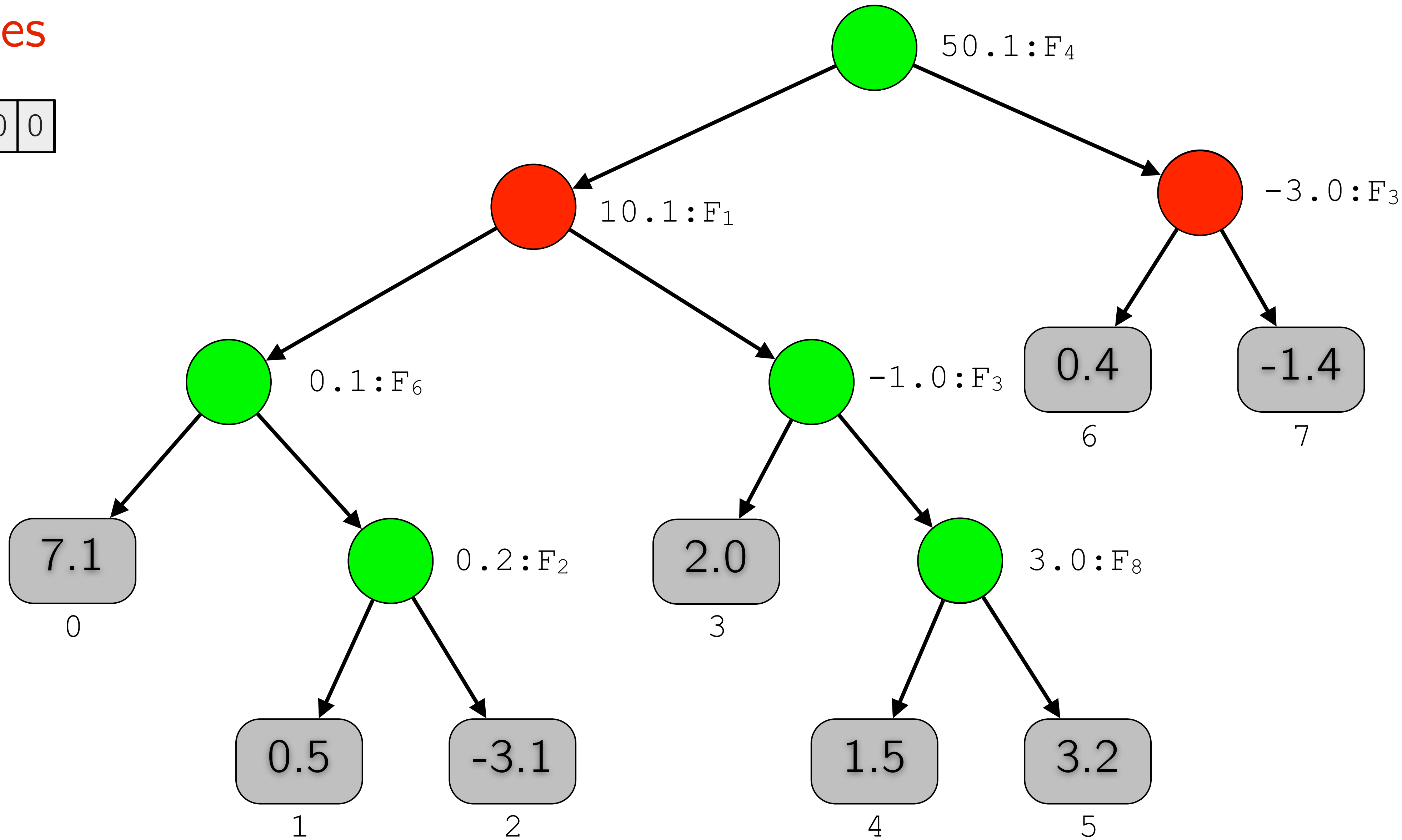
leaves = 4–64

An alternative traversing algorithm

Trees

Result

0	0	0	1	0	0	0	0



Documents

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

...

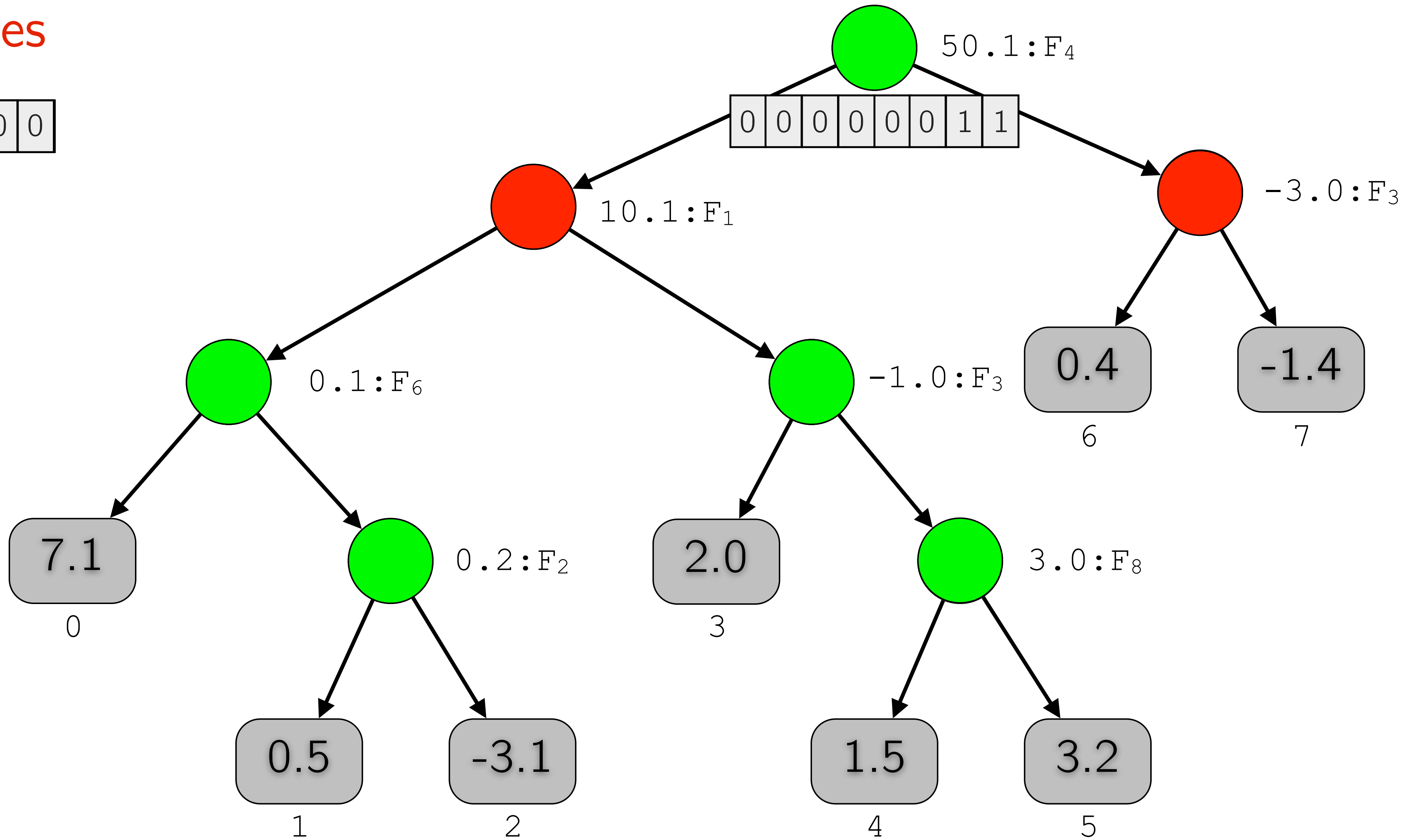
- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees

Result

0	0	0	1	0	0	0	0



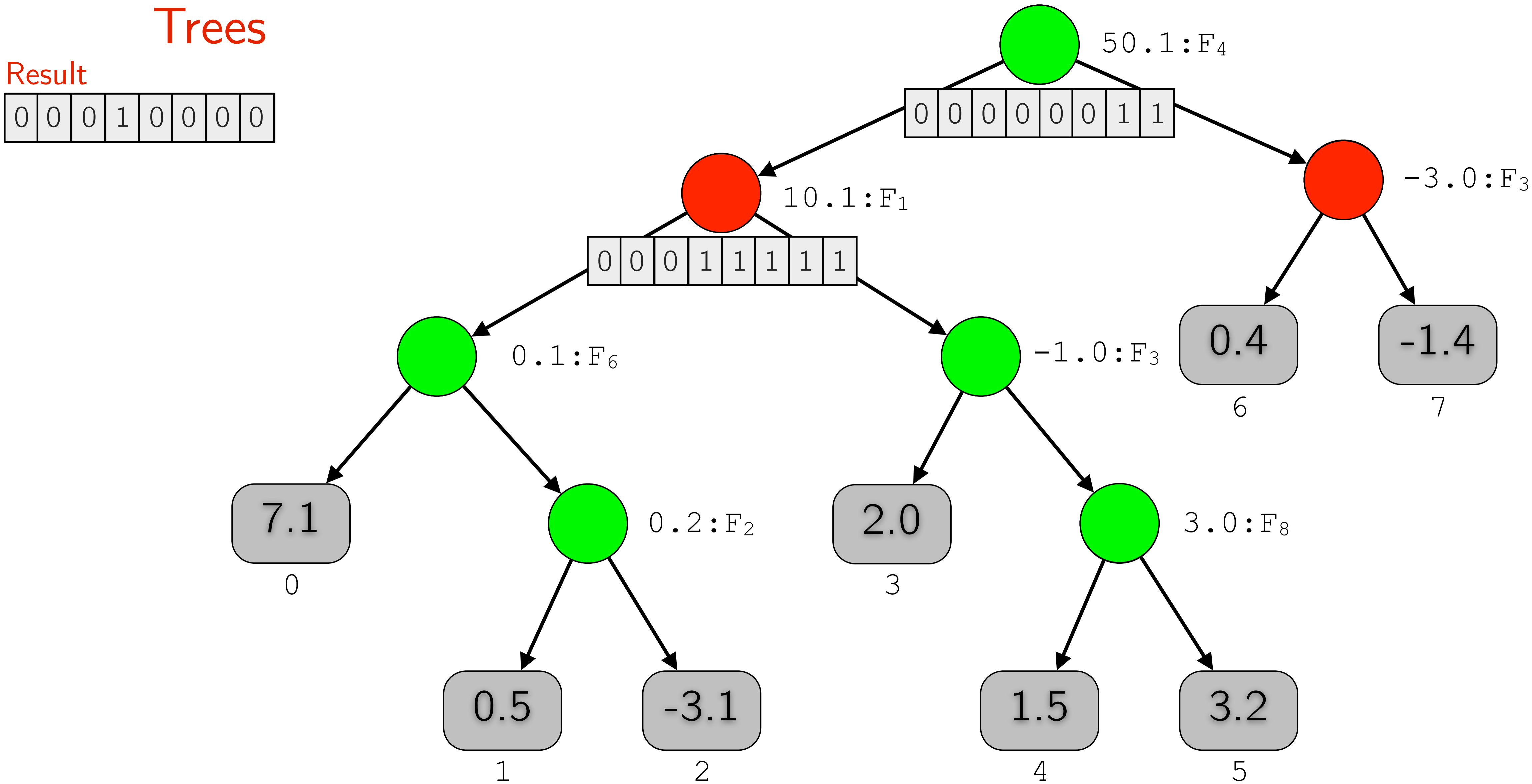
Documents

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

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm



Documents

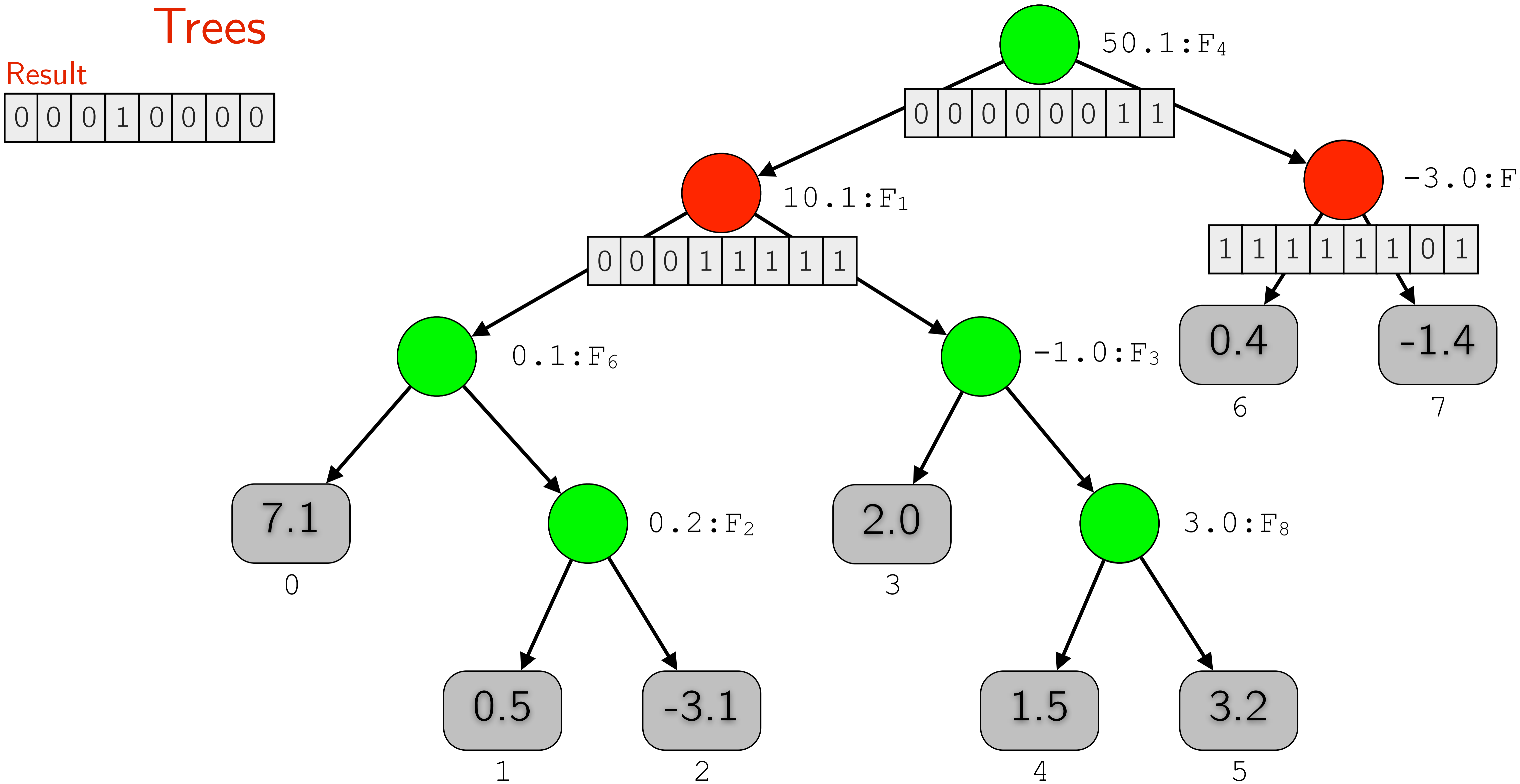
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees



Documents

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

...

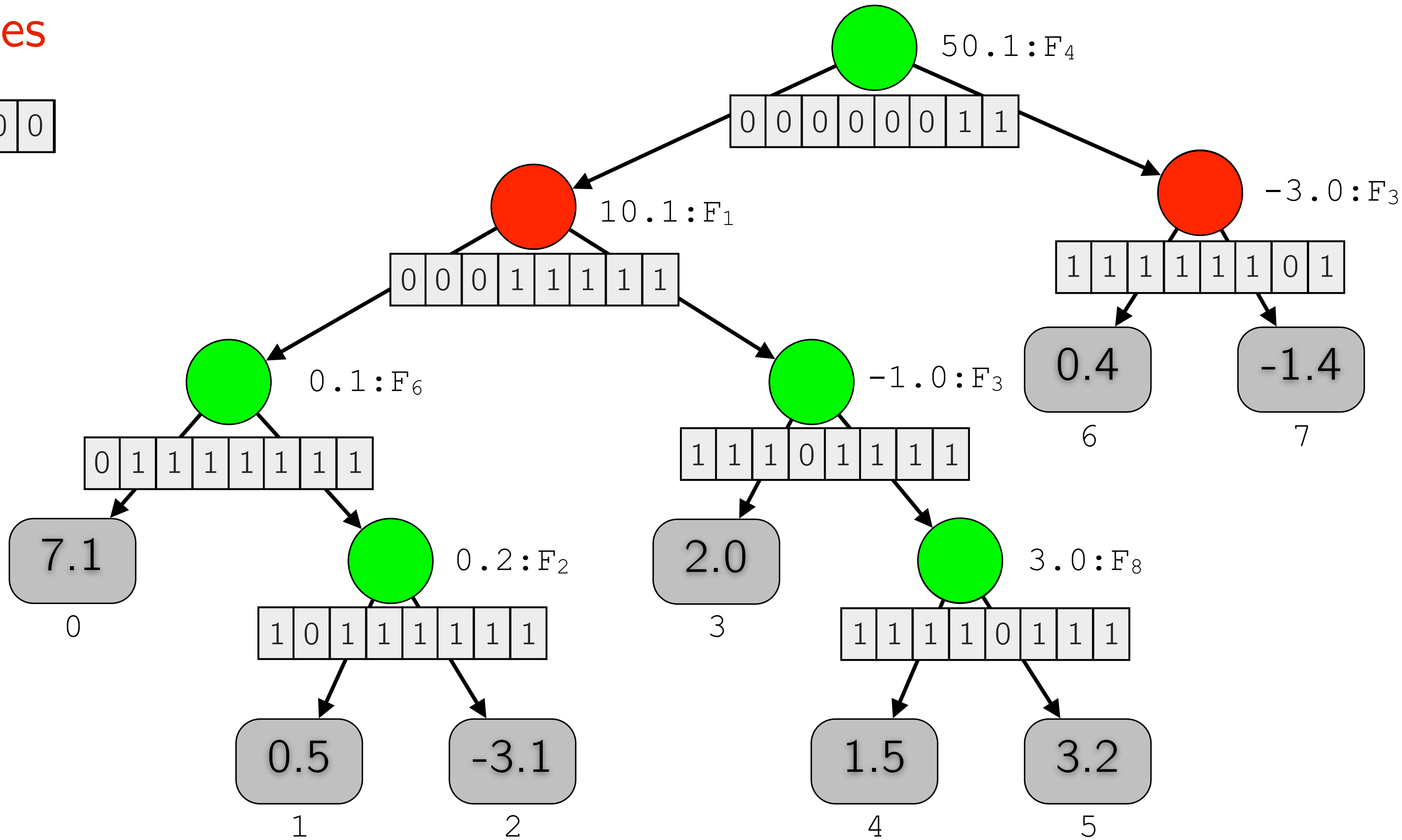
- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees

Result

0	0	0	1	0	0	0	0



Documents

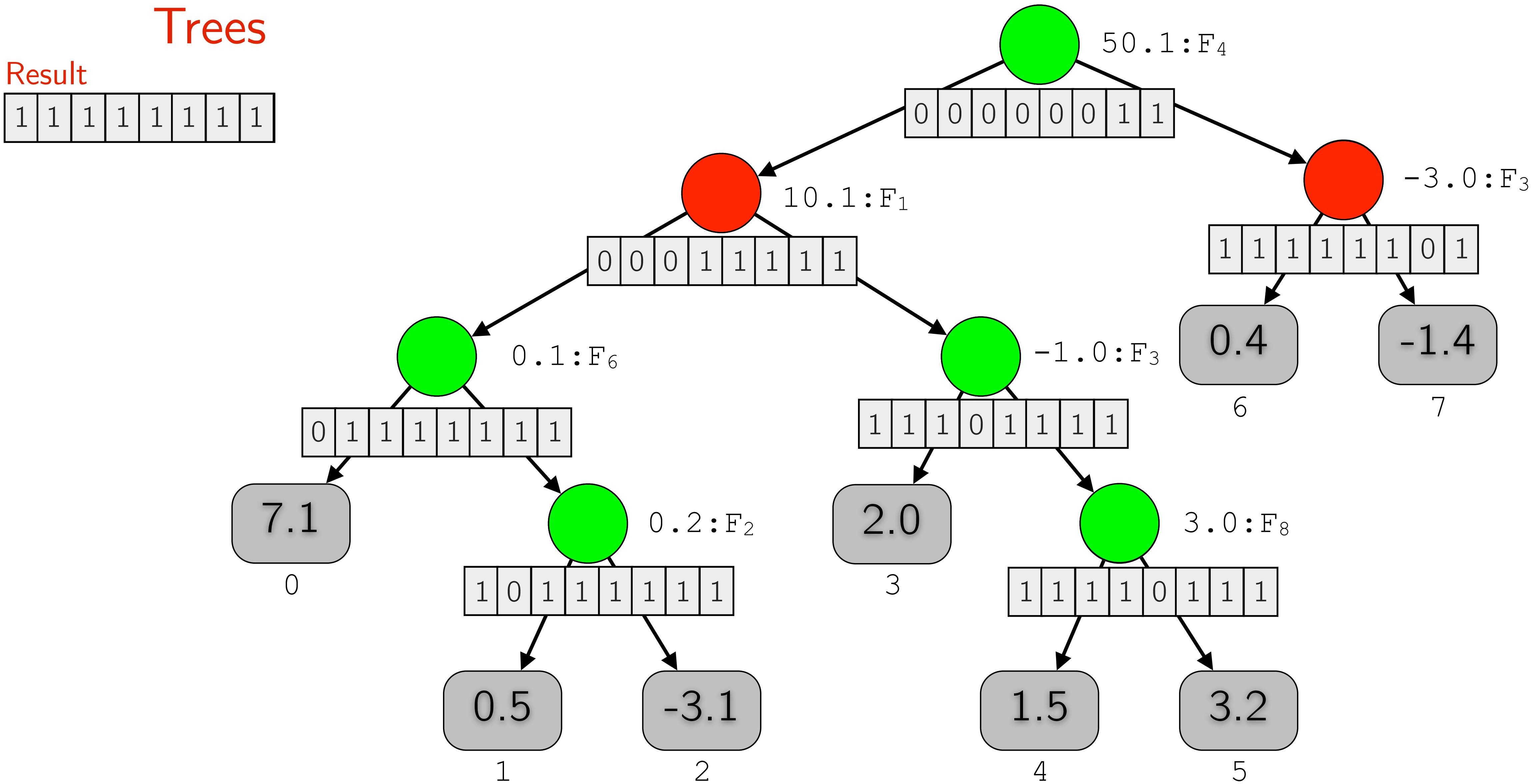
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm



Documents

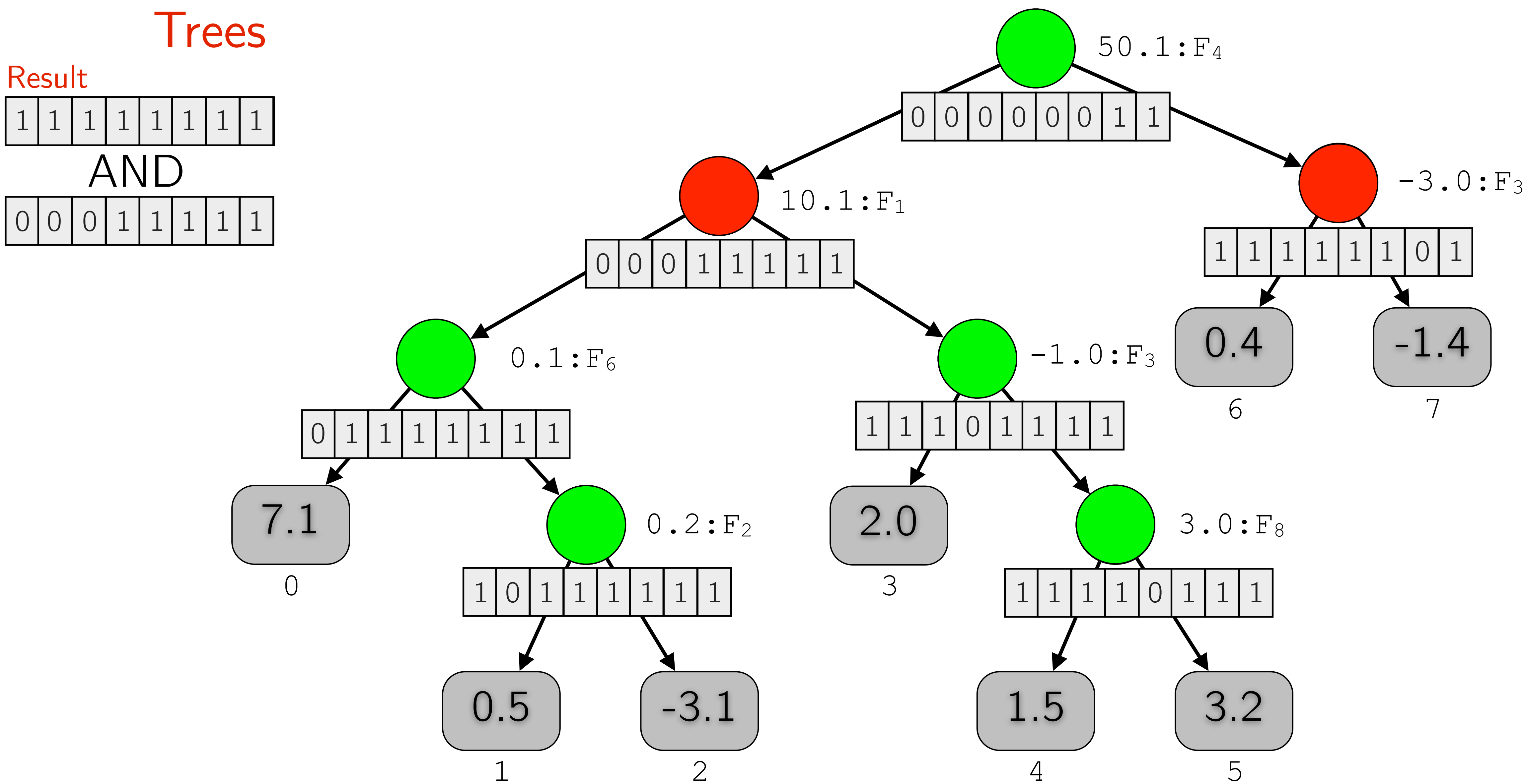
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

