INFORMATION RETRIEVAL

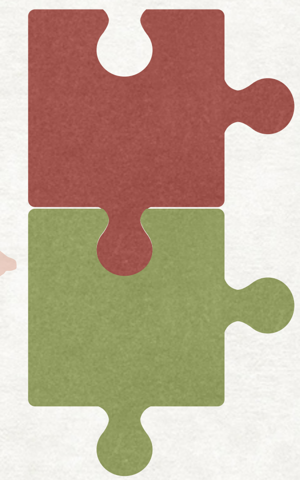
Luca Manzoni
lmanzoni@units.it

Lecture 7

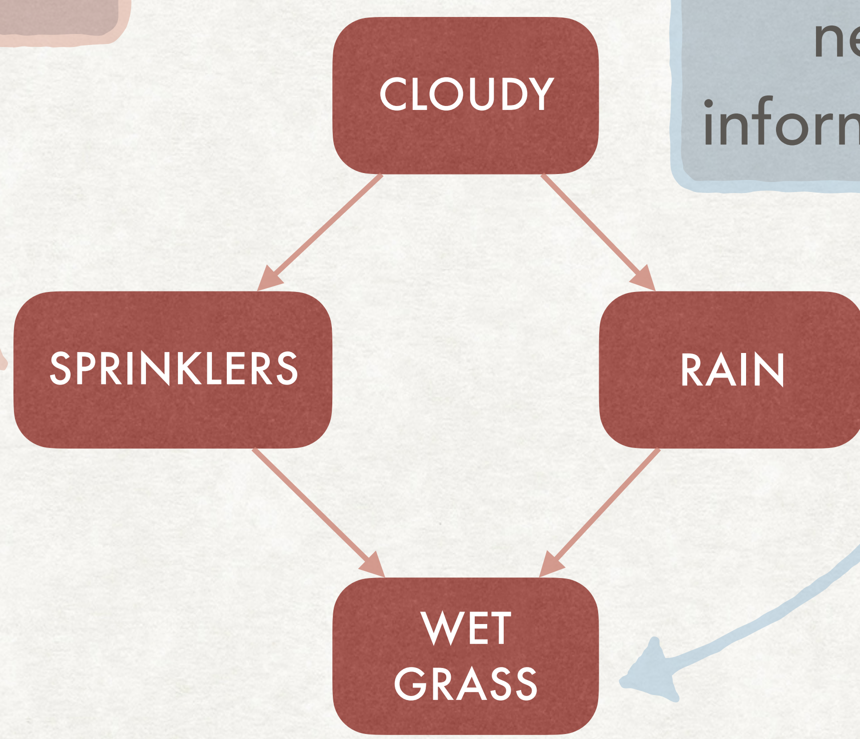
LECTURE OUTLINE

*SIDE EFFECTS MAY INCLUDE SIDE EFFECTS

Putting Everything Together

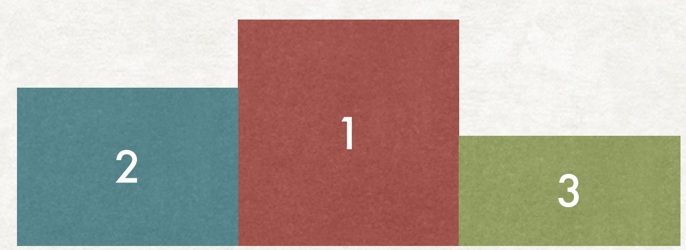


Bayesian Networks



Using Bayesian networks for information retrieval

Evaluation of IR systems



BAYESIAN NETWORKS

BAYESIAN NETWORKS

WHAT ARE THEM

- Also called **Bayesian belief networks, decision network**, etc.

- A **graphical model** is a statistical model using a graph to represent the conditional dependency between random variables.

- BN are a kind **graphical model** using a directed acyclic graph.

- Intuitively they are useful because when we need to compute $P(y|x_1, x_2, \ldots, x_k)$ we actually need to compute only $p(y|\mathrm{Pa}(y))$ with $\mathrm{Pa}(y)$ the parent nodes of y.

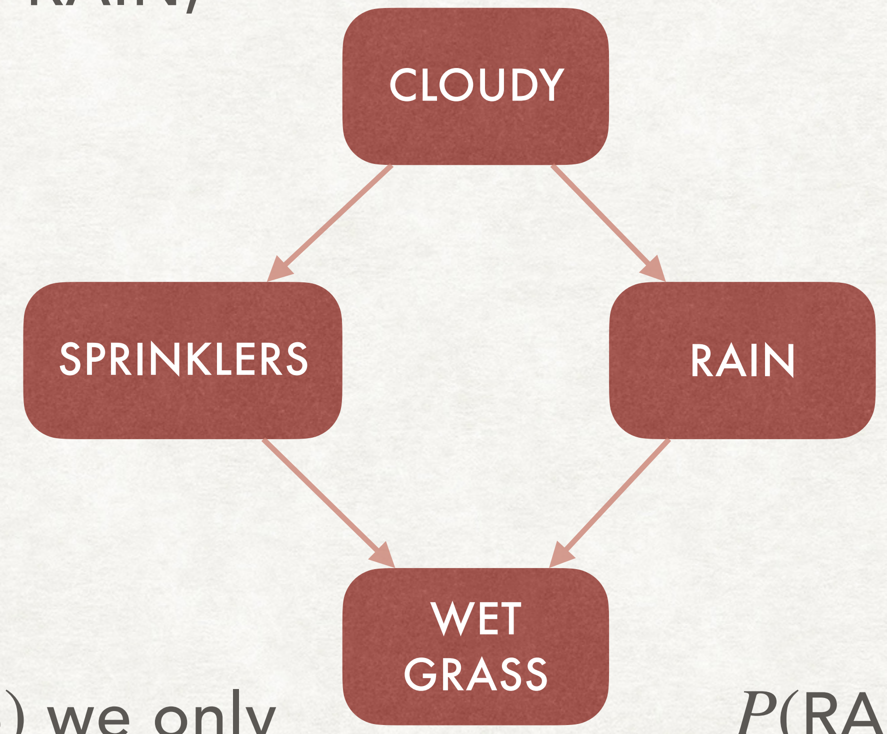
- An example should clarify this.

BAYESIAN NETWORKS

A SIMPLE EXAMPLE

There are four random variables: CLOUDY, SPRINKLERS, RAIN, and WET GRASS.

The edges represents the conditional dependencies



If we want to compute P(CLOUDY | SPRINKLES) we only compute P(SPRINKLES | CLOUDY), and we will have to "rewrite" it.

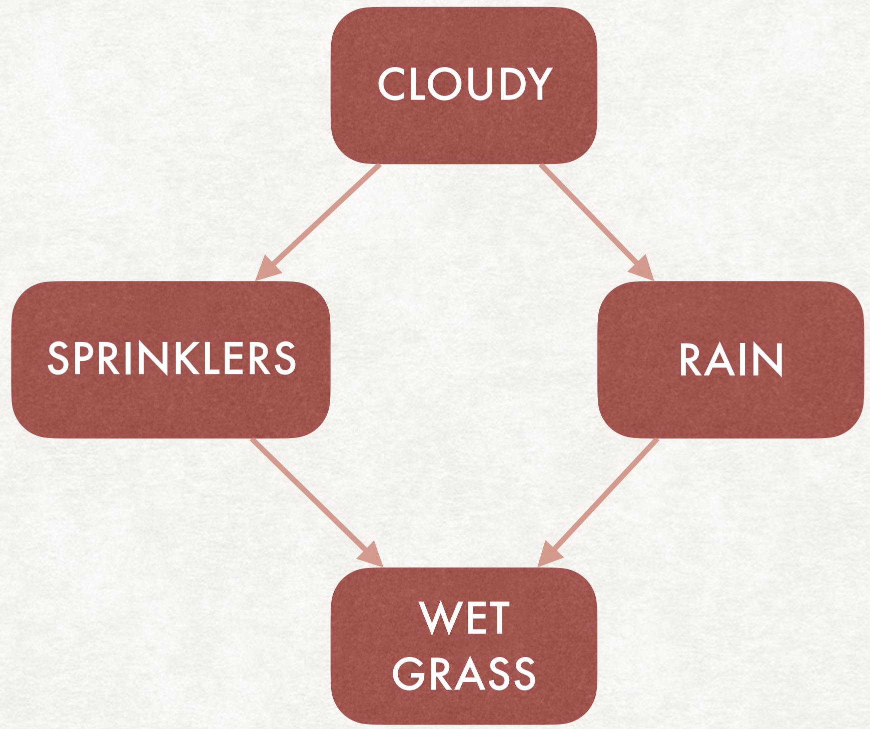
If we want to compute P(RAIN | CLOUDY) we can find it directly in our table

