INFORMATION RETRIEVAL

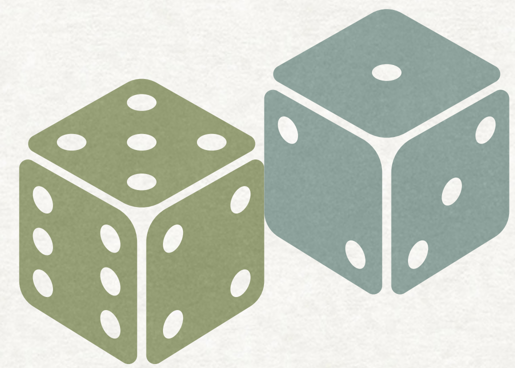
Luca Manzoni

lmanzoni@units.it

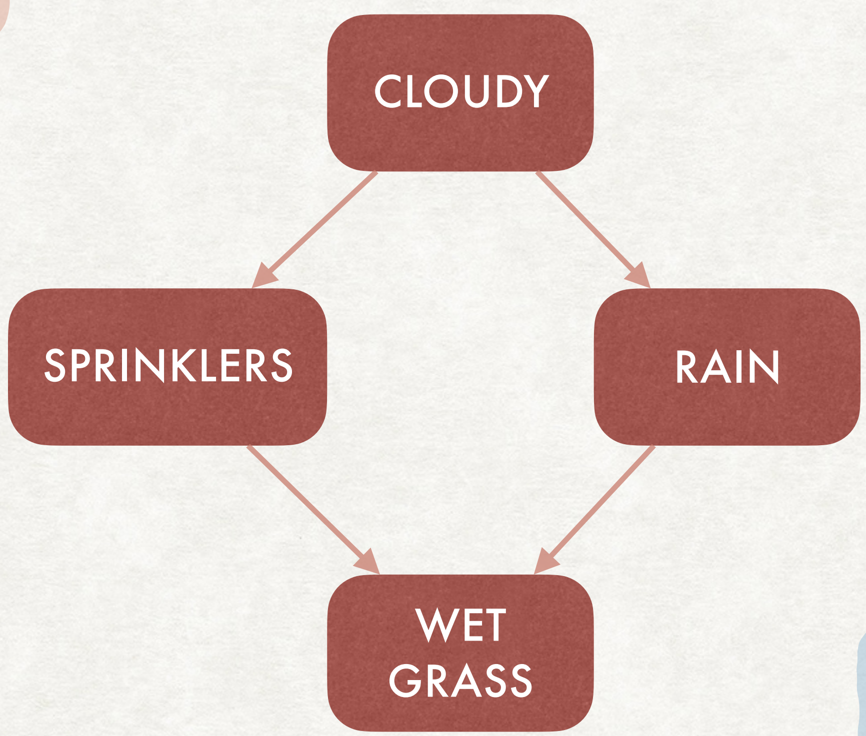
Lecture 8

LECTURE OUTLINE

*SUBTITLE INTENTIONALLY LEFT BLANK



Bayesian Networks



Probabilistic Information Retrieval

Using Bayesian networks for information retrieval

PROBABILISTIC INFORMATION RETRIEVAL

PROBABILISTIC IR

MAIN IDEAS

- If we know some relevant and some non-relevant documents for a query we can estimate the probability of a document to be relevant given the terms it contains.
- This is the main idea of a probabilistic model of IR: estimate probabilities of a document being relevant with respect to a query based on its content.
- There will be some assumptions to simplify the computation of this probability…
- …and some estimates: we do not known most of the probabilities involved!

A QUICK REVIEW

BASICS OF PROBABILITY THEORY

- The probability of A and B can be written as a conditional probability:
  $P(A, B) = P(A \mid B)P(B) = P(B \mid A)P(A)$

- The probability of B and A plus the probability of B and not A is simply the probability of B:
  $P(B) = P(B, A) + P(B, \overline{A})$

- The *odds* of an event A is defined as:
  $$O(A) = \frac{P(A)}{P(\overline{A})} = \frac{P(A)}{1 - P(A)}$$

A QUICK REVIEW

BASICS OF PROBABILITY THEORY

- The classical Bayes' rule is:
- $$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)}{\sum_{X \in [A,\bar{A}]} P(B|X)P(X)} P(A)$$
- Which can be interpreted as:
  - Given the prior probability $P(A)$ of A…
  - …how we can update it based on the evidence B, thus obtaining a posterior probability $P(A|B)$.

PROBABILITY RANKING PRINCIPLE

AND THE BASIS FOR PROBABILISTIC IR

For each document we consider the random variable $R_{d,q}$ (or R for short) representing wether a document is relevant to not.

We want to rank documents according to their probability of being relevant to a given query q:

$$P(\underbrace{R=1}_{\text{Probability of having something relevant}}\,|\,\underbrace{d,q}_{\text{Given that the document is } d \text{ and the query is } q})$$

Probability of having something relevant

Given that the document is d and the query is q

1/0 LOSS

AN THE OPTIMAL DECISION RULE

The simples case:

- Penalty when we retrieve a document that is not relevant.
- Penalty when we miss a relevant document.
- The penalty is the same in all cases, there are no costs associated to retrieving documents.