An alternative traversing algorithm



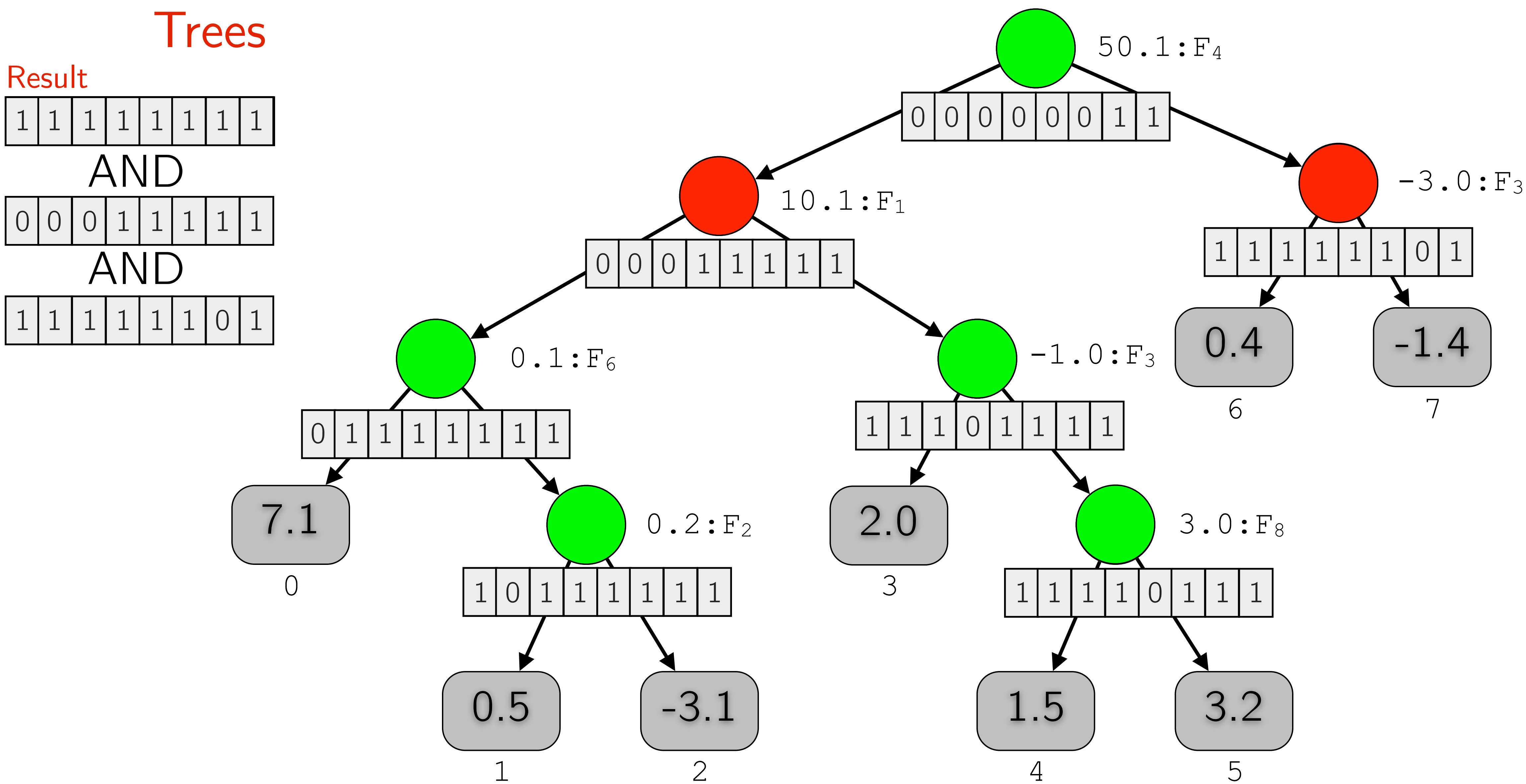
Documents

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

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm



Documents

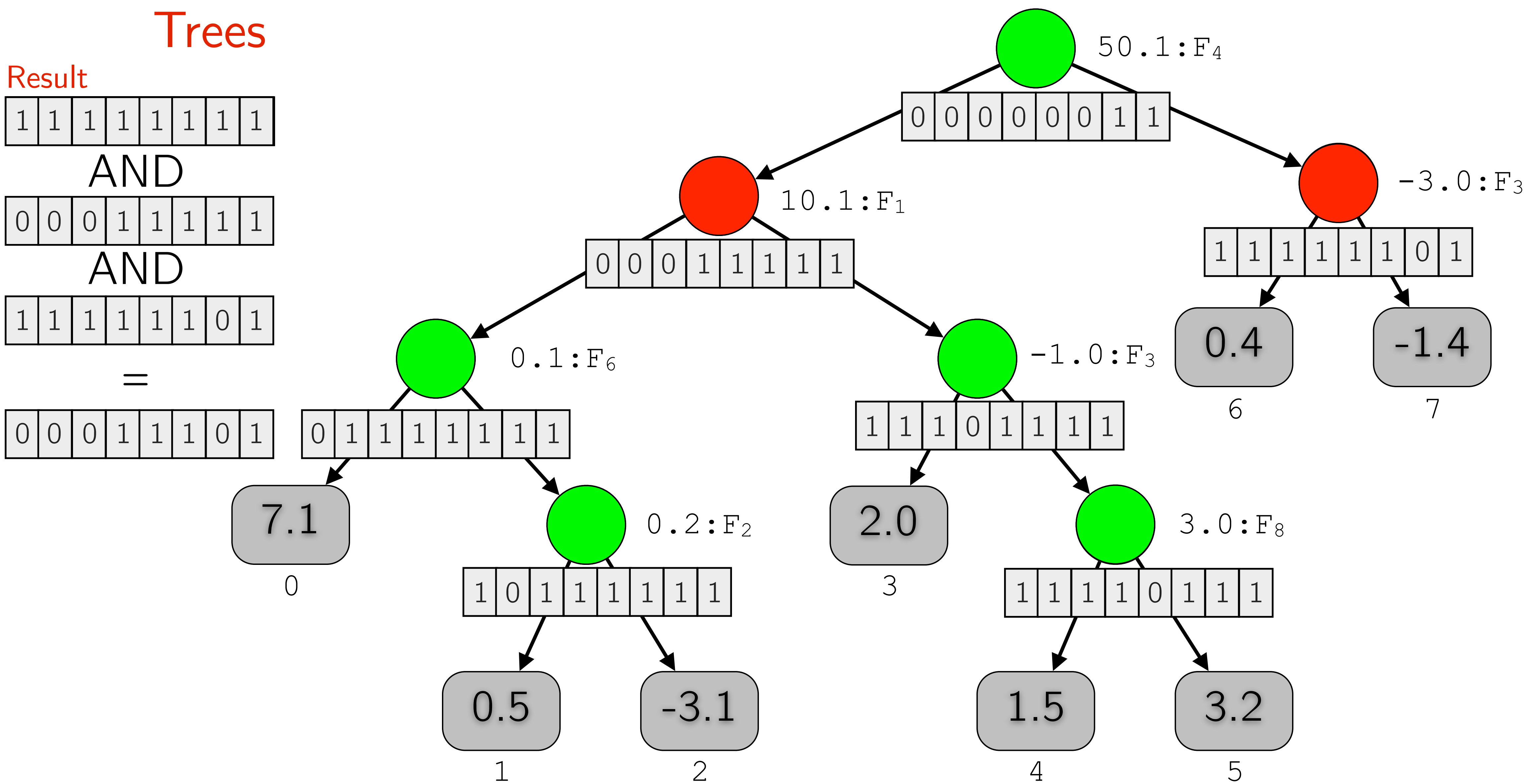
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees



Documents

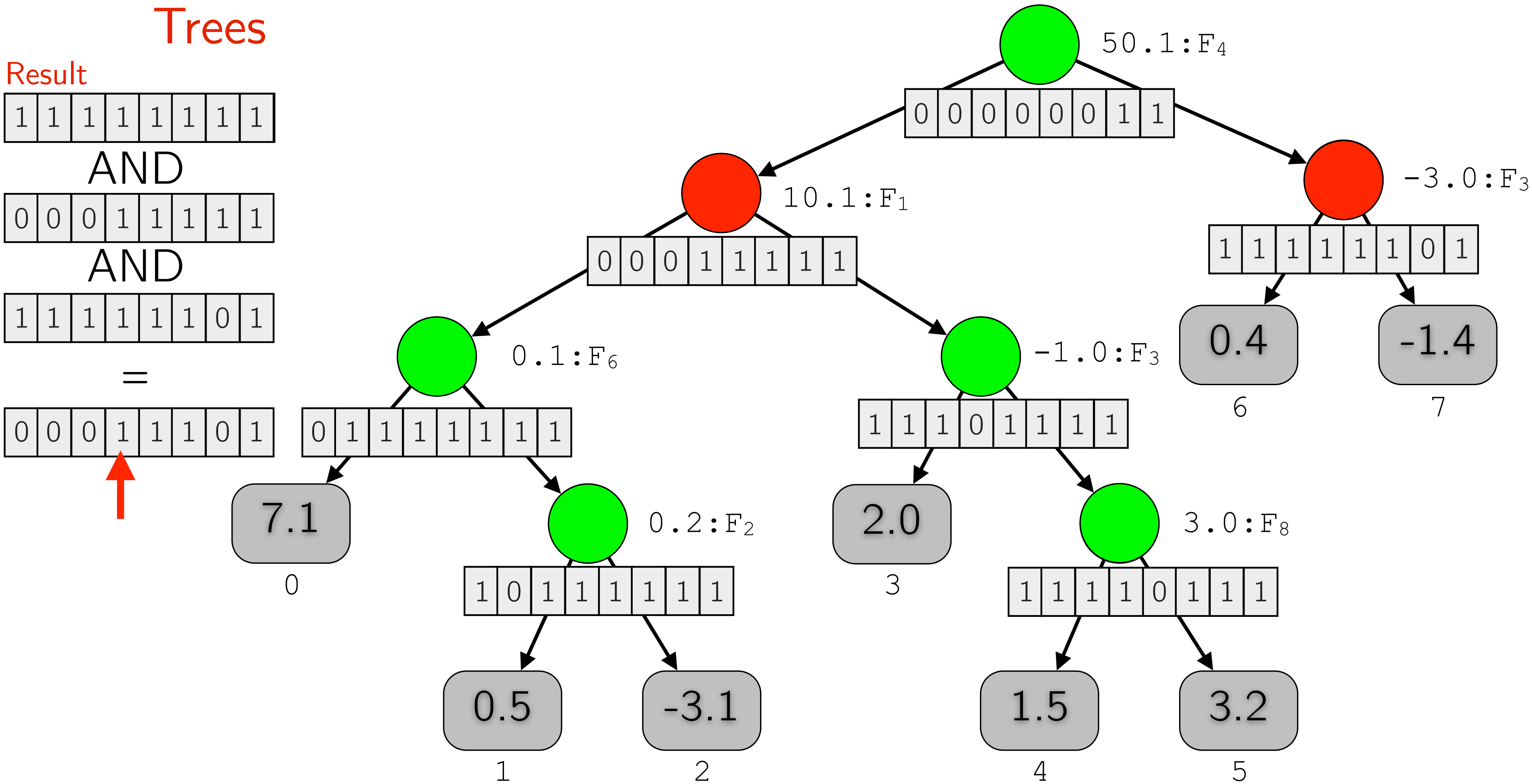
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees



Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm

Trees

Result

1	1	1	1	1	1	1	1

AND

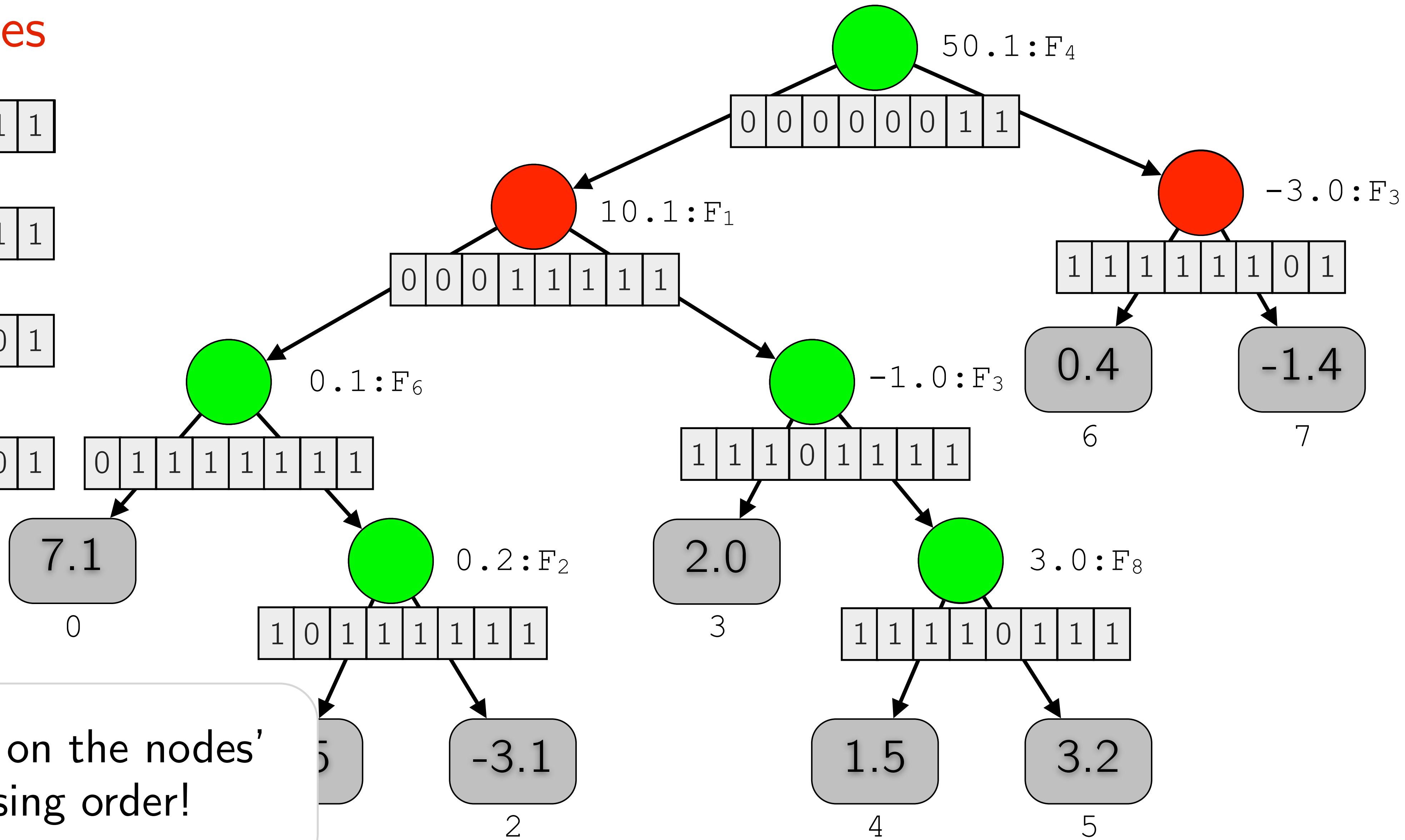
0	0	0	1	1	1	1	1

AND

1	1	1	1	1	1	0	1

=

0	0	0	1	1	1	0	1



Insensitive on the nodes' processing order!

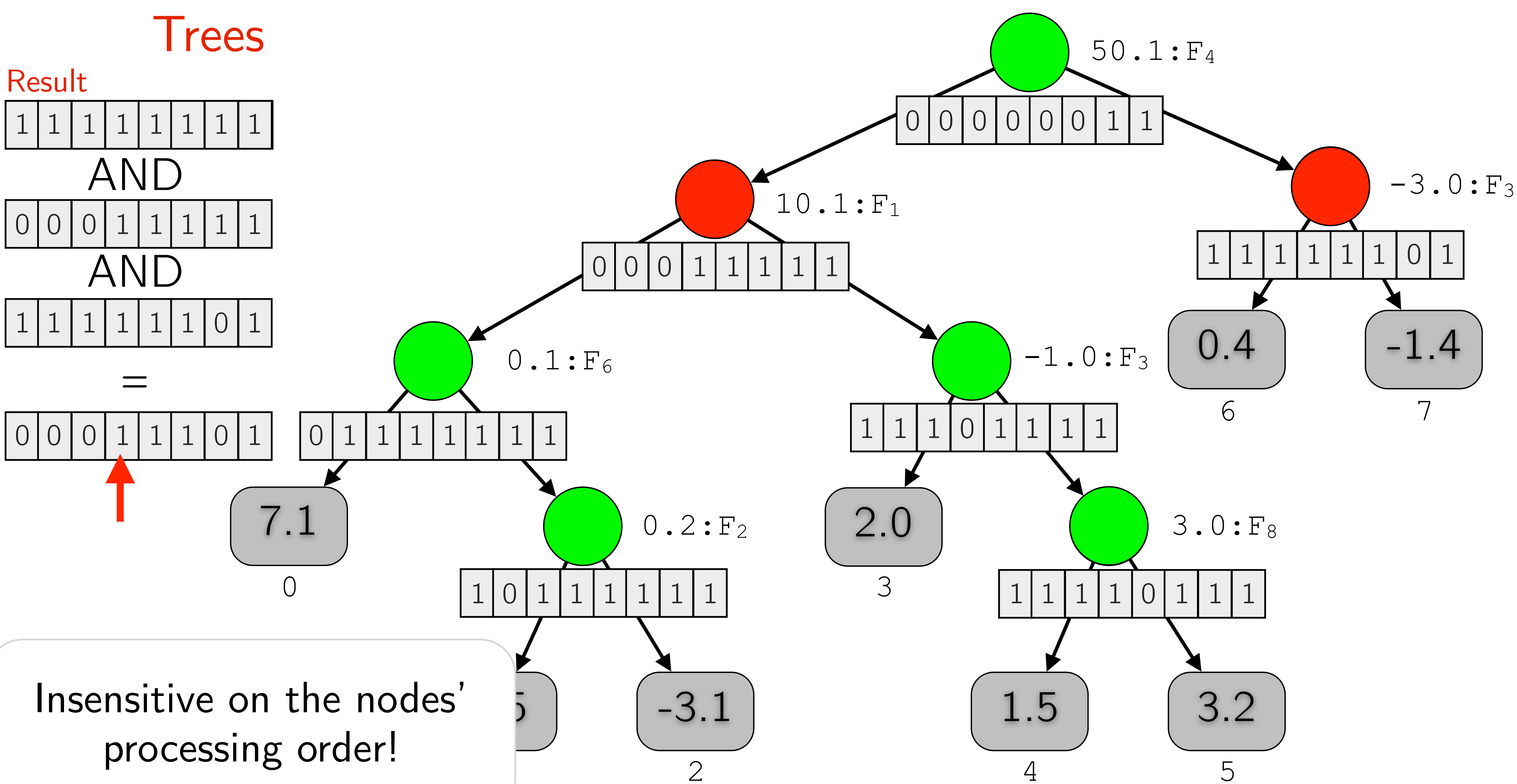
Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

...

- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

An alternative traversing algorithm



Insensitive on the nodes' processing order!

Documents

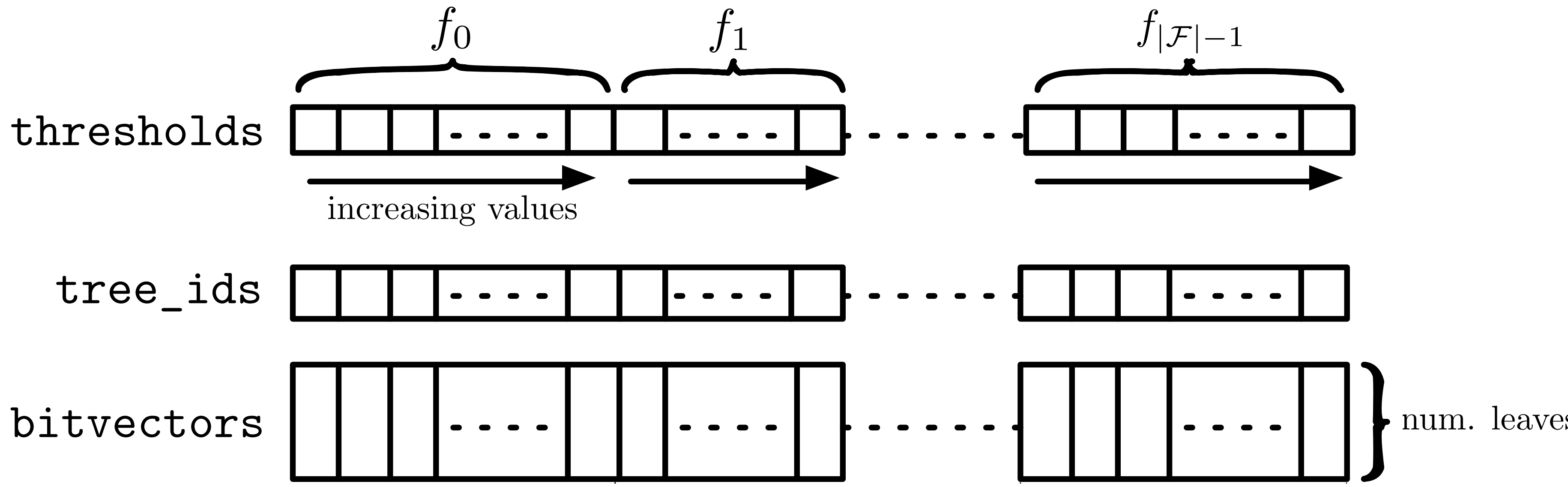
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
13.3	0.12	-1.2	43.9	11	-0.4	7.98	2.55
10.9	0.08	-1.1	42.9	15	-0.3	6.74	1.65
11.2	0.6	-0.2	54.1	13	-0.5	7.97	3

...