BAYESIAN NETWORKS

A SIMPLE EXAMPLE

C = 0	C =1
0,5	0,5

	S = 0	S =1
C = 0	0,5	0,5
C = 1	0,9	0,1



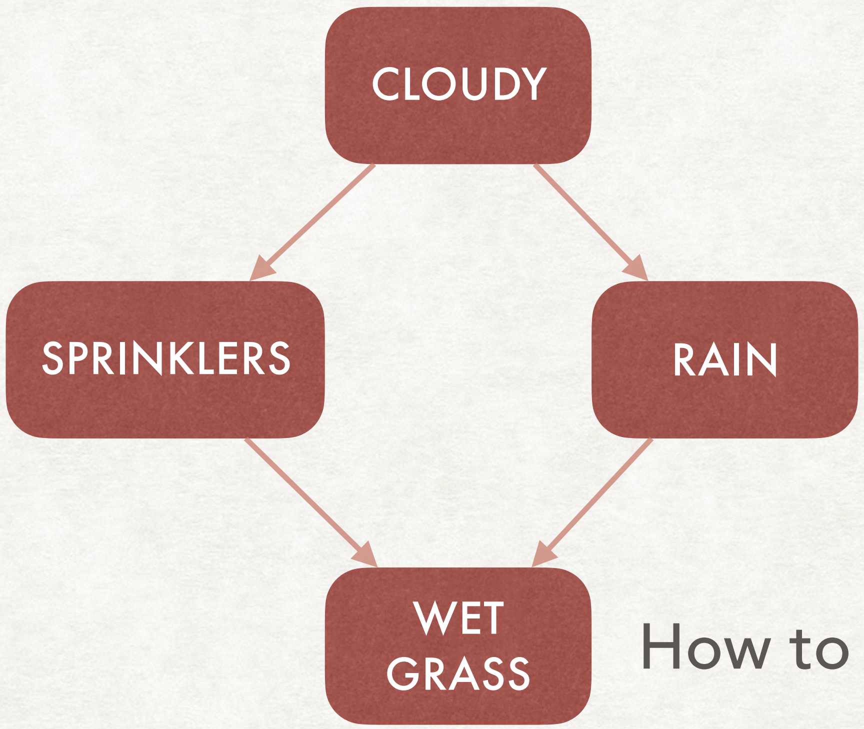
	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

		W = 0	W =1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99

BAYESIAN NETWORKS

A SIMPLE EXAMPLE

C = 0	C =1
0,5	0,5



	S = 0	S =1
C = 0	0,5	0,5
C = 1	0,9	0,1

	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

How to find $P(\mathrm{W} = 1 \,|\, \mathrm{S} = 1, \mathrm{R} = 0)$?

		W = 0	W =1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99

BAYESIAN NETWORKS

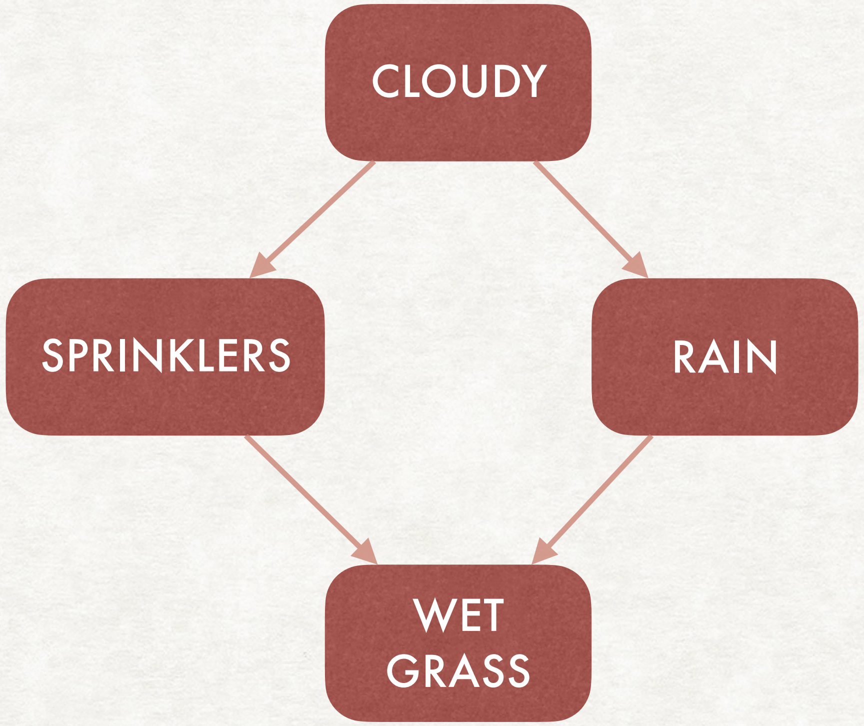
A SIMPLE EXAMPLE

C = 0	C =1
0,5	0,5

	S = 0	S =1
C = 0	0,5	0,5
C = 1	0,9	0,1

	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

		W = 0	W =1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99



How to find

$P(\mathrm{W}=1\,|\,\mathrm{C}=1,\mathrm{R}=0)$?

$$P(\mathrm{W}=1\,|\,\mathrm{C}=1,\mathrm{R}=0)$$

$$= P(\mathrm{W}=1\,|\,\mathrm{R}=0,\mathrm{S}=1)\cdot P(\mathrm{S}=1\,|\,\mathrm{C}=1)$$
$$+P(\mathrm{W}=1\,|\,\mathrm{R}=0,\mathrm{S}=0)\cdot P(\mathrm{S}=0\,|\,\mathrm{C}=1)$$

$$= 0.9\cdot 0.1 + 0\cdot 0.9$$

$$= 0.09$$

BAYESIAN NETWORKS

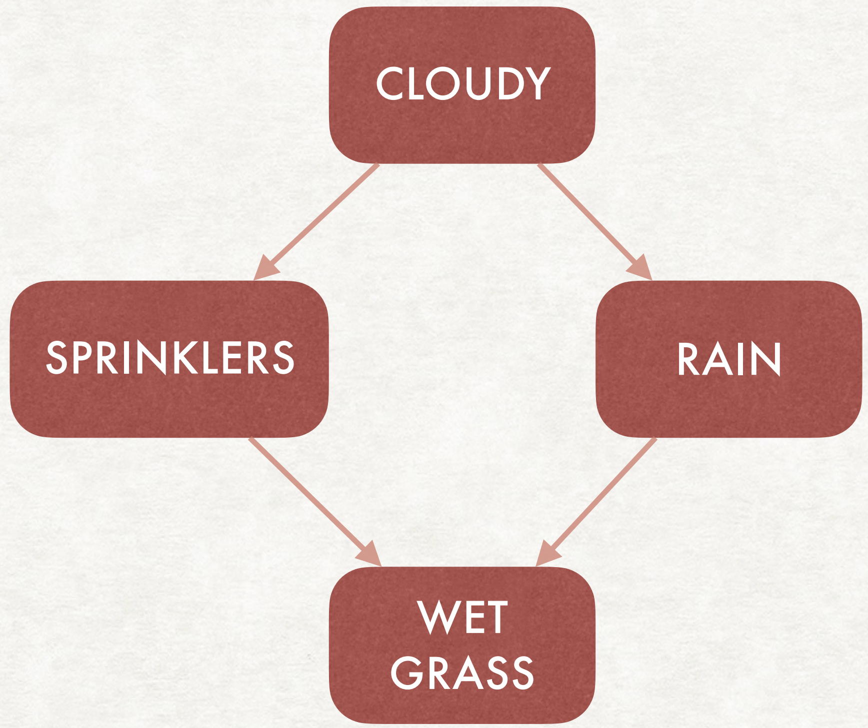
A SIMPLE EXAMPLE

C = 0	C = 1
0,5	0,5

	S = 0	S = 1
C = 0	0,5	0,5
C = 1	0,9	0,1

	R = 0	R = 1
C = 0	0,8	0,2
C = 1	0,2	0,8

		W = 0	W = 1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99



How to find

$P(S = 1 \mid C = 1, W = 1)$?

$$P(S = 1 \mid C = 1, W = 1)$$

$$= \frac{P(W = 1 \mid C = 1, S = 1)}{P(W = 1 \mid C = 1)} \cdot P(S = 1 \mid C = 1)$$