If we need to rank documents then we rank them by decreasing $P(R = 1 \mid d, q)$.

If we need to return a *set* of documents we return all then ones where $P(R = 1 \mid d, q) > P(R = 0 \mid d, q)$.

It can be proved that this choice minimise the *expected loss* under the 1/0 loss.

RETRIEVAL COSTS

MORE THAN THE 1/0 LOSS

We can also have a more complex model for costs:

- C_1 is the cost of retrieving a relevant document.
- C_0 is the cost of retrieving a non-relevant document

Then to select the document to be retrieved d we must the one where for all *non-retrieved* documents d' it holds that:

$$\underbrace{C_1 \cdot P(R = 1 \mid d, q) + C_0 \cdot P(R = 0 \mid d, q)}_{\text{Weighted cost of retrieving } d} \leq \underbrace{C_1 \cdot P(R = 1 \mid d', q) + C_0 \cdot P(R = 0 \mid d', q)}_{\text{Weighted cost of retrieving } d'}$$

THE BINARY INDEPENDENCE MODEL

THE BINARY INDEPENDENCE MODEL

OR "BIM"

Binary	Or "Boolean". Each document (and query) is represented as a vector $\overrightarrow{x} = (x_1, \ldots, x_M)$ where $x_i = 1$ if the term is present and $x_i = 0$ otherwise
Independence	We assume that all terms occurs in a document independently. Not a correct assumption, but "it works"

Additionally, we assume the relevant of a document to be independent on the relevance of other documents.
This is not true in practice: e.g., duplicate and near-duplicate documents are not independent.

ESTIMATION OF THE PROBABILITY

$$P(R = 1 \mid d, q)$$

In out model this is given by

WE DO NOT KNOW THE EXACT VALUE, WE WILL NEED TO PROVIDE ESTIMATES!

$$P(R = 1 \mid \vec{x}, \vec{q})$$

By Bayes' rule

$$\frac{P(\vec{x} \mid R = 1, \vec{q})\, P(R = 1 \mid \vec{q})}{P(\vec{x} \mid \vec{q})}$$

Probability for a document with representation $\vec{x}$ is retrieved given that a relevant document for the query q is retrieved

Probability of retrieving a relevant document for the query q

ESTIMATION OF THE PROBABILITY

$$P(R = 0 \mid d, q)$$

In out model this is given by

$$P(R = 0 \mid \vec{x}, \vec{q})$$

By Bayes' rule

$$\frac{P(\vec{x} \mid R = 0, \vec{q})\, P(R = 0 \mid \vec{q})}{P(\vec{x} \mid \vec{q})}$$

Probability for a document with representation $\vec{x}$ is retrieved given that a **non**-relevant document for the query q is retrieved

Probability of retrieving a **non**-relevant document for the query q

DO WE REALLY NEED TO KNOW THE PROBABILITY?

FOR RANKING ODDS ARE SUFFICIENT

For the purpose of ranking, we can use a monotone function of the probability. For example, the odds of R given $\overrightarrow{x}$ and $\overrightarrow{q}$:

$$O(R \mid \overrightarrow{x}, \overrightarrow{q}) = \frac{P(R=1 \mid \overrightarrow{x}, \overrightarrow{q})}{P(R=0 \mid \overrightarrow{x}, \overrightarrow{q})}$$

$$\frac{\dfrac{P(\overrightarrow{x} \mid R=1, \overrightarrow{q})\, P(R=1 \mid \overrightarrow{q})}{P(\overrightarrow{x} \mid \overrightarrow{q})}}{\dfrac{P(\overrightarrow{x} \mid R=0, \overrightarrow{q})\, P(R=0 \mid \overrightarrow{q})}{P(\overrightarrow{x} \mid \overrightarrow{q})}} \rightarrow \frac{P(\overrightarrow{x} \mid R=1, \overrightarrow{q})\, P(R=1 \mid \overrightarrow{q})}{P(\overrightarrow{x} \mid R=0, \overrightarrow{q})\, P(R=0 \mid \overrightarrow{q})}$$

CAN WE SIMPLIFY IT FURTHER?

RANKING AND PROBABILITIES

$$\boxed{\frac{P(\overrightarrow{x}\,|\,R=1,\overrightarrow{q})}{P(\overrightarrow{x}\,|\,R=0,\overrightarrow{q})}}\boxed{\frac{P(R=1\,|\,\overrightarrow{q})}{P(R=0\,|\,\overrightarrow{q})}}$$

Depends on the document

The same for all documents

Does not affect the ranking

We can remove it

We now have to estimate:

$$\frac{P(\overrightarrow{x}\,|\,R=1,\overrightarrow{q})}{P(\overrightarrow{x}\,|\,R=0,\overrightarrow{q})}$$

USING THE BIM

$$\frac{P(\overrightarrow{x} \mid R = 1, \overrightarrow{q})}{P(\overrightarrow{x} \mid R = 0, \overrightarrow{q})}$$