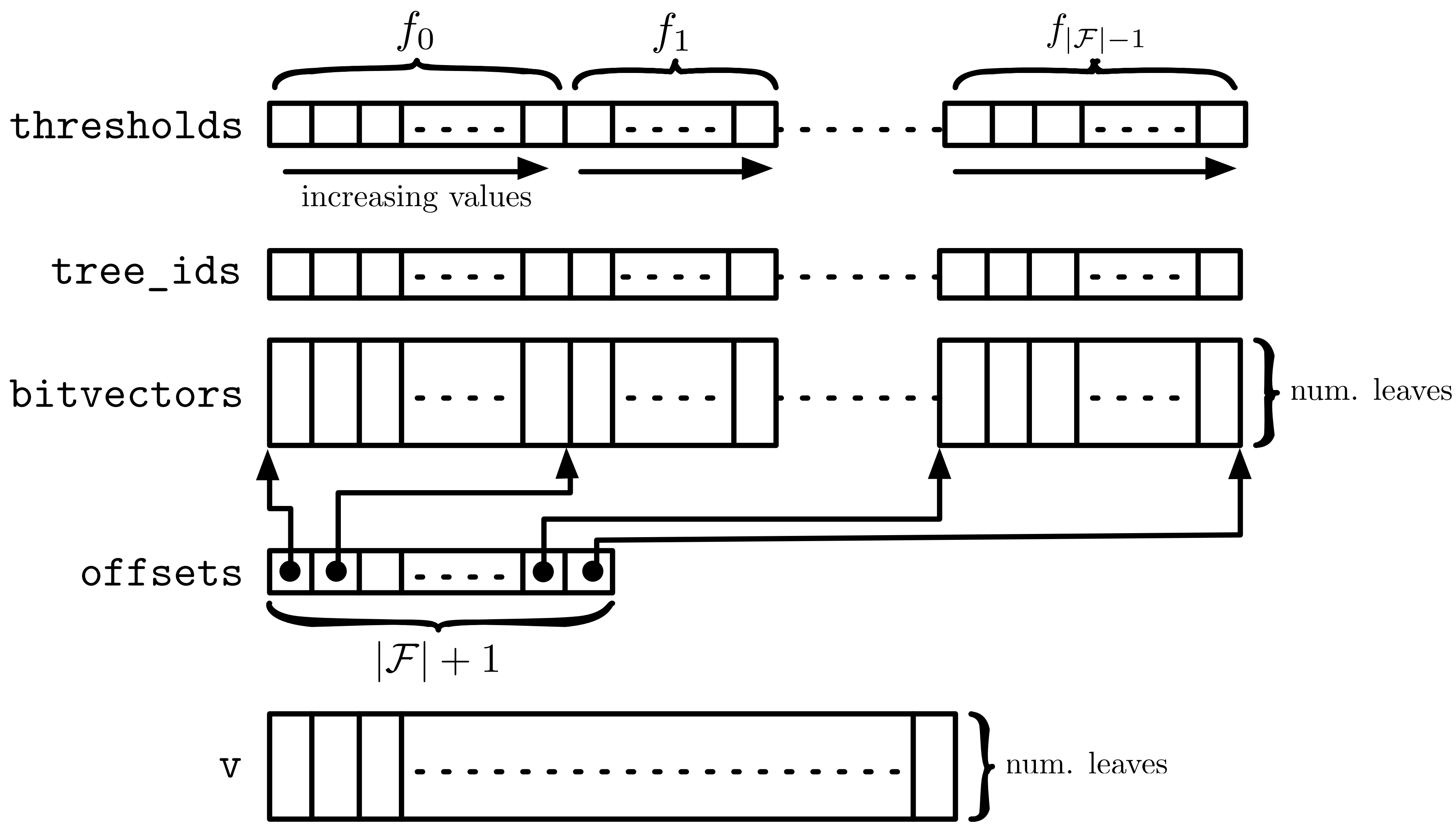
- # docs = >100K
- # trees = 1K–20K
- # features = 100–1000
- # leaves = 4–64

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Interleaved execution of several tree traversals

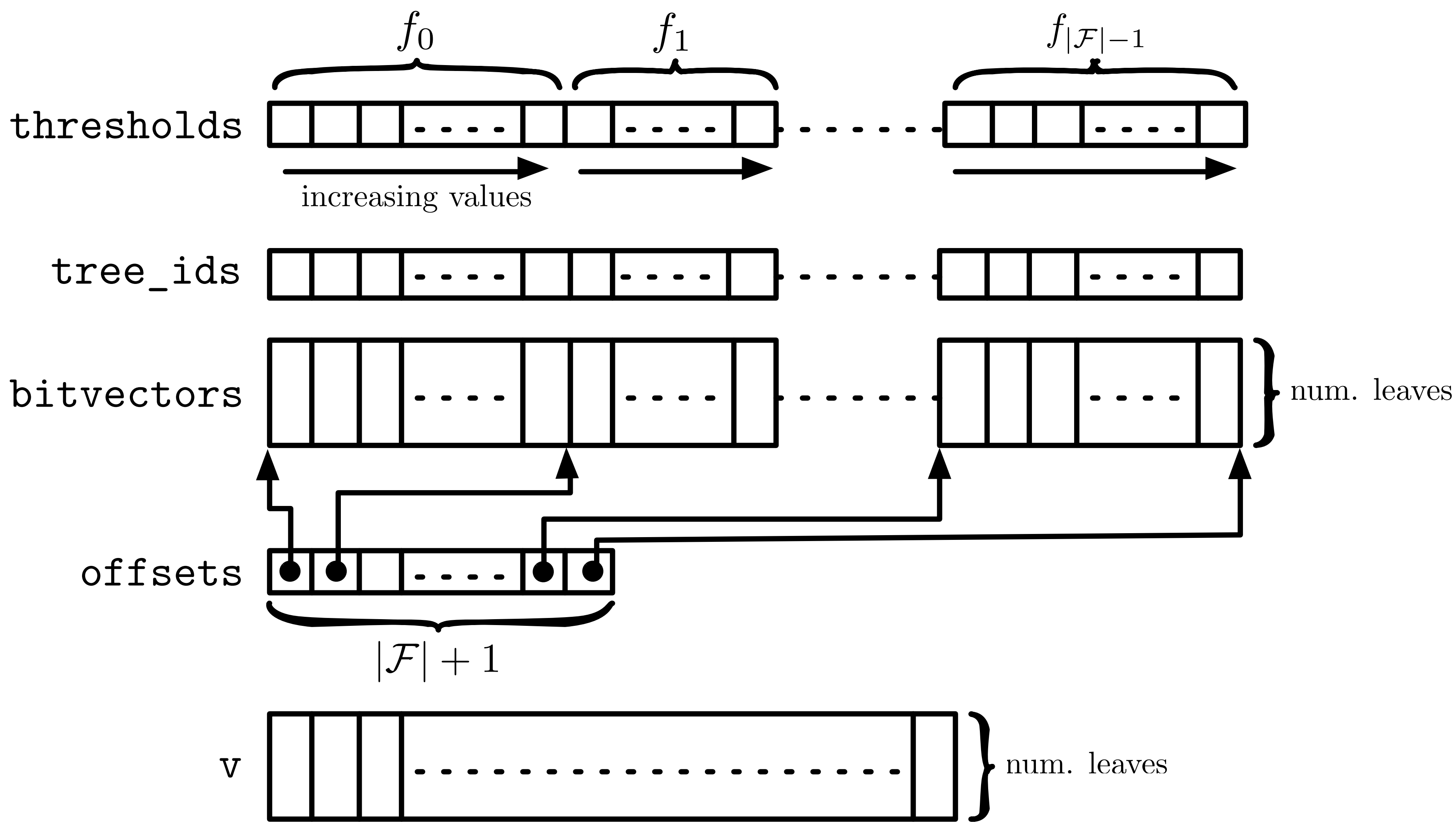


Interleaved execution of several tree traversals



ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Interleaved execution of several tree traversals

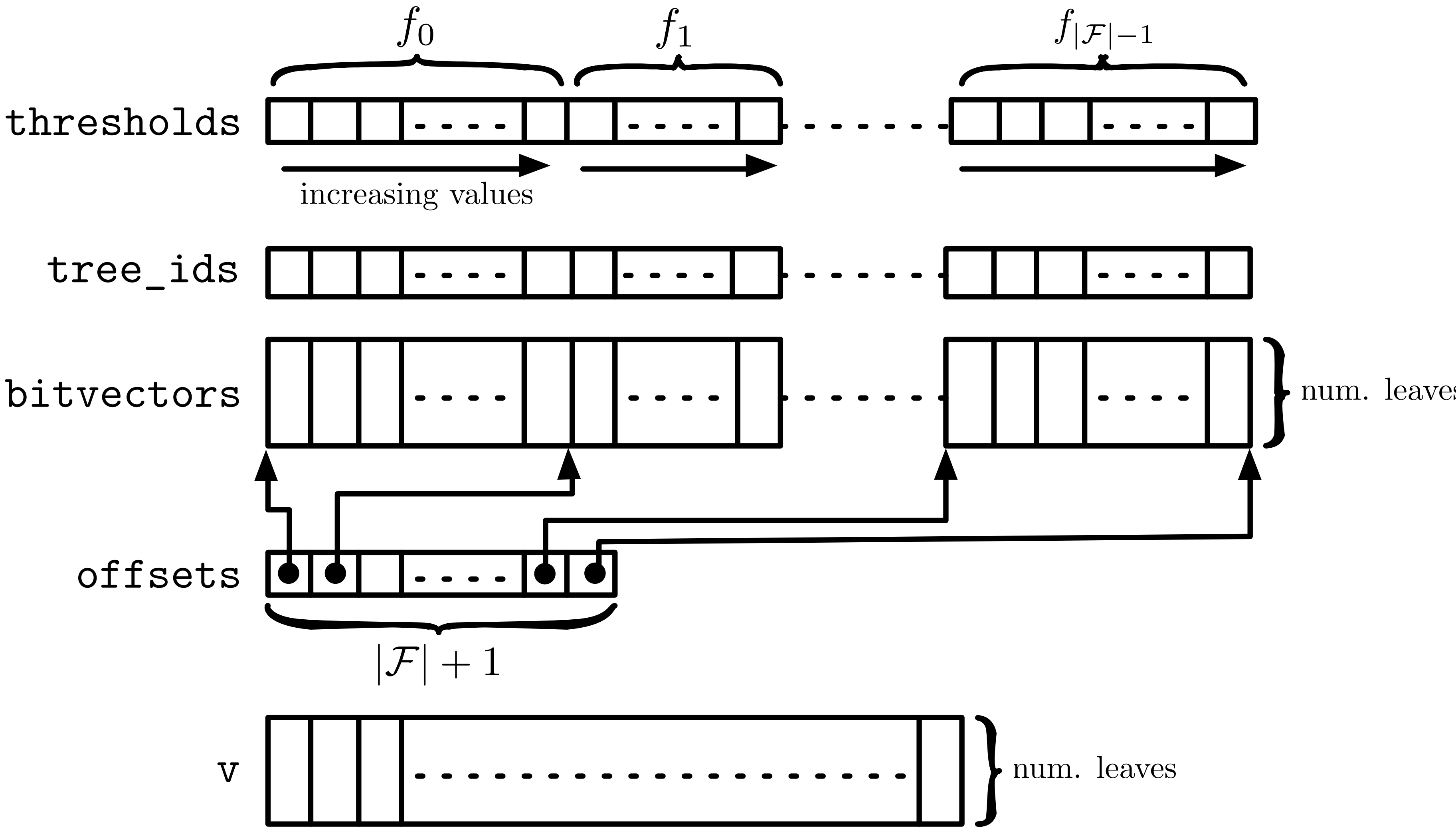


Documents

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

...