$$= \frac{P(W = 1 \mid C = 1, S = 1)}{P(W = 1 \mid C = 1)} \cdot 0.1$$

$P(W = 1 \mid C = 1, S = 1)$ $P(W = 1 \mid C = 1)$

BAYESIAN NETWORKS

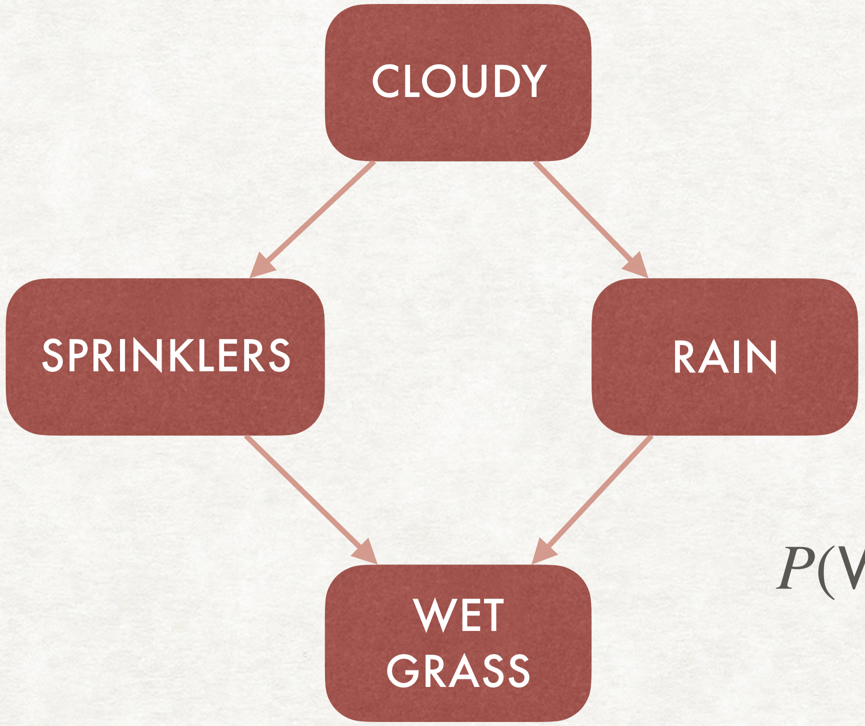
A SIMPLE EXAMPLE

C = 0	C =1
0,5	0,5

	S = 0	S =1
C = 0	0,5	0,5
C = 1	0,9	0,1

	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

		W = 0	W =1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99



$$P(\mathrm{W} = 1 \mid \mathrm{C} = 1, \mathrm{S} = 1) = 0.0972$$

$$P(\mathrm{W} = 1 \mid \mathrm{C} = 1)$$

$$P(\mathrm{W} = 1 \mid \mathrm{S} = 0, \mathrm{R} = 0) \cdot P(\mathrm{S} = 0 \mid \mathrm{C} = 1) \cdot P(\mathrm{R} = 0 \mid \mathrm{C} = 1) +$$
$$P(\mathrm{W} = 1 \mid \mathrm{S} = 0, \mathrm{R} = 1) \cdot P(\mathrm{S} = 0 \mid \mathrm{C} = 1) \cdot P(\mathrm{R} = 1 \mid \mathrm{C} = 1) +$$
$$P(\mathrm{W} = 1 \mid \mathrm{S} = 1, \mathrm{R} = 0) \cdot P(\mathrm{S} = 1 \mid \mathrm{C} = 1) \cdot P(\mathrm{R} = 0 \mid \mathrm{C} = 1) +$$
$$P(\mathrm{W} = 1 \mid \mathrm{S} = 1, \mathrm{R} = 1) \cdot P(\mathrm{S} = 1 \mid \mathrm{C} = 1) \cdot P(\mathrm{R} = 1 \mid \mathrm{C} = 1)$$

$$0 \cdot 0.9 \cdot 0.2 + 0.9 \cdot 0.9 \cdot 0.8 + 0.9 \cdot 0.1 \cdot 0.2 + 0.99 \cdot 0.1 \cdot 0.8 = 0.7452$$

BAYESIAN NETWORKS

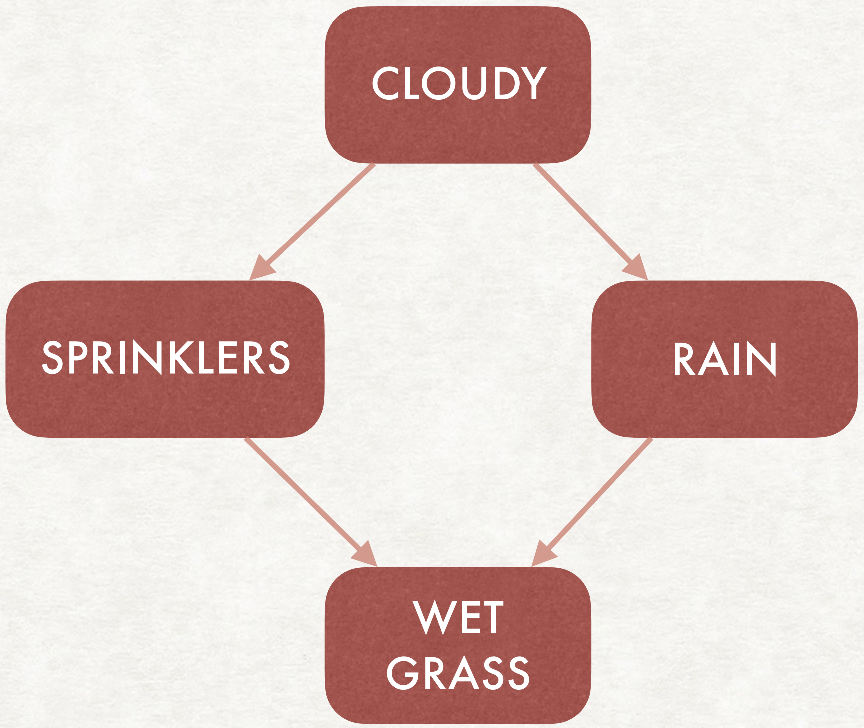
A SIMPLE EXAMPLE

C = 0	C =1
0,5	0,5

	S = 0	S =1
C = 0	0,5	0,5
C = 1	0,9	0,1

	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

		W = 0	W =1
S = 0	R = 0	1	0
S = 1	R = 0	0,1	0,9
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99



How to find

$P(S = 1 | C = 1, W = 1)$?

$$P(S = 1 | C = 1, W = 1)$$

$$= \frac{P(W = 1 | C = 1, S = 1)}{P(W = 1 | C = 1)} \cdot P(S = 1 | C = 1)$$

$$= \frac{P(W = 1 | C = 1, S = 1)}{P(W = 1 | C = 1)} \cdot 0.1$$

$$= \frac{0.0972}{0.7452} \cdot 0.1 \approx 0.013$$

BAYESIAN NETWORKS

INFERENCE

- To find the probability of an event we can use the tables of conditional probabilities of the network.
- We can have more than binary variables by making larger tables.
- The size of the table depends on the number of edges entering the node. For binary variables it is 2^k with k the in-degree of the node.
- Inference in Bayesian networks is, in the general case, intractable from a computational point of view…
- …but for specific cases it can still be performed efficiently.

USE OF BN FOR INFORMATION RETRIEVAL

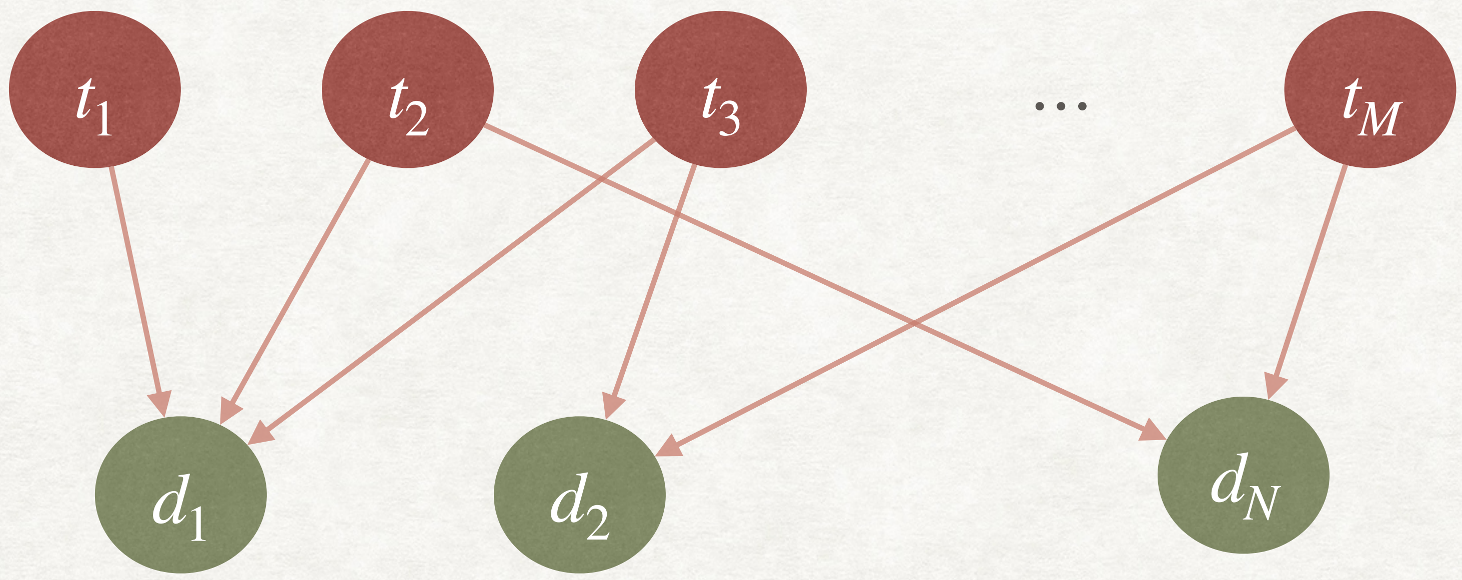
BAYESIAN NETWORKS IN IR

MAIN IDEAS

- Bayesian Networks can model dependencies between terms or documents (contrarily to the assumption of the BIM).
- However, we must always keep an eye to complexity!
- Here we see only one possible model. Other model with different topologies exist.

BN STRUCTURE

A SIMPLE STRUCTURE



Nodes for the terms

Nodes for the documents

Each edge connect a term with a document containing the term.