We can now employ the independence assumption: each of the terms is assumed to appear independently from the others

$$\frac{P(x_1 \mid R = 1, \overrightarrow{q})}{P(x_1 \mid R = 0, \overrightarrow{q})} \times \frac{P(x_2 \mid R = 1, \overrightarrow{q})}{P(x_2 \mid R = 0, \overrightarrow{q})} \times \cdots \times \frac{P(x_M \mid R = 1, \overrightarrow{q})}{P(x_M \mid R = 0, \overrightarrow{q})}$$

Which means the the value to estimate is now:

$$\prod_{i=1}^{M} \frac{P(x_i \mid R = 1, \overrightarrow{q})}{P(x_i \mid R = 0, \overrightarrow{q})}$$

SPLITTING UP FURTHER

$$\prod_{i=1}^{M} \frac{P(x_i \mid R=1, \overrightarrow{q})}{P(x_i \mid R=0, \overrightarrow{q})}$$

Each x_i can only assume two values:

0 if the i^{th} term is not present

1 if the i^{th} term is present

$$\prod_{i:x_i=1} \frac{P(x_i=1 \mid R=1, \overrightarrow{q})}{P(x_i=1 \mid R=0, \overrightarrow{q})} \cdot \prod_{i:x_i=0} \frac{P(x_i=0 \mid R=1, \overrightarrow{q})}{P(x_i=0 \mid R=0, \overrightarrow{q})}$$

For the terms in the document

For the terms not in the document

HOW MANY PROBABILITIES TO ESTIMATE?

$$\prod_{i:x_i=1} \frac{P(x_i = 1 \,|\, R = 1, \overrightarrow{q})}{P(x_i = 1 \,|\, R = 0, \overrightarrow{q})} \cdot \prod_{i:x_i=0} \frac{P(x_i = 0 \,|\, R = 1, \overrightarrow{q})}{P(x_i = 0 \,|\, R = 0, \overrightarrow{q})}$$

For each term we need only to estimate four probabilities:

	Document relevant	Document not relevant
Term present	p_i	u_i
Tern absent	$1 - p_i$	$1 - u_i$

SIMPLIFYING FURTHER

$$\prod_{i:x_i=1} \frac{p_i}{u_i} \cdot \prod_{i:x_i=0} \frac{1-p_i}{1-u_i}$$

Let us assume that all query terms **not** in the query appears equally in relevant and non-relevant documents. That is, $p_i = u_i$ when $q_i = 0$.

We can remove the factors for all terms not in the query, obtaining:

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i} \cdot \prod_{i:x_i=0;q_i=1} \frac{1-p_i}{1-u_i}$$

SIMPLIFYING FURTHER

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i} \cdot \prod_{i:x_i=0;q_i=1} \frac{1-p_i}{1-u_i}$$

We now multiply everything by

Each term is actually 1.

$$\prod_{i:x_i=1;q_i=1} \frac{1-p_i}{1-u_i} \cdot \frac{1-u_i}{1-p_i}$$

By rearranging the factors we obtain:

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i} \frac{1-u_i}{1-p_i} \cdot \prod_{i:q_i=1} \frac{1-p_i}{1-u_i}$$

SIMPLIFYING FURTHER

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i}\frac{1-u_i}{1-p_i} \cdot \prod_{i:q_i=1} \frac{1-p_i}{1-u_i}$$

This part does **not** depend on the document!
We can remove it

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i}\frac{1-u_i}{1-p_i}$$

RATIO OF ODDS

$$\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i} \frac{1-u_i}{1-p_i}$$

Each factor can be seen as two odds:

$$\frac{p_i}{1-p_i}$$

Odds of the term appearing
in the document if
the document is relevant

$$\frac{1-u_i}{u_i}$$

Inverse odds of the term
appearing in the document if
the document is **not** relevant

RETRIEVAL STATUS VALUE

The **Retrieval Status Value (RSV) of a document** d is defined as the logarithm of the quantity that we now have:

$$\mathrm{RSV}_d = \log\left(\prod_{i:x_i=1;q_i=1} \frac{p_i}{u_i}\frac{1-u_i}{1-p_i}\right)$$

$$= \sum_{i:x_i=1;q_i=1} \log \frac{p_i}{u_i}\frac{1-u_i}{1-p_i}$$

RETRIEVAL STATUS VALUE

Consider each term of the sum:

$$c_i = \log \frac{p_i}{u_i} \frac{1 - u_i}{1 - p_i}$$

Which can be rewritten as a **log odds ratio**:

$$c_i = \log \frac{p_i}{1 - p_i} + \log \frac{1 - u_i}{u_i}$$

c_i can be considered the **weight** of the i^{th} term of the dictionary, and can be pre-computed (like other measures like the inverse document frequency)

RETRIEVAL STATUS VALUE

At the end the RSV of a document d can be written as:

$$\mathrm{RSV}_d = \sum_{i: x_i = q_i = 1} c_i$$

Which algorithmically, can be described as:

> *To compute the RSV of a document d, sum the weight c_i of each term contained in both the document and the query*

We now need a way to estimate the various probabilities to (pre-)compute all c_i.