Interleaved execution of several tree traversals

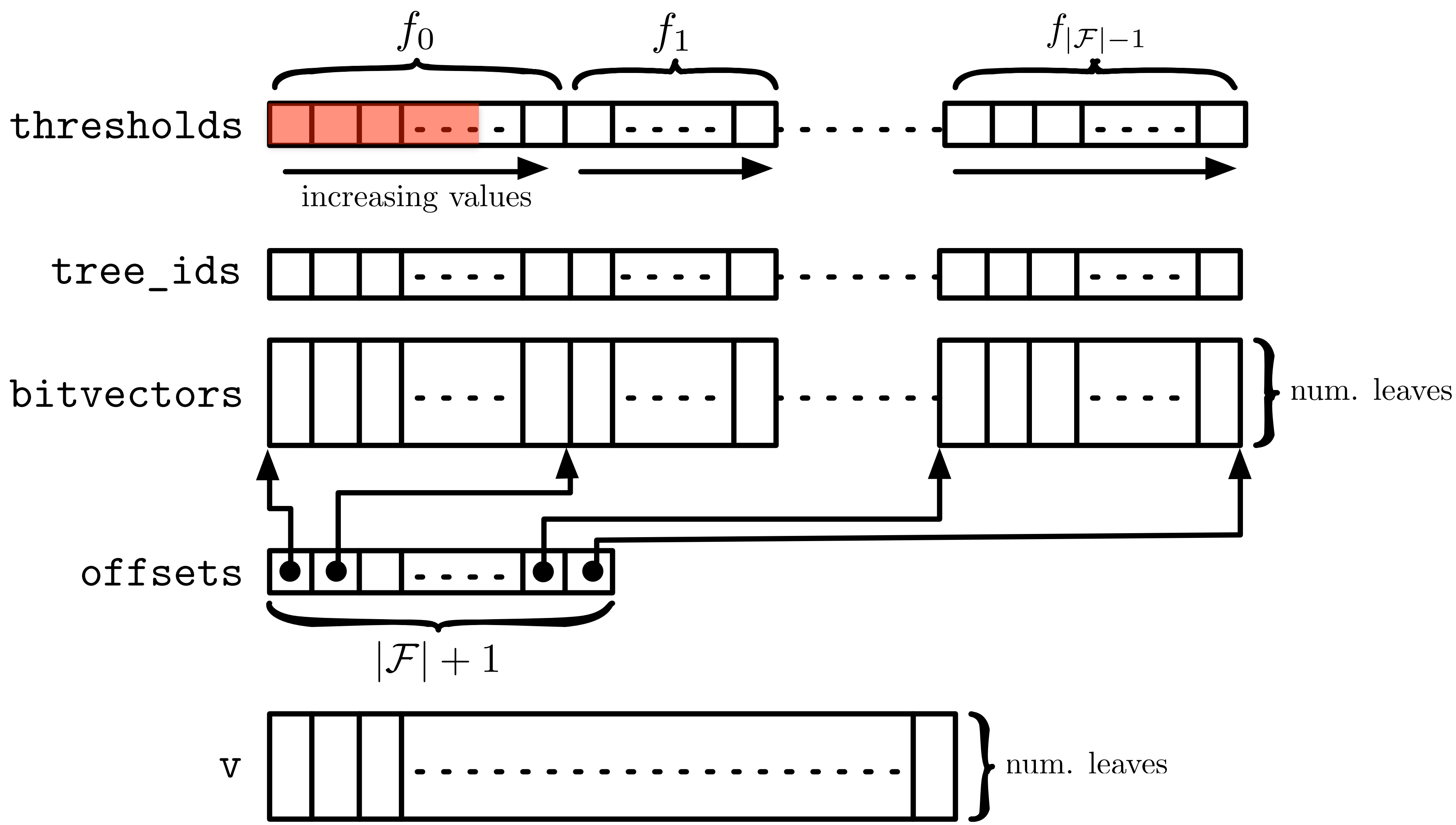


Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

...

Interleaved execution of several tree traversals

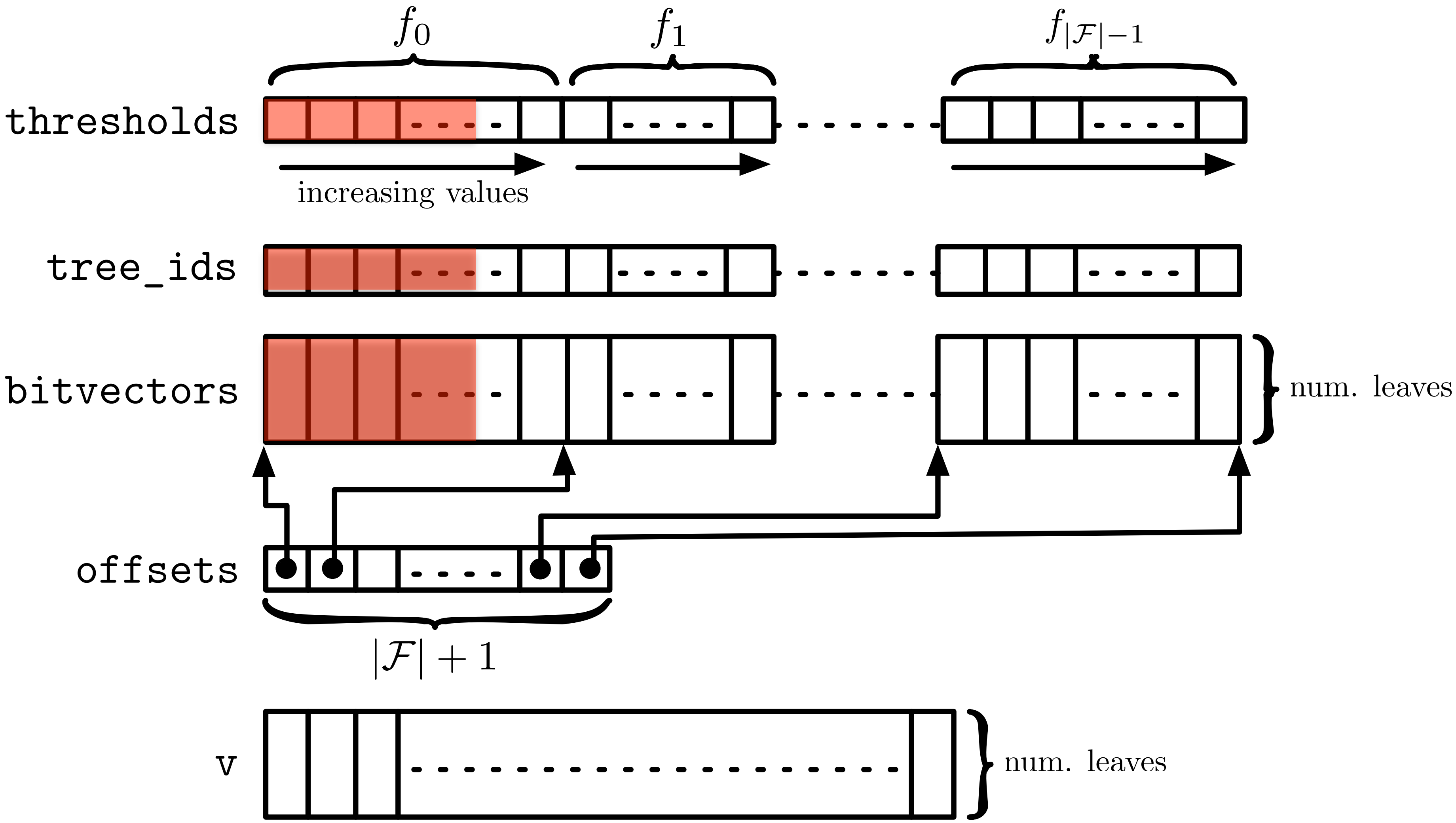


Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

...

Interleaved execution of several tree traversals



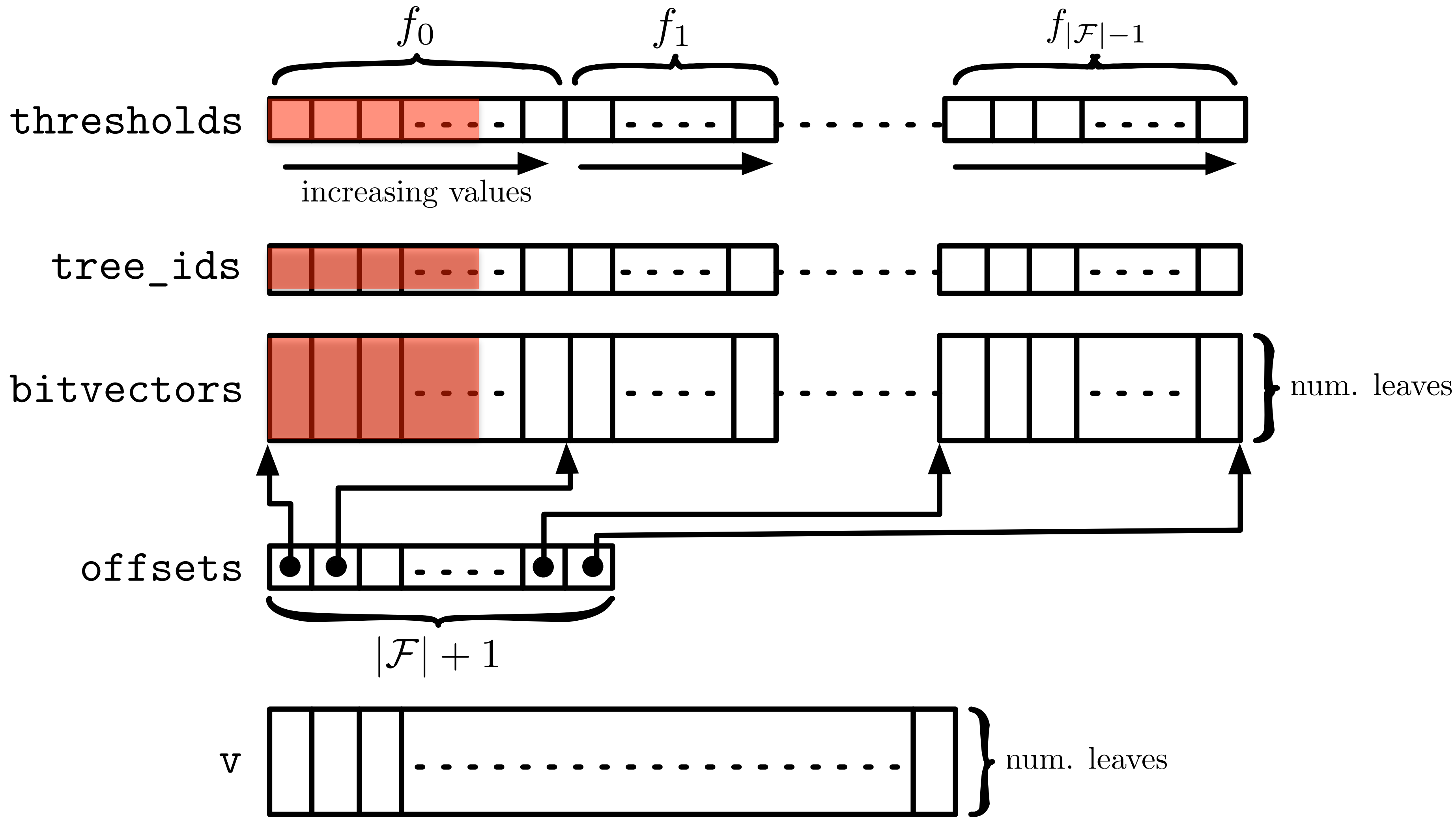
Documents

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

...

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Interleaved execution of several tree traversals

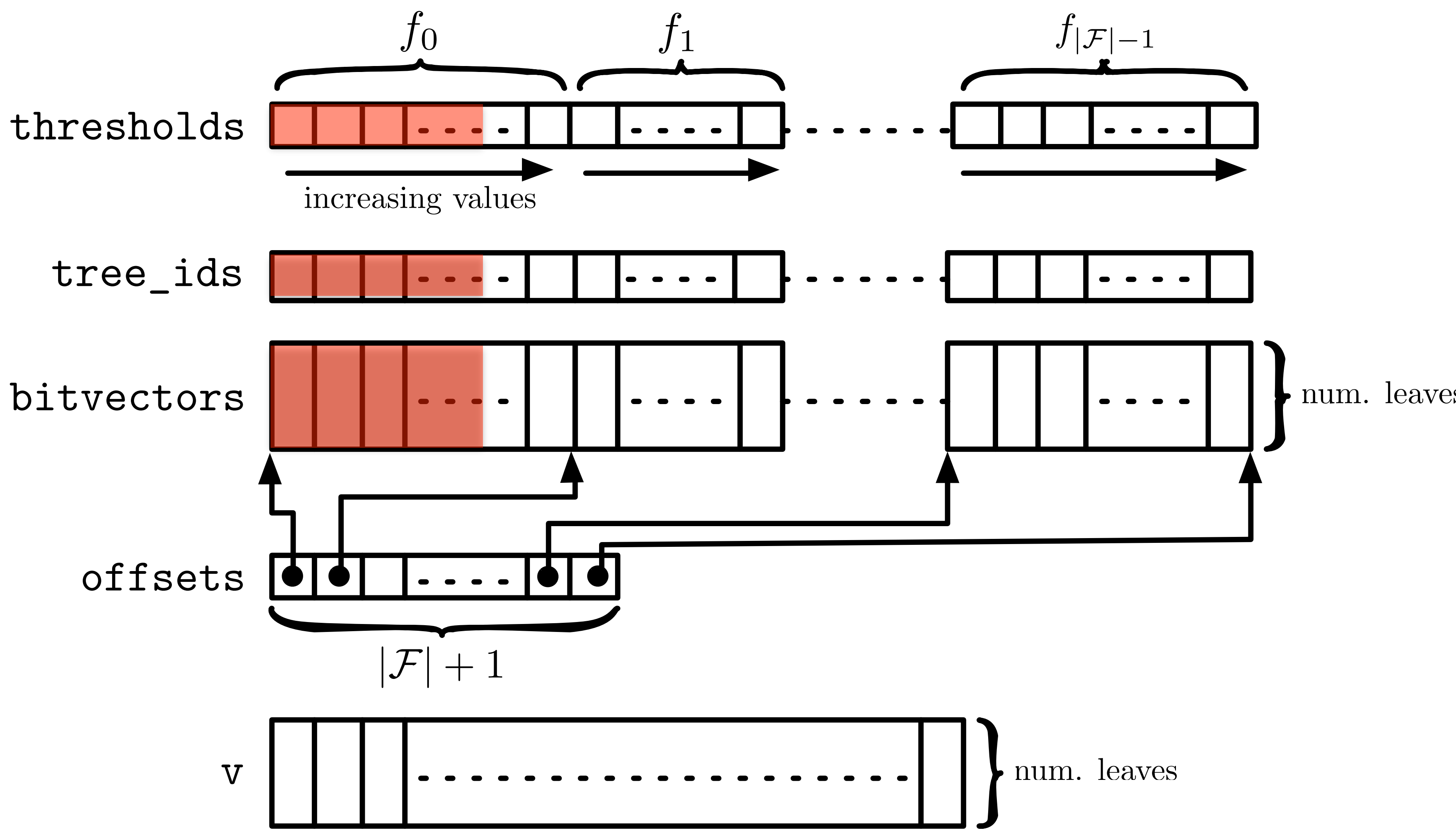


Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

...