Both the t_i and d_j are binary random variables with meanings:

- t_i means "the term t_i is relevant"
- d_j means "the document d_j is relevant"

SETTING THE PROBABILITIES

FOR TERMS AND DOCUMENTS

t_i

t_i	not t_i
1/M	1-1/M

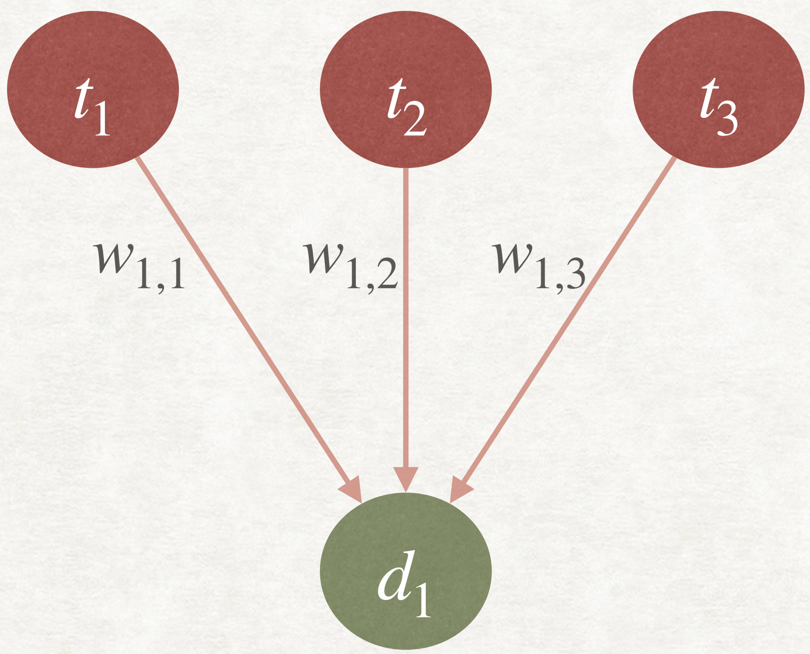
d_j

The size of the table depends *exponentially* by the number of terms in the document: with 50 terms we need a table of 2^{50} entries.

A different approach is needed to store the conditional probabilities

SETTING THE PROBABILITIES

FOR TERMS AND DOCUMENTS



We assign weights to each edge

The value $P(d_j \,|\, \mathrm{Pa}(d_j))$ is now computed as:

$$P(d_j \,|\, \mathrm{Pa}(d_j)) = \sum_{i: t_i \in \mathrm{Pa}(d_j),\ t_i = 1} w_{i.j}$$

i.e., sum all $w_{i,j}$ for all the parent nodes with state 1 (relevant)

SETTING THE WEIGHTS

ONE METHOD OF WEIGHTING

Multiple weighting methods are possible.
Two conditions to be respected are:

- $w_{i,j} \geq 0$ for all i and j.
- $\sum_{t_i \in d_j} w_{i,j} \leq 1$ for all documents d_j.

MADE TO "RESEMBLE" THE COSINE MEASURE

One possible weighting scheme is

$$w_{i,j} = \alpha^{-1} \frac{\text{tf-idf}_{i,j}^2}{\sum t_k \in d_j \left(\text{tf-idf}_{k,j}\right)^2}$$

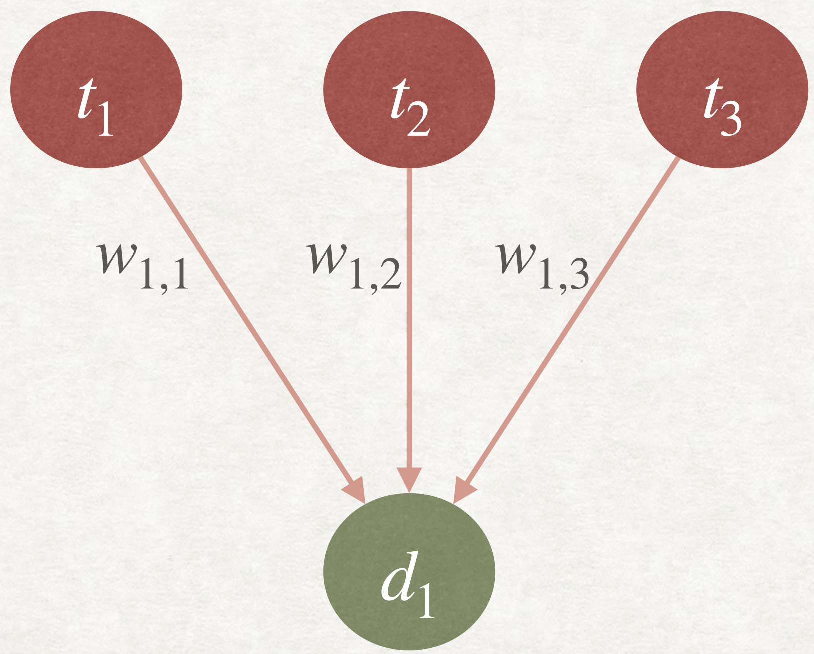
With α a normalising constant

USING A QUERY

HOW THE QUERY SETS THE STATE OF TERMS

Given a query q we assume that all terms in q are relevant (i.e., $t_i = 1$ if $t_i \in q$).
We use the notations $P(t_i \mid q)$ and $P(d_j \mid q)$

Suppose $q = t_1\ t_3$, then $P(d_1 \mid q)$ is:



$$P(d_1 \mid q) = w_{1,1} + w_{1,2} \cdot \frac{1}{M} + w_{1,3}$$

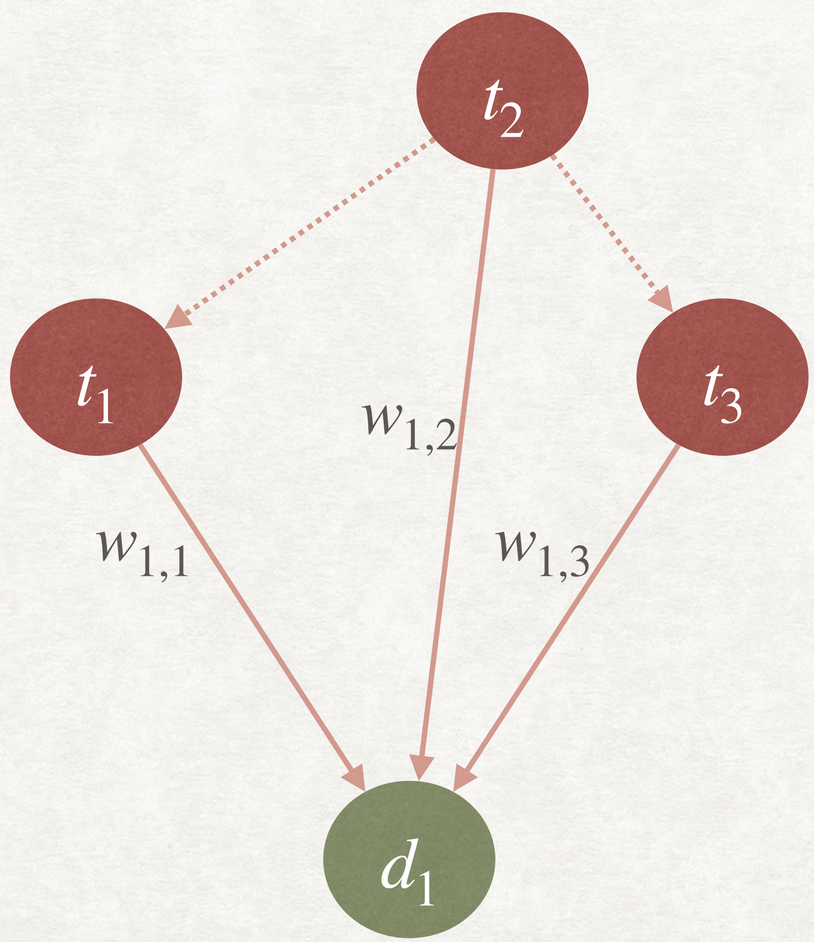
In general:

$$P(d_j \mid q) = \sum_{i:t_i \in \mathrm{Pa}(d_j)} w_{i,j} P(t_i \mid q)$$

ADDING DEPENDENCIES

AT LEAST AMONG TERMS

Until now we have considered the term independent from one another. We can now add some form of dependency between terms while keeping the graph acyclic.



Now we need a way to set the probabilities for root nodes (without any parent) and for nodes with parents.

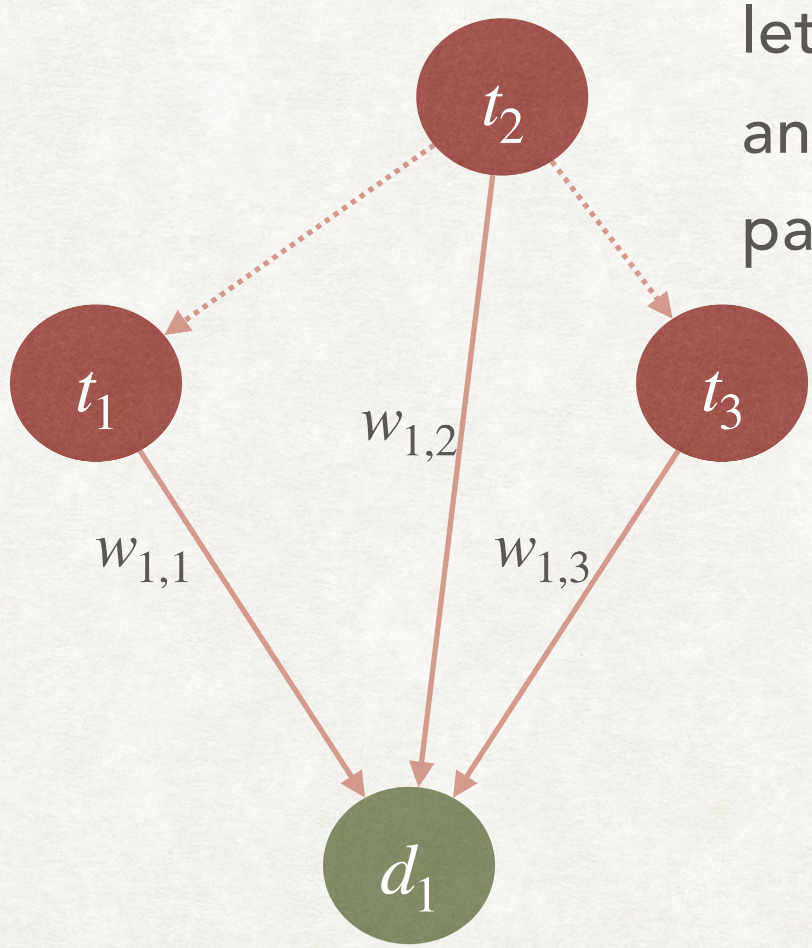
For root nodes we already have:

t_i	not t_i
1/M	1-1/M

ADDING DEPENDENCIES

SETTING THE WEIGHTS

We can use the idea for the Jaccard coefficient of "similarity" among terms



Given a "configuration" x of the parent terms (i.e., which terms are present and which are not) let $A_{\bar{t}_i,x}$ be the set of documents not containing t_i and containing the exact "configuration" x of the parent node. Similarly, define $A_{\bar{t}_i}$ and A_x. Then:

$$P(t_i = 0 \,|\, \mathrm{Pa}(t_i) = x) = \frac{|A_{\bar{t}_i,x}|}{|A_{\bar{t}_i}| + |A_x| - |A_{\bar{t}_i,x}|}$$

$$P(t_i = 1 \,|\, \mathrm{Pa}(t_i) = x) = 1 - P(t_i = 0 \,|\, \mathrm{Pa}(t_i) = x)$$

BAYESIAN NETWORKS

FINAL REMARKS

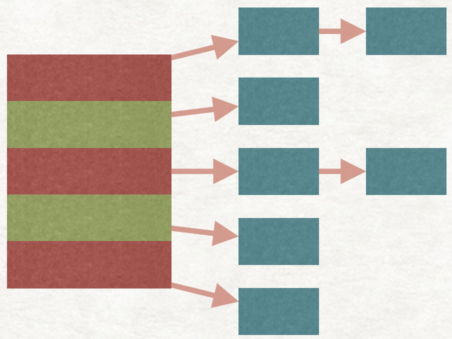
- We have seen only one model of IR using Bayesian networks.
- We can actually also add some dependencies between documents.
- In any case we must find a way to design or learn the dependencies. E.g., by estimating $P(d_i | d_j)$ and linking the "top documents"
- Other models are possible, including ones with completely different topologies, like mapping document to terms and then to "general concepts".

INTEGRATING EVERYTHING

TIERED INDEXES

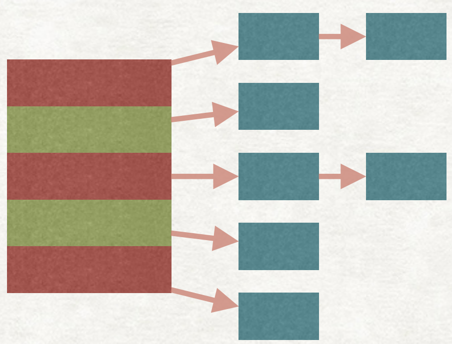
GENERALISATION OF CHAMPION LISTS

Rank 1



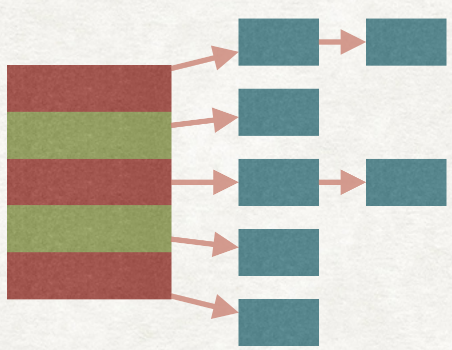
Index for documents with tf over 20

Rank 2



Index for documents with tf between 10 and 20

Rank 3



Index for documents with tf below 10

We search for K documents in the rank 1 index,
if we have less than K we continue in the rank 2 index, and so on

QUERY TERM PROXIMITY

TOWARDS A "SOFT CONJUNCTIVE" SEMANTICS

- If we have a query $q = t_1\ t_2\ \ldots, t_k$ we might want to give a higher score to documents in which the three terms appears close to each other.
- This is not a phrase query, but if the terms appears in close proximity the documents might be an indication that the document is more relevant.
- Let ω the length of the window (in term of number of words) in which $t_1, t_2, \ldots, t_k$ all appear.

QUERY TERM PROXIMITY

TOWARDS A "SOFT CONJUNCTIVE" SEMANTICS

Query: CAT XYLOPHONE

$\omega = 5$

Document 1: THE CAT JUMPED ON THE XYLOPHONE

$\omega =$ a lot more than 5

Document 2: CAT: NOUN, A FELINE [...] XYLOPHONE: NOUN, AN [...]

How can we use ω in out scoring function?

- Hand-coding a scoring function using ω
- As an additional linear term whose weight we can learn from training samples

BOOLEAN RETRIEVAL

HOW TO PERFORM IT IN THE VECTOR SPACE MODEL

- We can use the vector space representation to perform Boolean retrieval:
- A document d is inside the set of documents denoted by t iff $\overrightarrow{v}(d)_t > 0$ (i.e., if the entry t of the vector of d is positive).
- The reverse is not true: the Boolean model does not keep trace of frequencies.
- The two models are different in a more fundamental way: in the Boolean model the queries are written to *select documents*, in the vector space model queries are a form of *evidence accumulation*.

WILDCARD QUERIES

CAN WE IMPLEMENT IT IN THE VECTOR SPACE MODEL?

- In most cases wildcard queries need an additional (and separate) index.
- We can return, from that index, the set of terms that satisfy the wildcards present in the query.
- Suppose that we have CAT* as a query. We obtain the terms "CAT", "CATASTROPHE", and "CATERPILLAR".
- How can we score a document?
- We simply consider the three terms as "normal" query terms: if a document contains all three of them then it will probably be more relevant.

PHRASE QUERIES

PHRASES IN A "BAG OF WORDS" MODEL

- In the vector space model our documents are "bags of words", without any ordering, while in phrase queries the ordering is important.
- The two models are, in some sense, incompatible: a bag of words model cannot be directly used for phrase queries.
- They can still be combined in some meaningful way:
  - Perform the phrase query and rank only the documents returned by the query.
  - If less than K documents are present then "reduce" the share query and start again.

EVALUATION OF IR SYSTEMS

STANDARD TEST COLLECTIONS

STANDARD BENCHMARKS

CRANFIELD COLLECTION

ONE OF THE OLDEST, NOW TOO SMALL. 1398 ABSTRACTS OF AERODYNAMICS JOURNAL ARTICLES AND 225 QUERIES.

TREC
(TEXT RETRIEVAL CONFERENCE)