PROBABILITY ESTIMATION IN PRACTICE

ESTIMATION FOR NON-RELEVANT DOCUMENTS

- We assume that non-relevant documents are a majority inside the collection.
- Thus, we approximate the probability for non-relevant documents with statistics computed using the entire collection.
- Usually $\log \frac{1-u_i}{u_i} = \log \frac{N-\mathrm{df}_i}{\mathrm{df}_\mathrm{i}}$ for a term i.
- Which is approximately $\log \frac{N}{\mathrm{df}_i}$, which is actually the inverse document frequency idf_i for the term i.

ESTIMATION FOR RELEVANT DOCUMENTS

- Estimation for relevant documents is more complex. There are multiple approaches used in practice:

- We can estimate the probabilities by looking at statistics on a set of relevant documents that we have obtained in some way.

- We can put all probabilities equal to 0.5. With this estimate and assuming idf_i for non-relevant documents, this approximation is the sum of the idf_i for all query terms that occurs in the document.

- Another possibility is using some collection level statistics, for example obtaining $p_i = \dfrac{\mathrm{df}_i}{N}$.

COMBINATION WITH RELEVANCE FEEDBACK

We can combine relevance feedback to help us estimate the probability used in computing the RSV_d:

1. Start with probabilities estimated as before
2. Retrive a set V of documents
3. The user classifies the documents retrieved and gives us a set of relevant documents: $VR = \{d \in V \colon R_{d,q} = 1\}$
4. Re-compute our estimates for p_i and u_i

COMBINATION WITH RELEVANCE FEEDBACK

RE-COMPUTING ESTIMATES

If VR is large enough we can use the following updating:

For each i let VR_i be the set of relevant documents containing the i^{th} term:

$$p_i = \frac{|VR_i|}{|VR|} \qquad u_i = \frac{\mathrm{df}_i - |VR_i|}{N - |VR|}$$

However in most case the set of documents evaluated by the user is not large, so we use a "smoothed" version:

$$p_i = \frac{|VR_i| + \frac{1}{2}}{|VR| + 1} \qquad u_i = \frac{\mathrm{df}_i - |VR_i| + \frac{1}{2}}{N - |VR| + 1}$$

COMBINATION WITH RELEVANCE FEEDBACK

PSEUDO-RELEVANCE FEEDBACK

We can extend the previous model to allow for pseudo-relevance feedback.

Select the first k highest ranked documents, consider them as a set V

Consider all of them relevant, and update the probability accordingly (simply substituting VR with V in the previous equations):

$$p_i = \frac{|V_i| + \frac{1}{2}}{|V| + 1} \qquad u_i = \frac{\mathrm{df}_i - |V_i| + \frac{1}{2}}{N - |V| + 1}$$

Repeat until the ranking converges

OKAPI BM25

OKAPI BM25

AKA BM25 WEIGHTING OR OKAPI WEIGHTING

This model is non-binary, since it takes into account the *frequency* of the terms inside the document.

We start with:

$$\mathrm{RSV}_d = \sum_{t \in q} \mathrm{idf}_t$$

Recall that this is the formula that we obtain with one of our estimates.

We now need a way to add information about the term frequencies

OKAPI BM25

AKA BM25 WEIGHTING OR OKAPI WEIGHTING

Let L_d be the length of the document and L_{avg} the average length of the documents in the collection.

$$\mathrm{RSV}_d = \sum_{t \in q} \mathrm{idf}_t \cdot \frac{(k_1 + 1)\mathrm{tf}_{t,d}}{k_1((1 - b) + b \cdot \frac{L_d}{L_{avg}}) + \mathrm{tf}_{t,d}}$$

k_1 and b are two parameters, with $b \in [0,1]$ and $k_1 \geq 0$, usually $k_1 \in [1.2, 2.0]$

OKAPI BM25

AKA BM25 WEIGHTING OR OKAPI WEIGHTING

Let us break up the formula in its components

How much to consider term frequency,
With $k_1 = 0$ we have the binary model

$$\mathrm{RSV}_d = \sum_{t \in q} \mathrm{idf}_t \cdot \frac{(k_1 + 1)\mathrm{tf}_{t,d}}{k_1((1-b) + b \cdot \frac{L_d}{L_{avg}}) + \mathrm{tf}_{t,d}}$$

How much to normalise with respect to length,
regulated by b, with $b = 0$: no normalisation,
with $b = 1$, full scaling by document length

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 $P(y \mid x_1, x_2, \ldots, x_k)$ we actually need to compute only $p(y \mid \mathrm{Pa}(y))$ with $\mathrm{Pa}(y)$ the parent nodes of y.