Interleaved execution of several tree traversals

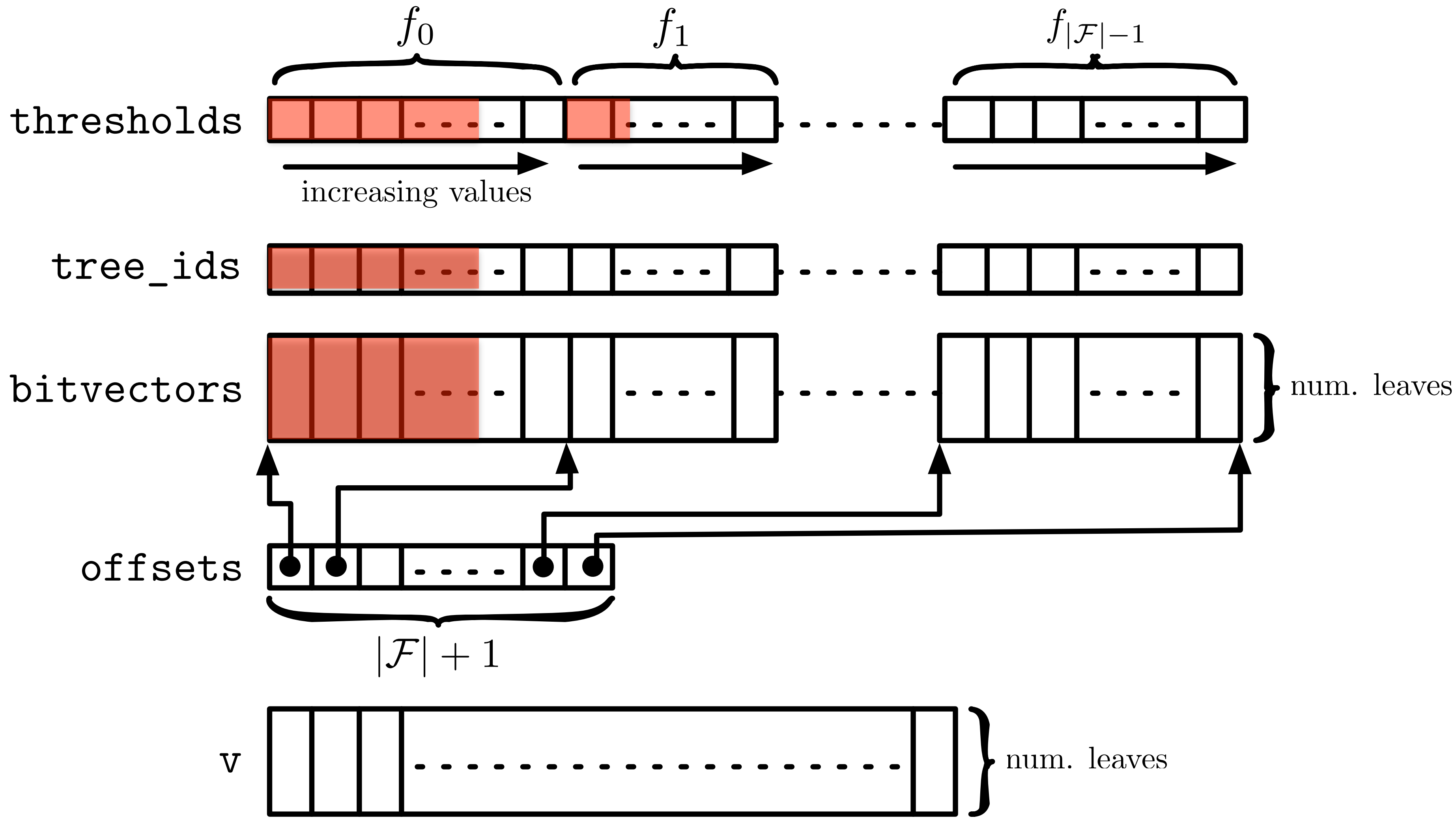


Documents

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

...

Interleaved execution of several tree traversals

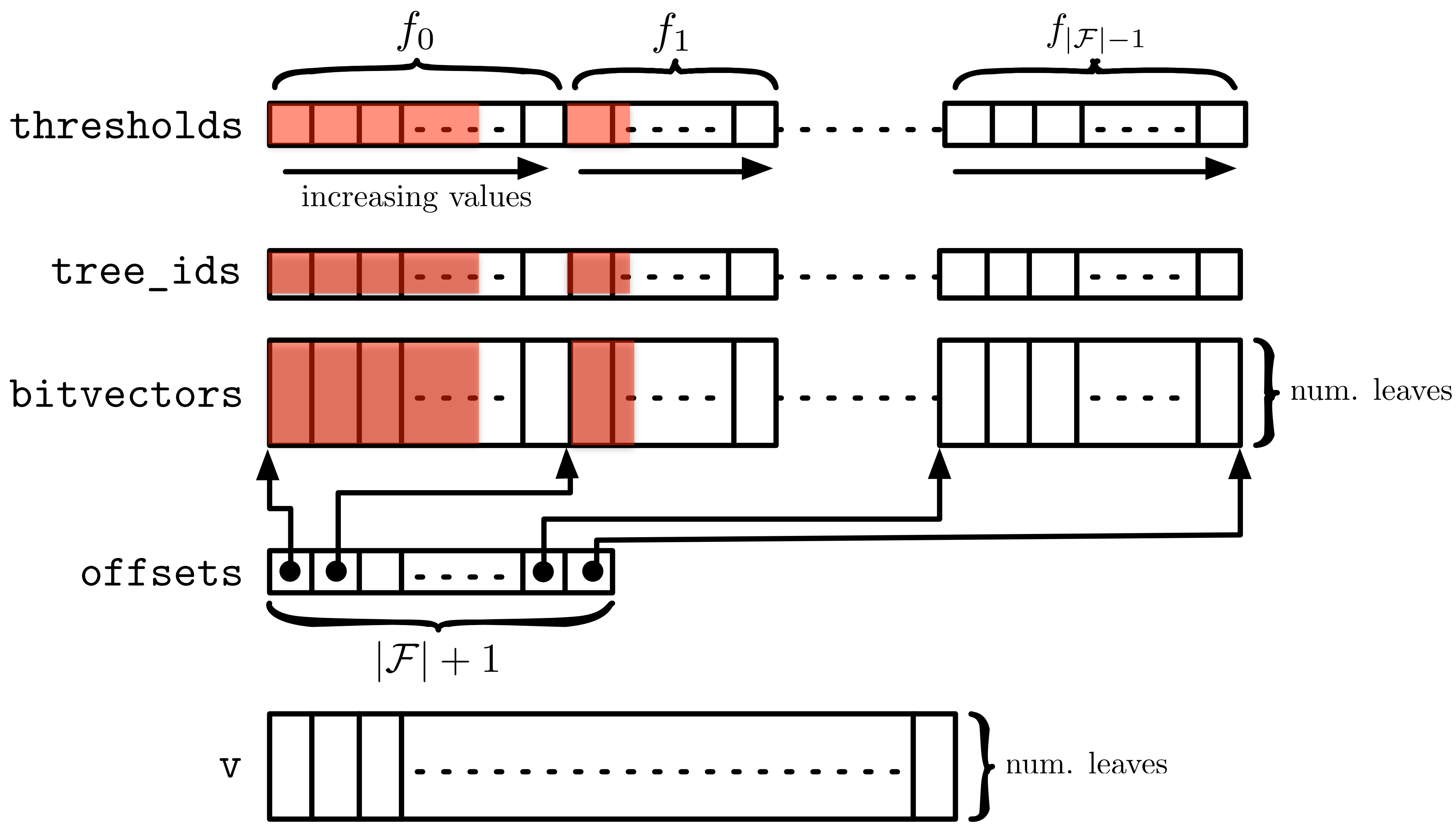


Documents

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

...

Interleaved execution of several tree traversals

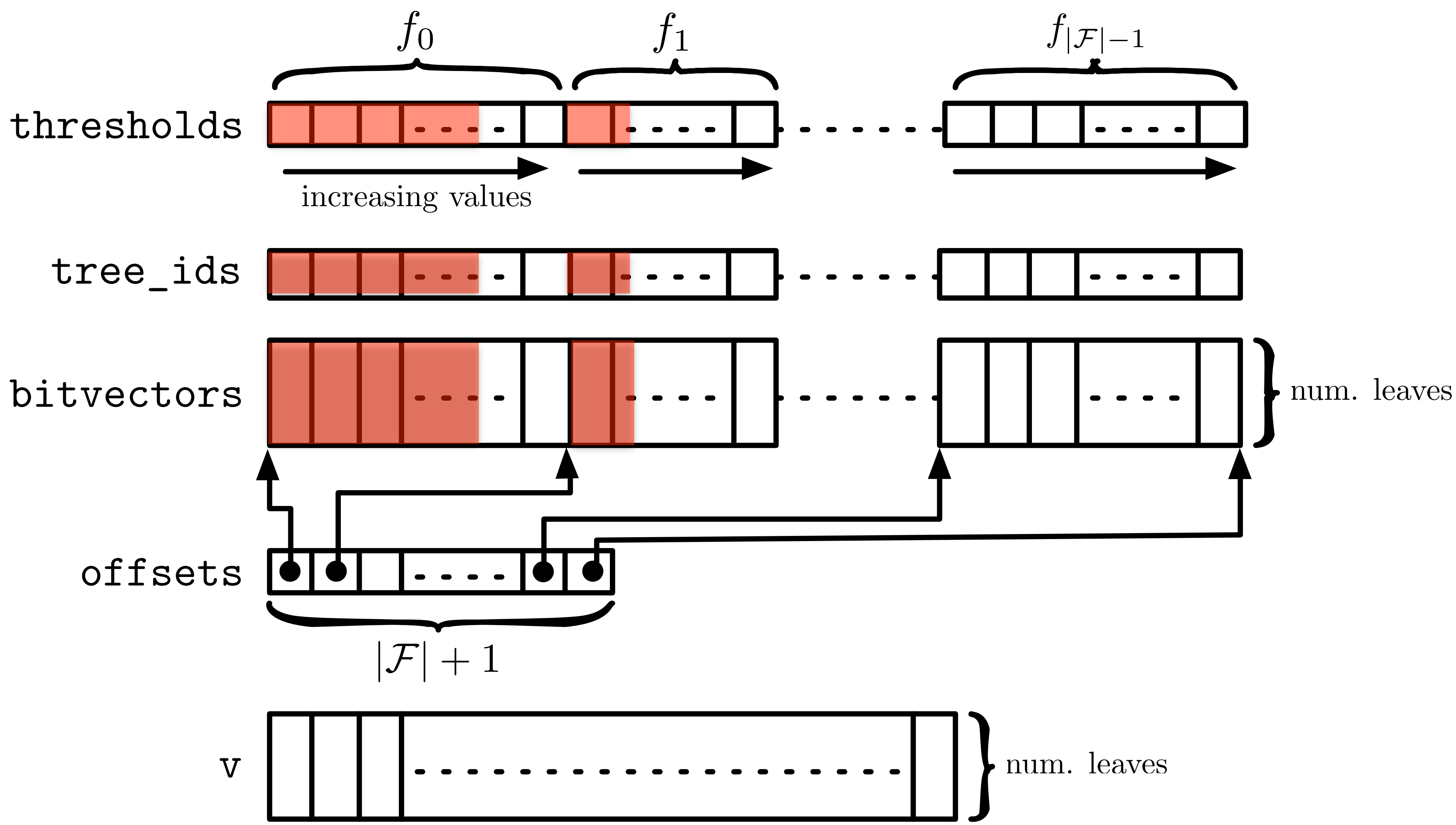


Documents

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

...