NOT A SINGLE COLLECTION. THERE IS A RANGE OF TEXT COLLECTIONS ON DIFFERENT TOPICS.
SEE : HTTPS://TREC.NIST.GOV

REUTERS

REUTERS-21578 (21578 DOCUMENTS) AND REUTERS-RCV1 (806791 DOCUMENTS) COLLECT A LARGE NUMBER OF NEWSWIRE ARTICLES

Also see: http://ir.dcs.gla.ac.uk/resources/test_collections/

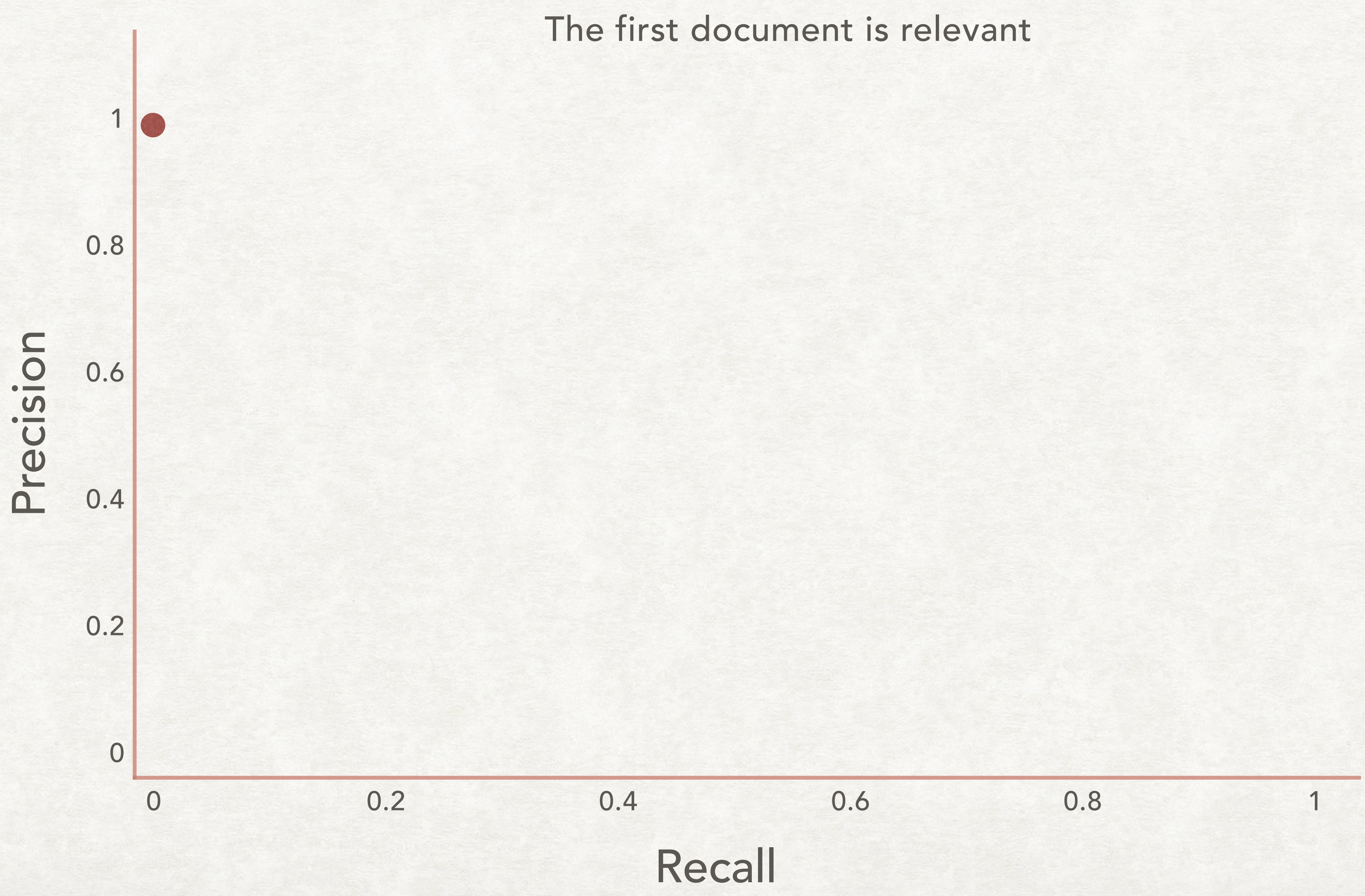
RANKED RETRIEVAL

HOW TO COMPUTE PRECISION AND RECALL?

- We usually evaluate the effectiveness of a IR system with precision and recall (other measures are also possible)…
- …and this works well with *unranked* results.
- How can we extend it to *ranked* results, where position is important?
  - Precision-recall curve and interpolated precision
  - Eleven-point interpolated average precision
  - Mean average precision (MAP)
  - Precision at k and R-precision

PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:

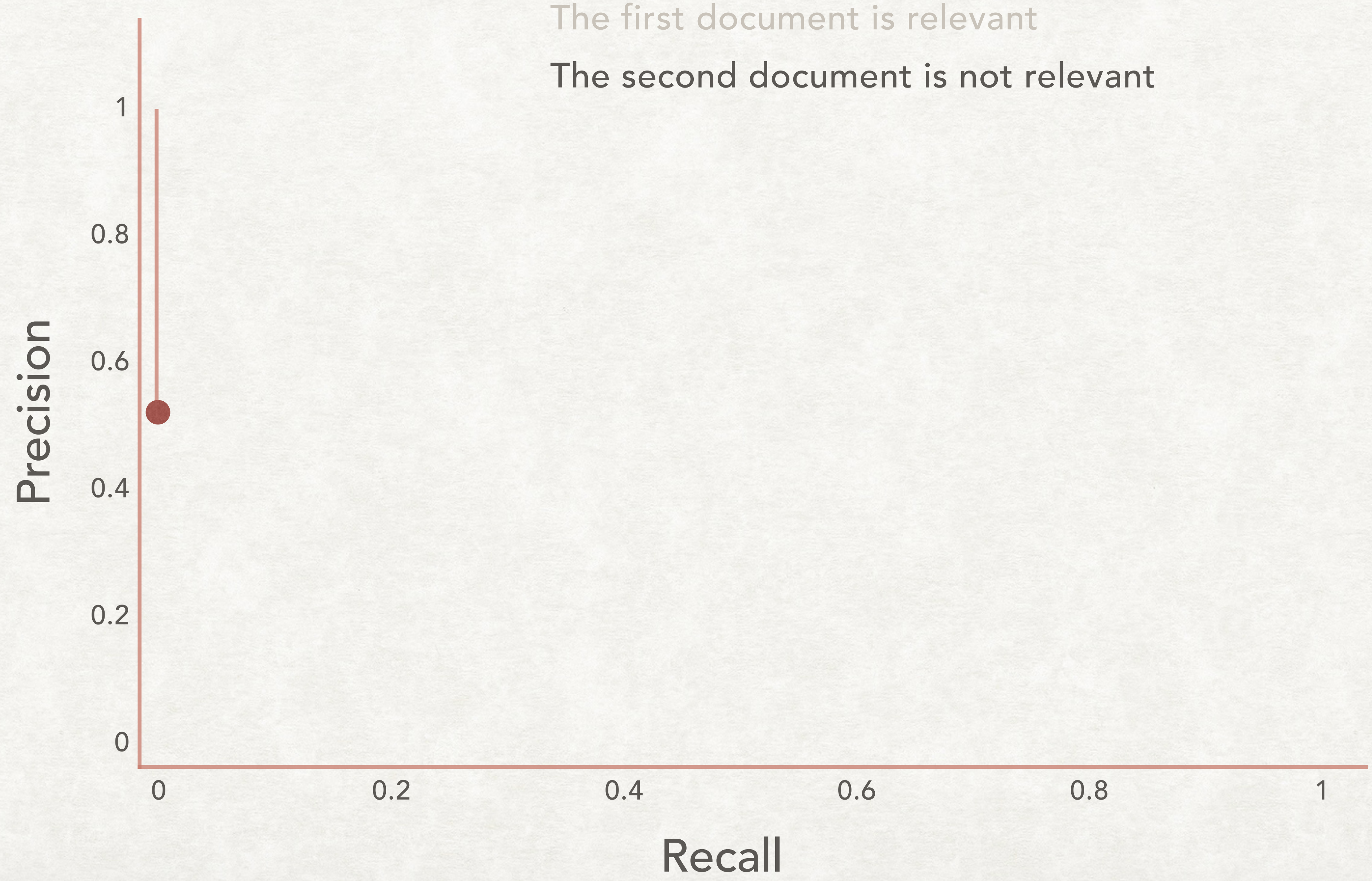


PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:

The first document is relevant

The second document is not relevant



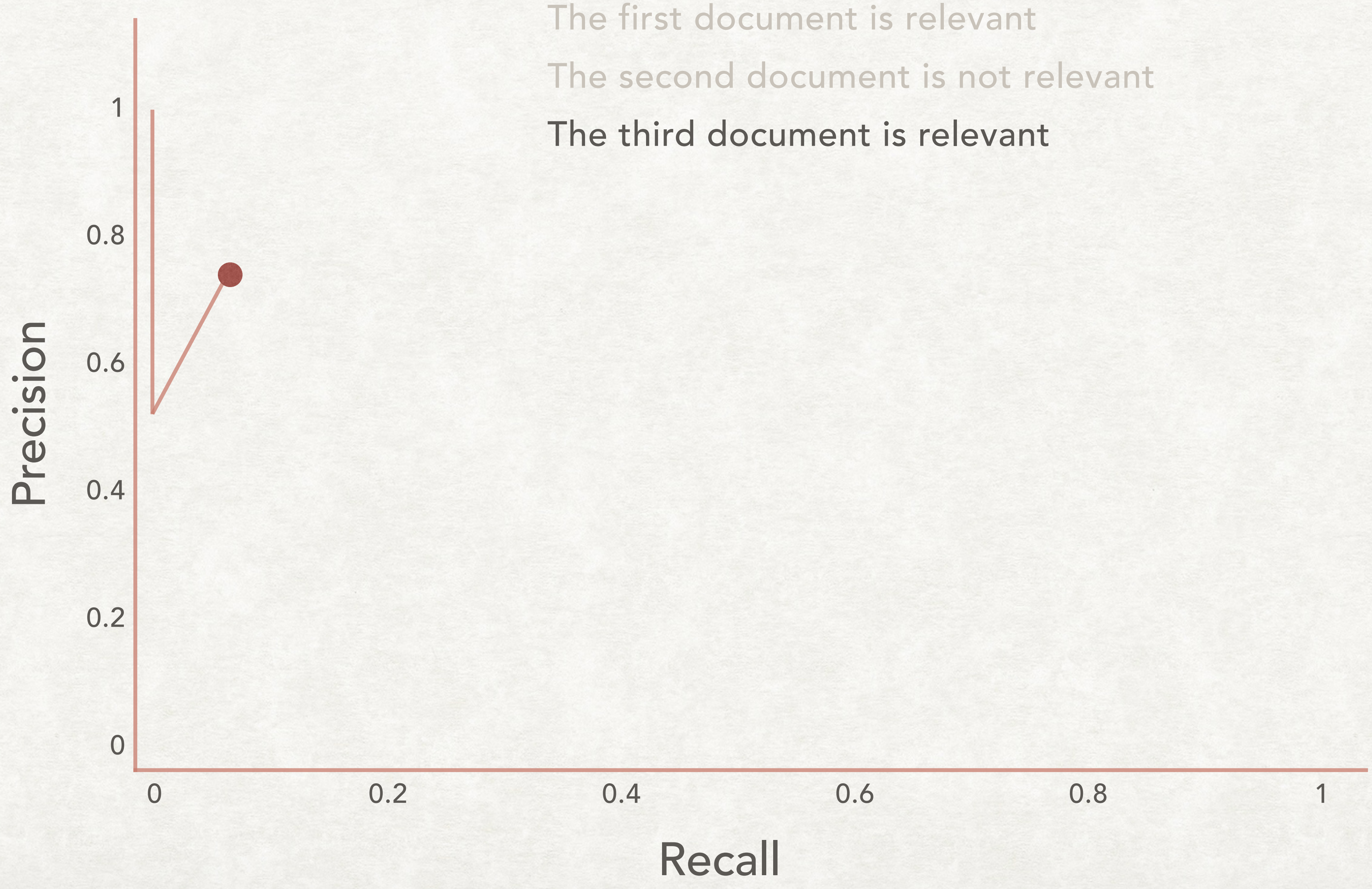
PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:

The first document is relevant

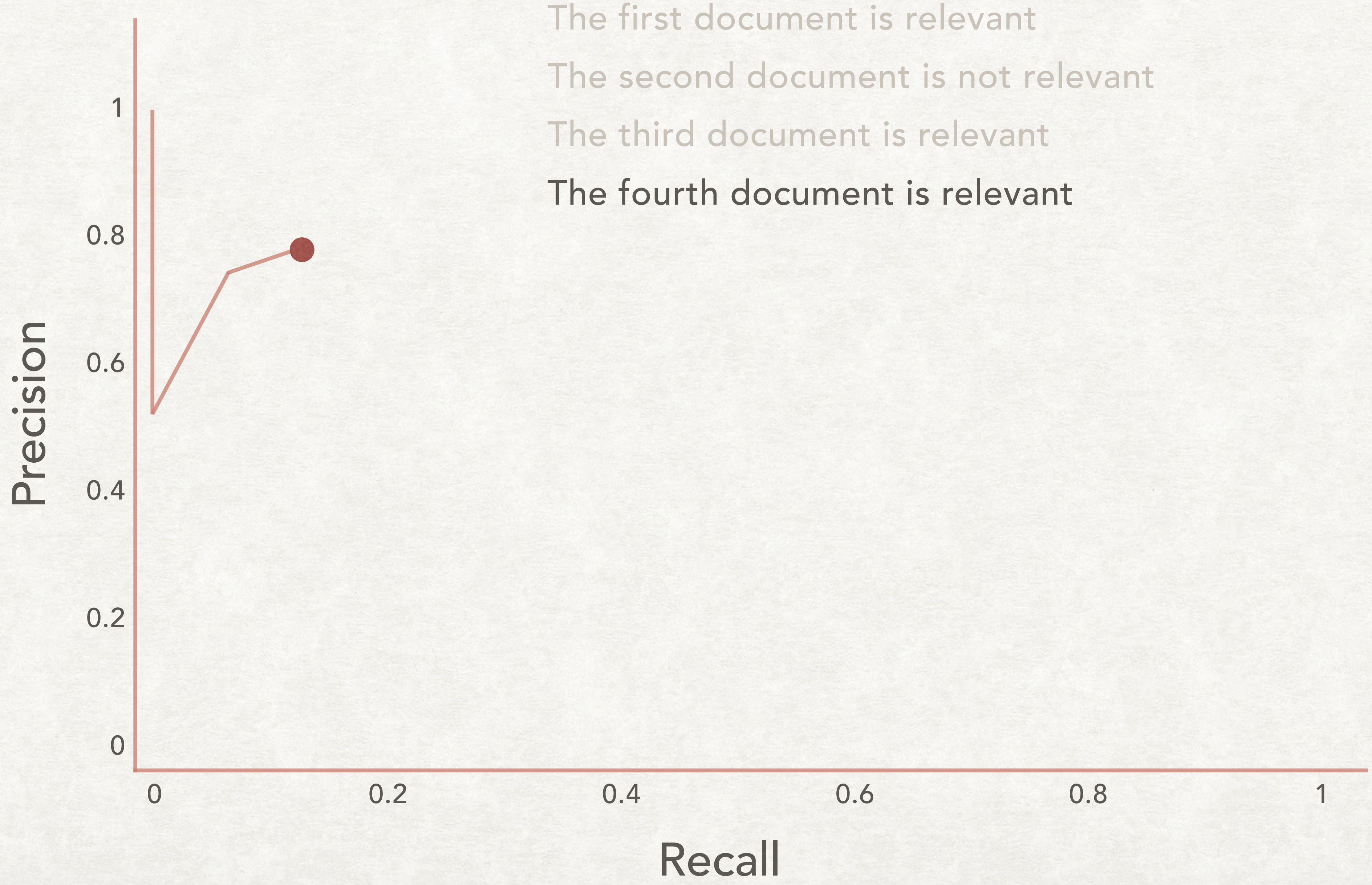
The second document is not relevant

The third document is relevant



PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



The first document is relevant

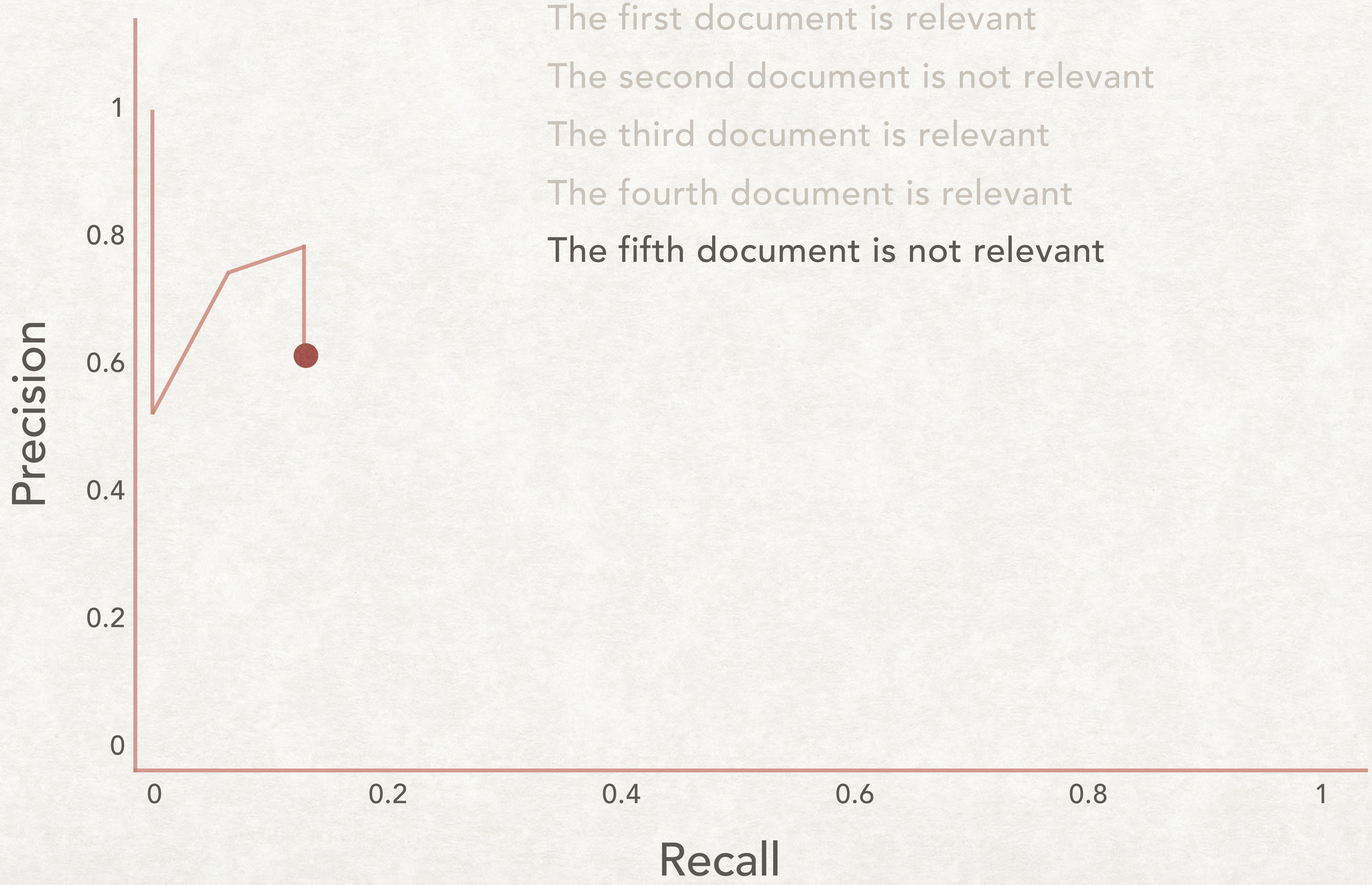
The second document is not relevant

The third document is relevant

The fourth document is relevant

PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



The first document is relevant

The second document is not relevant

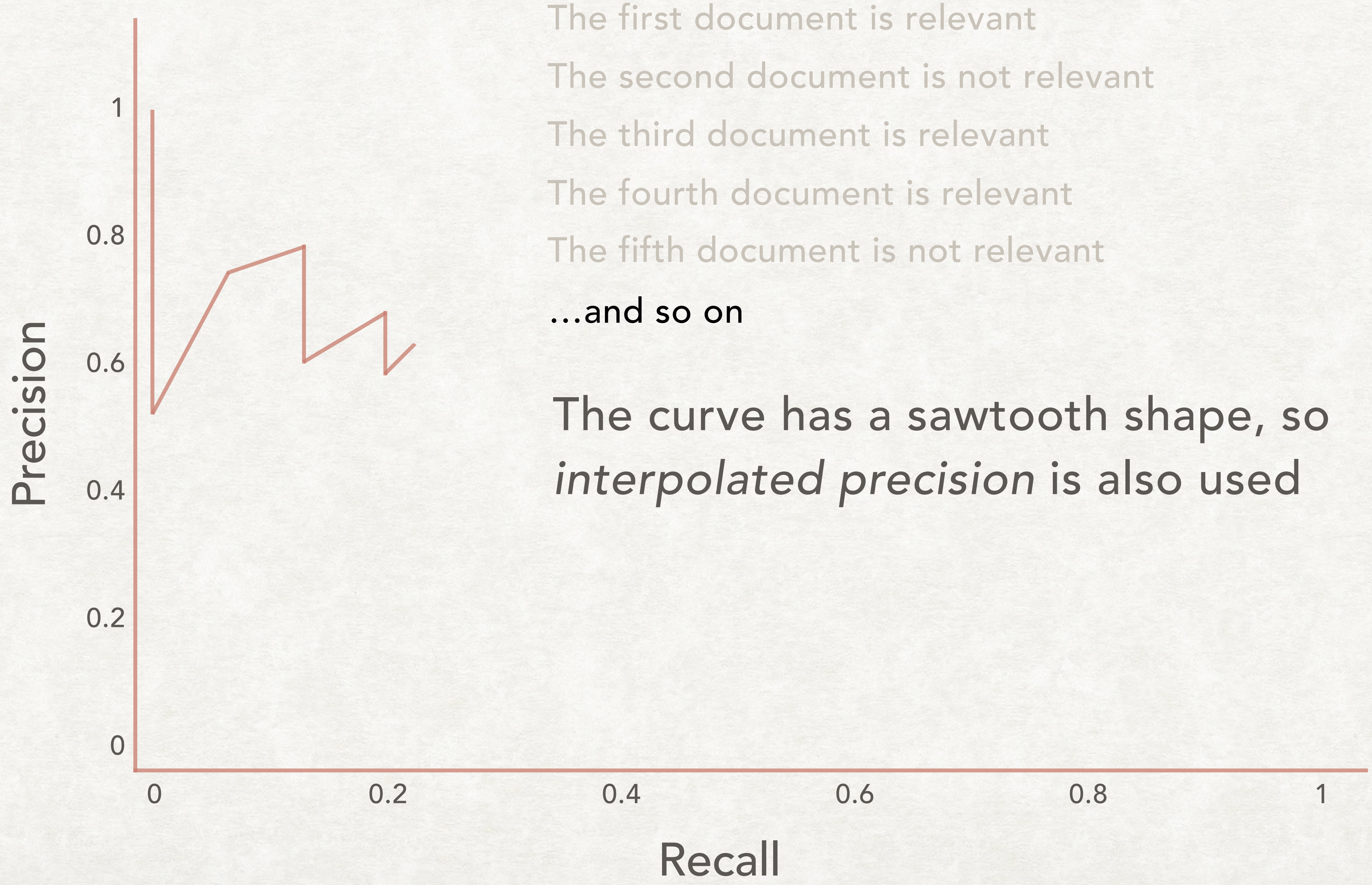
The third document is relevant

The fourth document is relevant

The fifth document is not relevant

PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



The first document is relevant

The second document is not relevant

The third document is relevant

The fourth document is relevant

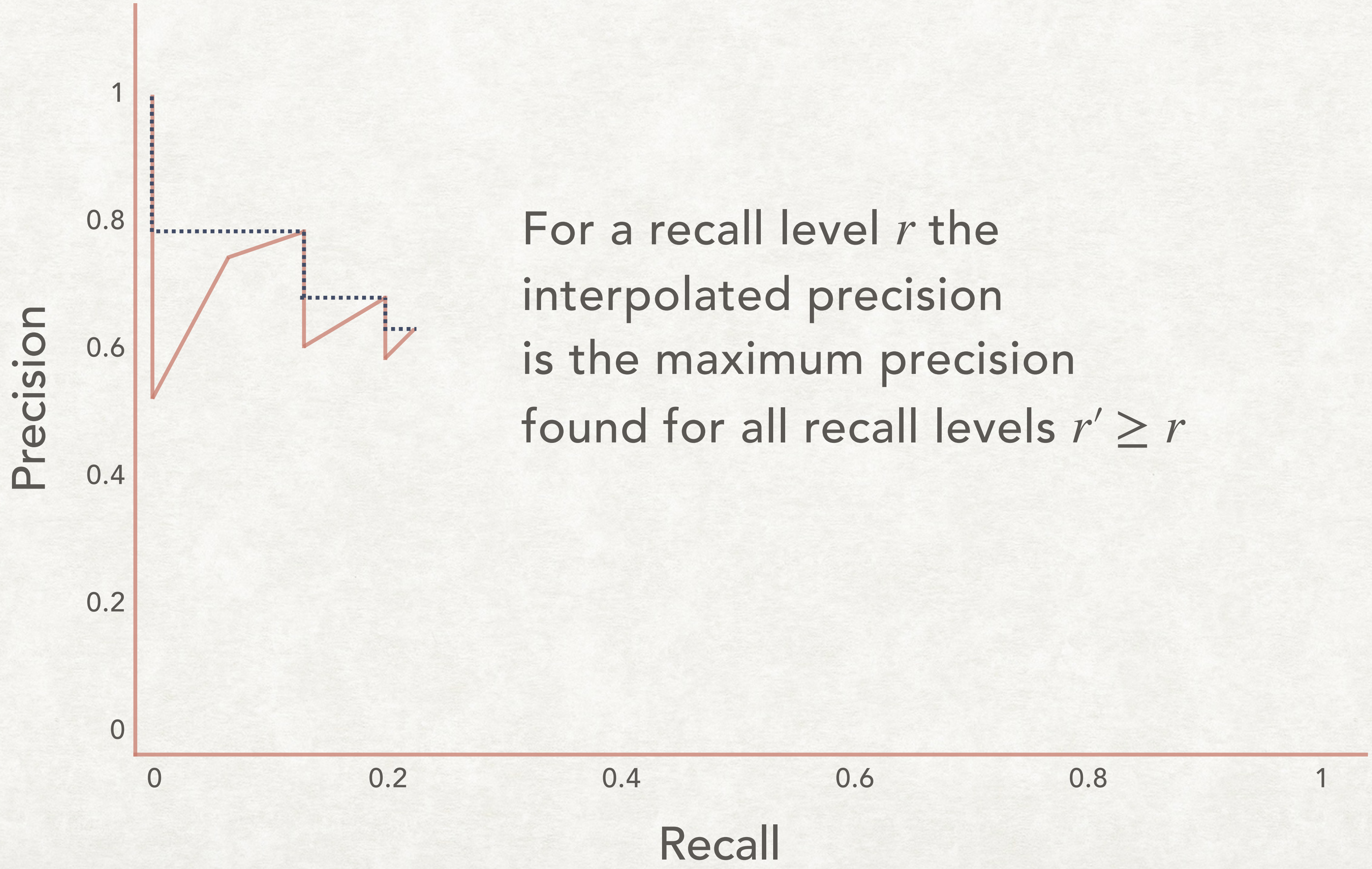
The fifth document is not relevant

...and so on

The curve has a sawtooth shape, so *interpolated precision* is also used

PRECISION-RECALL CURVE

We compute precision and recall for the first 1, 2, 3, 4, etc. retrieved documents:



For a recall level r the interpolated precision is the maximum precision found for all recall levels $r' \geq r$

ELEVEN POINT INTERPOLATED PRECISION