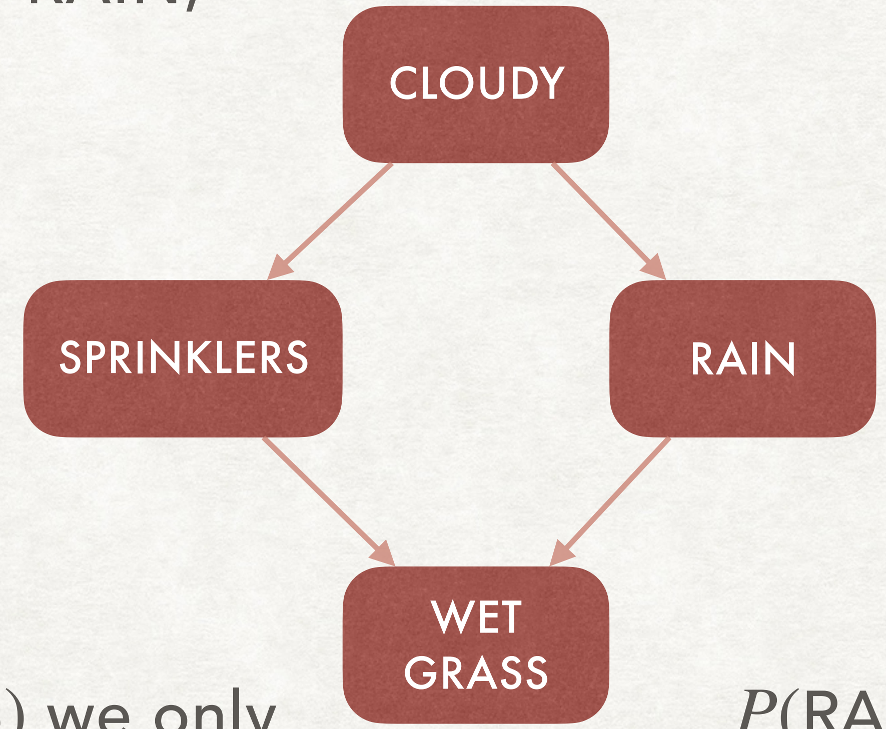
- An example should clarify this.

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 $P(\text{CLOUDY} \mid \text{SPRINKLES})$ we only compute $P(\text{SPRINKLES} \mid \text{CLOUDY})$, and we will have to “rewrite” it.

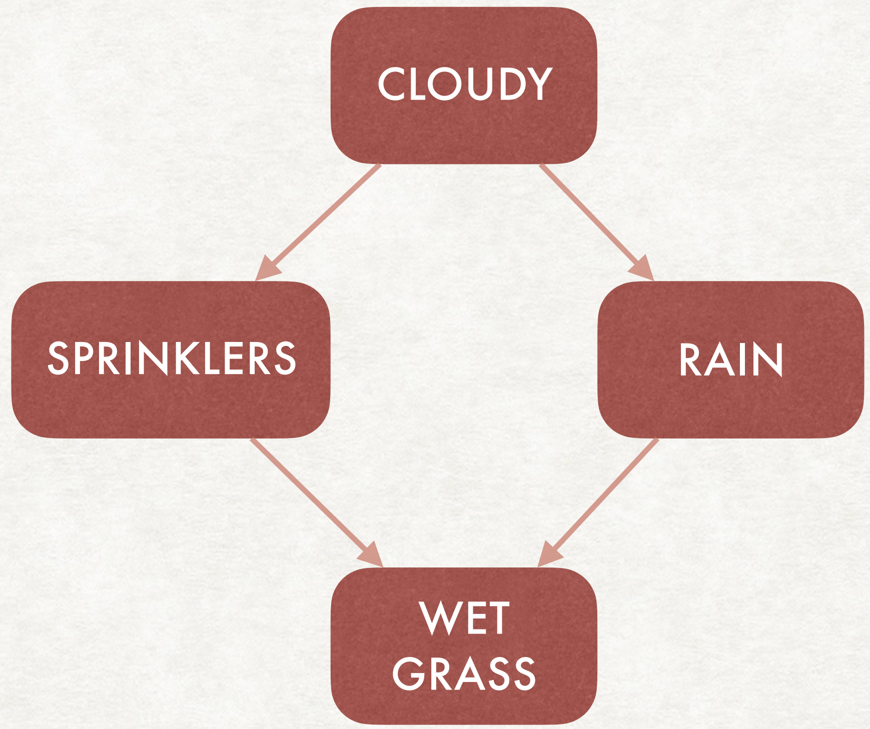
If we want to compute $P(\text{RAIN} \mid \text{CLOUDY})$ we can find it directly in our table