Interleaved execution of several tree traversals



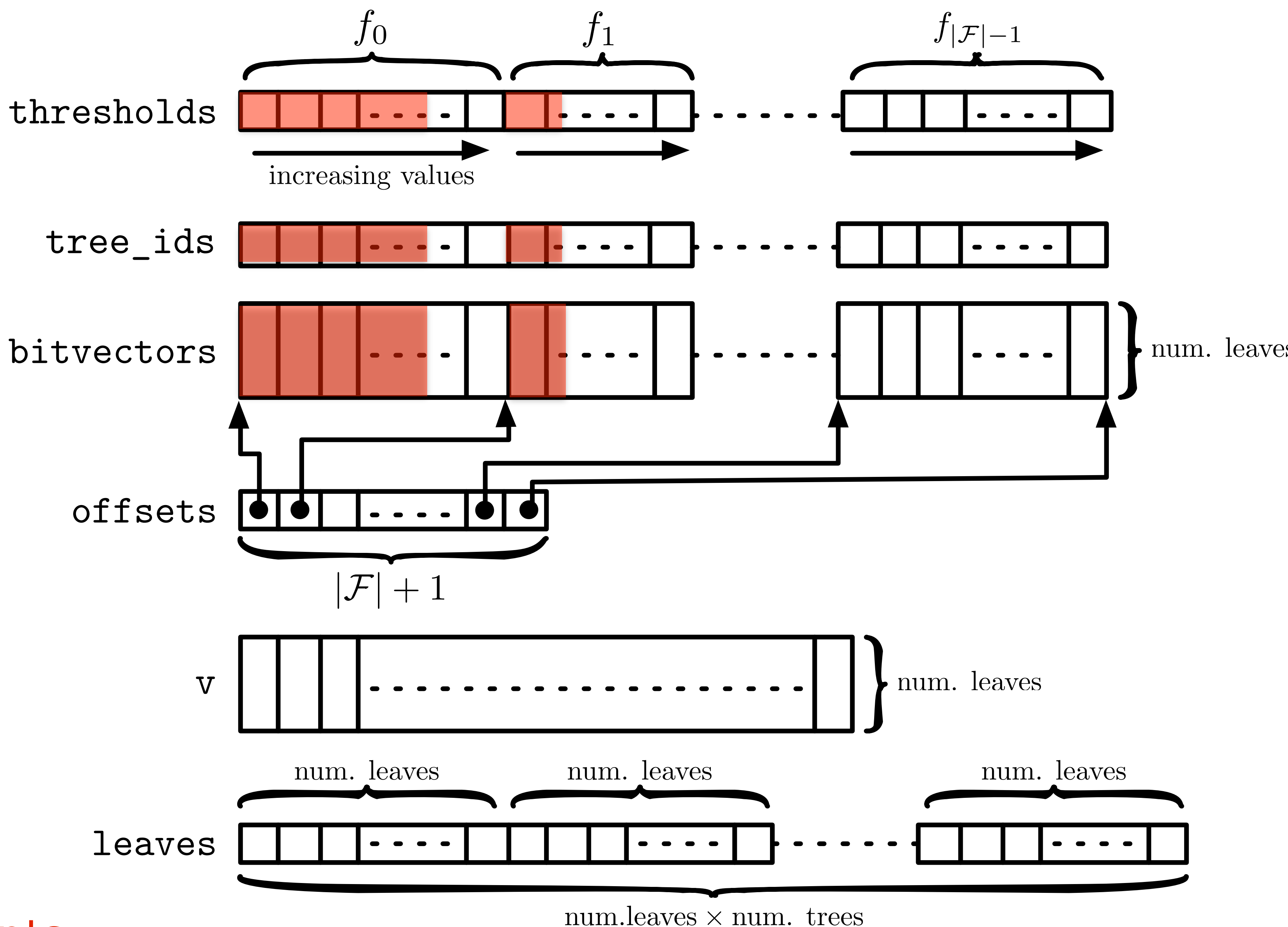
Documents

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

...

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Interleaved execution of several tree traversals

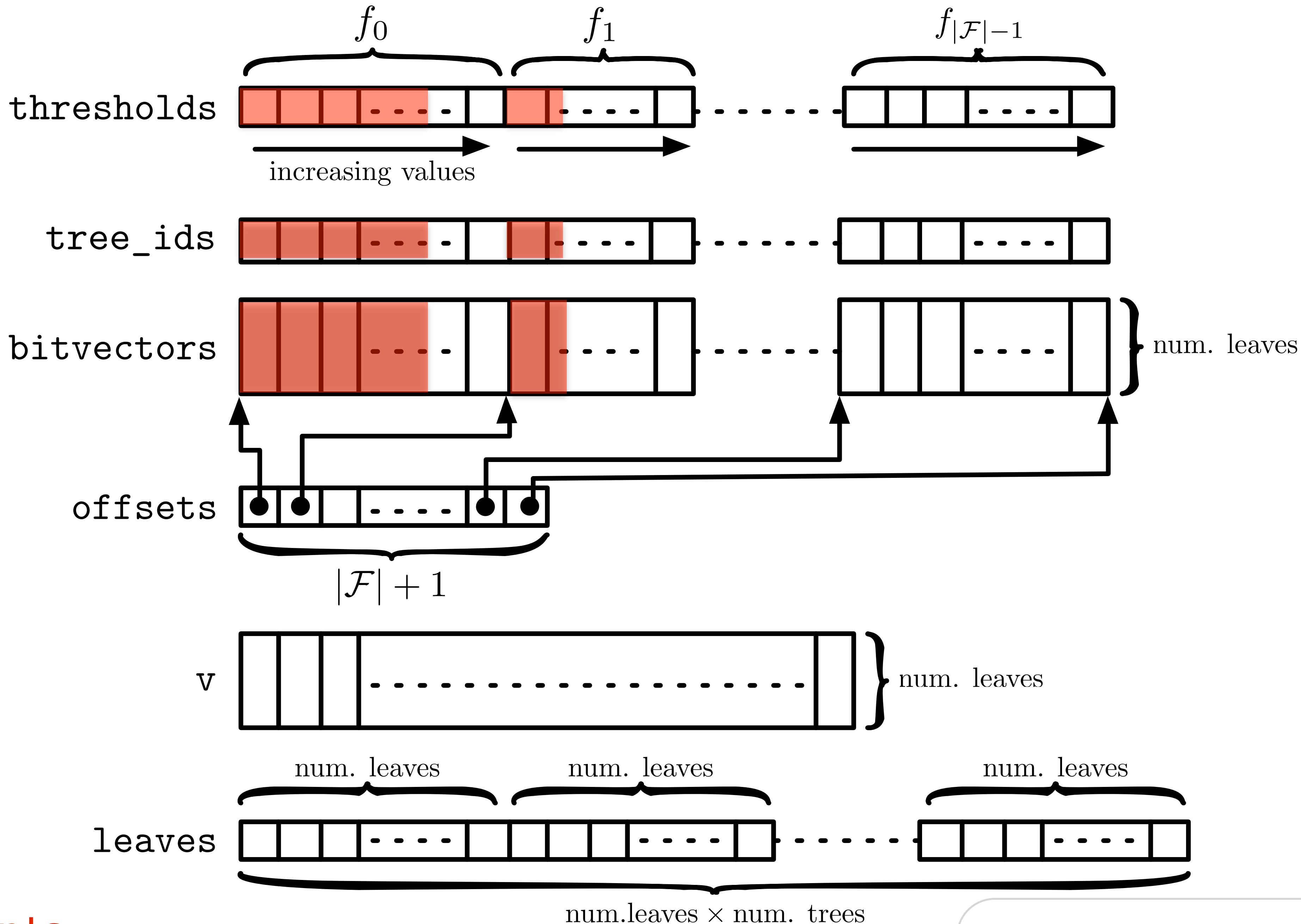


Documents

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

...

Interleaved execution of several tree traversals



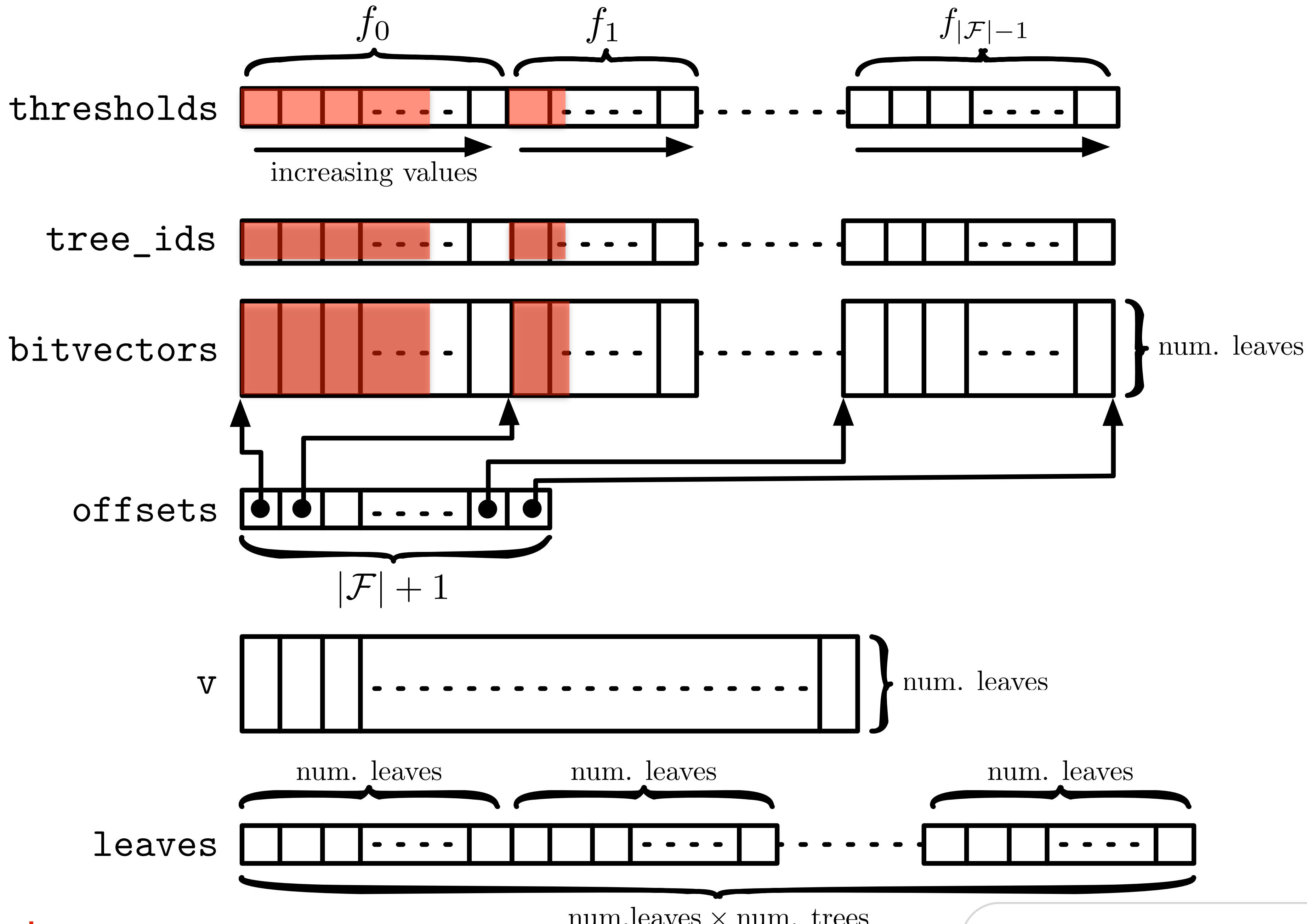
Documents

F1	F2	F3	F4	F5	F6	F7	F8
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

Interleaved execution of several tree traversals



Documents

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

...

Low branch misprediction rate

High cache hit ratio

Results

MSN-1

\# docs = X
\# trees = 1K-5K-10K-20K
\# features = 136
\# leaves = 8-16-32-64

Y!S1

\# docs = Y
\# trees = 1K-5K-10K-20K
\# features = 700
\# leaves = 8-16-32-64

λ-MART for performing the training phase optimizing NDCG@10

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-document scoring time in microsecs

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-document scoring time in microsecs

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-document scoring time in microsecs

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-document scoring time in microsecs

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-document scoring time in microsecs

Questions & Comments

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"