PRECISION AT ELEVEN RECALL LEVELS

Recall	Precision
0,0	1,0
0,1	0,73
0,2	0,64
0,3	0,58
0,4	0,51
0,5	0,45
0,6	0,38
0,7	0,27
0,8	0,21
0,9	0,13
1,0	0,09

The recall levels are fixed and for each recall level the corresponding precision is recorded.

MEAN AVERAGE PRECISION

A SINGLE FIGURE

We have a set of queries $Q = \{q_1, \ldots, q_n\}$

For each q_j we know the set of documents $\{d_1, \ldots, d_{m_j}\}$ that are relevant

Let R_{jk} the set of ranked documents retrieved for the j^{th} query that we get to obtain k relevant documents

Then the mean average precision $\text{MAP}(Q)$ is:

$$\frac{1}{n}\sum_{j=1}^{n}\left(\underline{\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk})}\right)$$

Average precision of the j^{th} query

PRECISION AT K AND R-PRECISION

OTHER SINGLE FIGURES

- Precision at k simply means that we record the precision of the first k retrieved documents. Like "precision at 10".
- If there are less than k relevant documents then the value cannot be one. Its value is highly dependant on the number of relevant documents that exists.
- A solution to this is the R-precision. If there are R relevant documents for a query, the R-precision is the precision of the top R ranked documents returned by the query.
- R-precision can be averaged across queries.