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



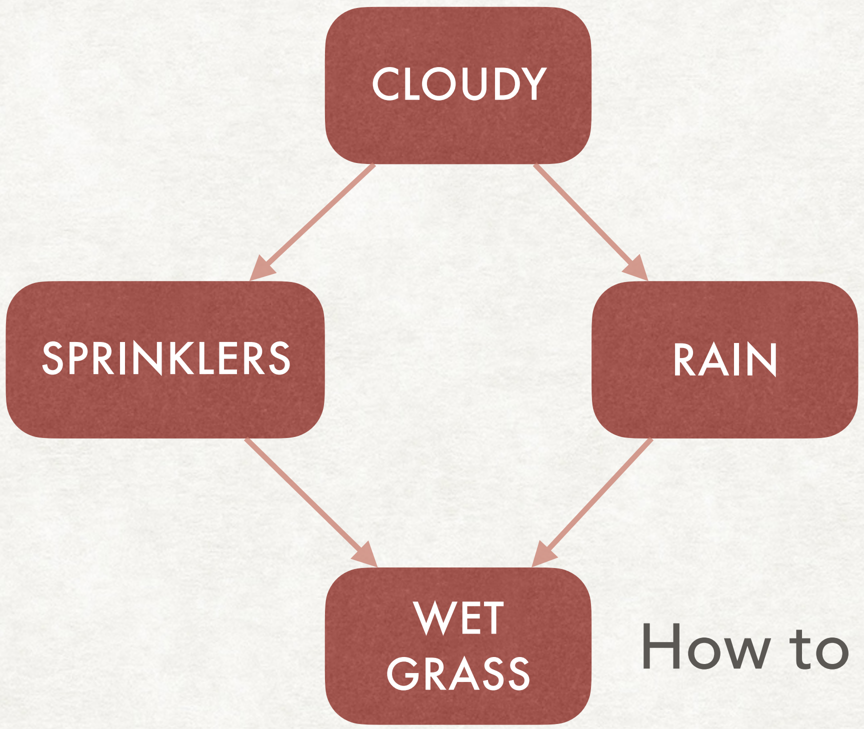
	R = 0	R =1
C = 0	0,8	0,2
C = 1	0,2	0,8

		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	$\boxed{0,9}$
S = 0	R = 1	0,1	0,9
S = 1	R = 1	0,01	0,99

BAYESIAN NETWORKS

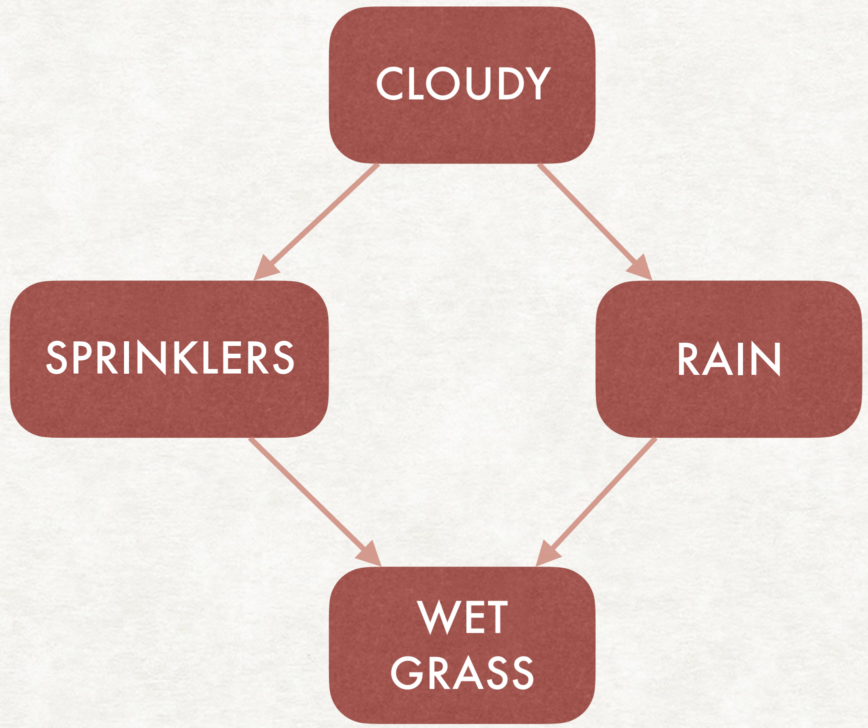
A SIMPLE EXAMPLE

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

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

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

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

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

$$= 0.09$$

BAYESIAN NETWORKS

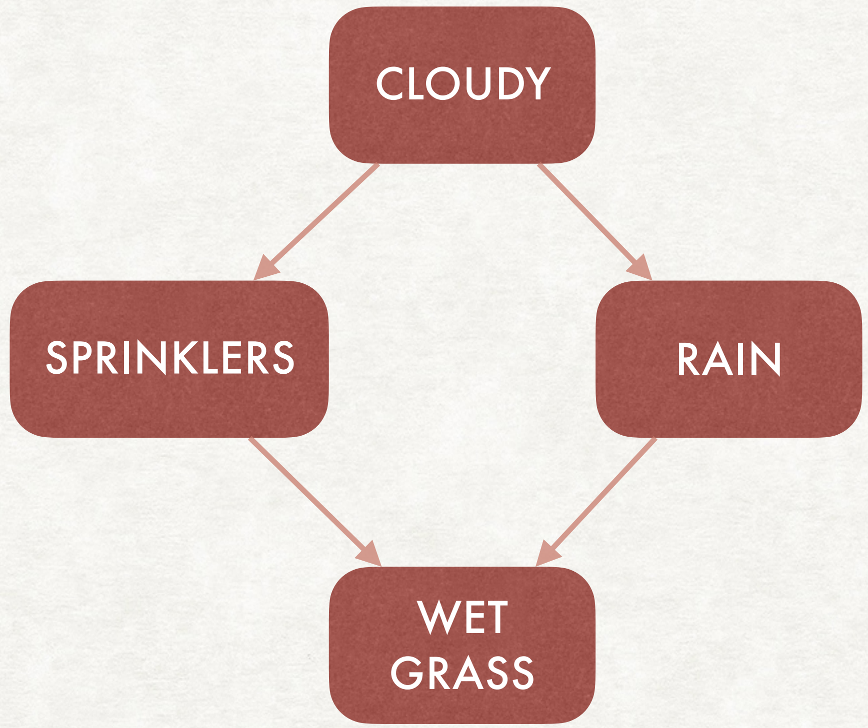
A SIMPLE EXAMPLE

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

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

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

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

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

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

BAYESIAN NETWORKS

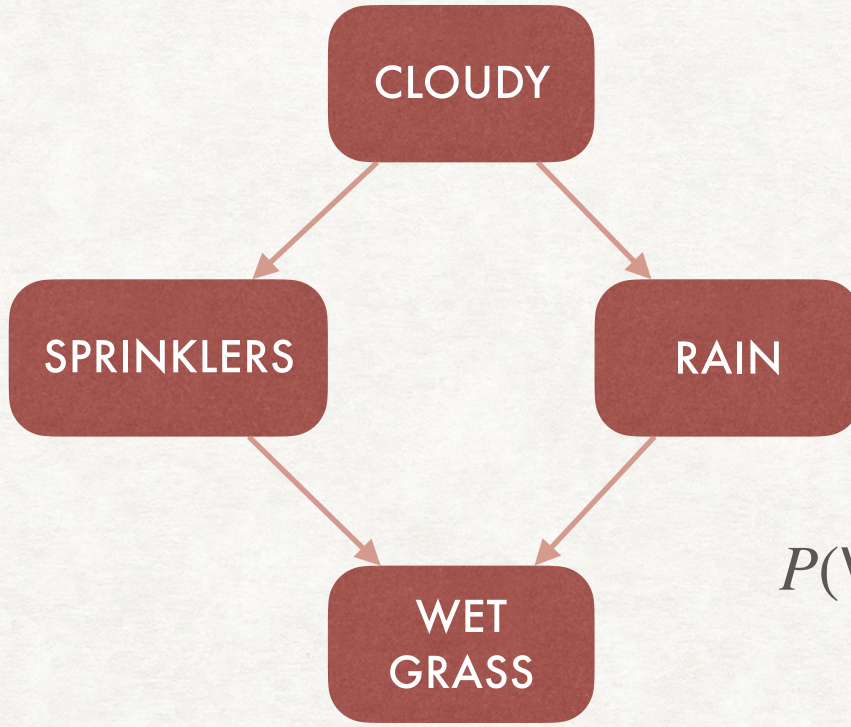
A SIMPLE EXAMPLE

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



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

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

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

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

BAYESIAN NETWORKS

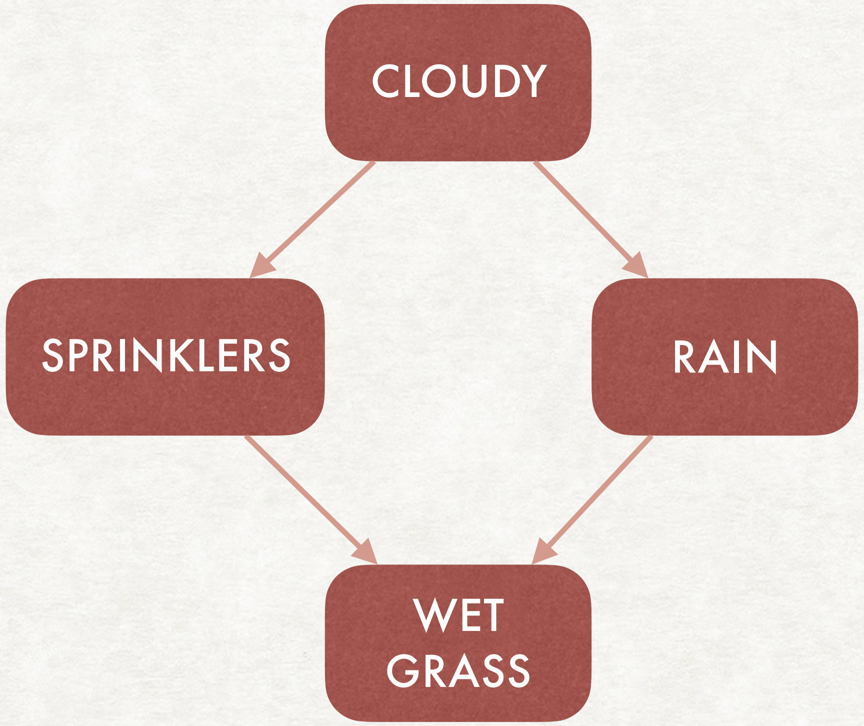
A SIMPLE EXAMPLE

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

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

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

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

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

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

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

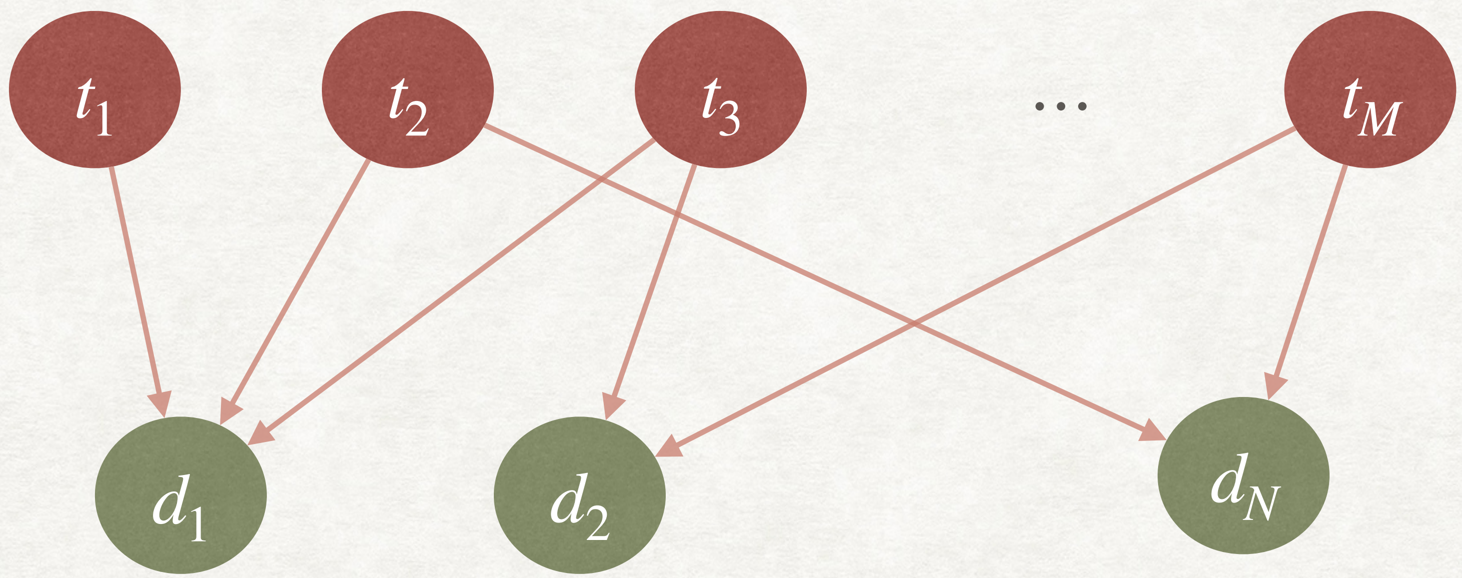
BAYESIAN NETWORKS IN IR

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

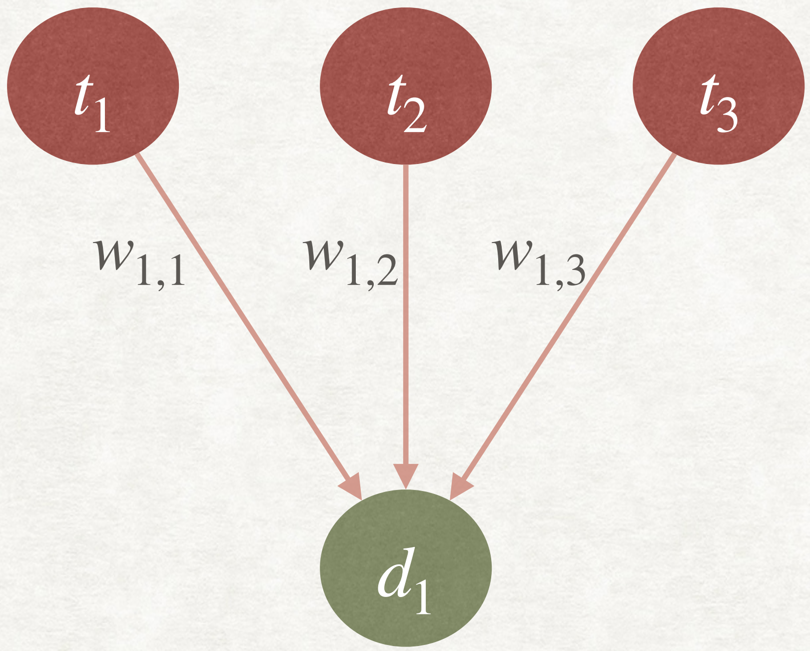
d_j

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 $P(d_j \mid \mathrm{Pa}(d_j))$ is now computed as:

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

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

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

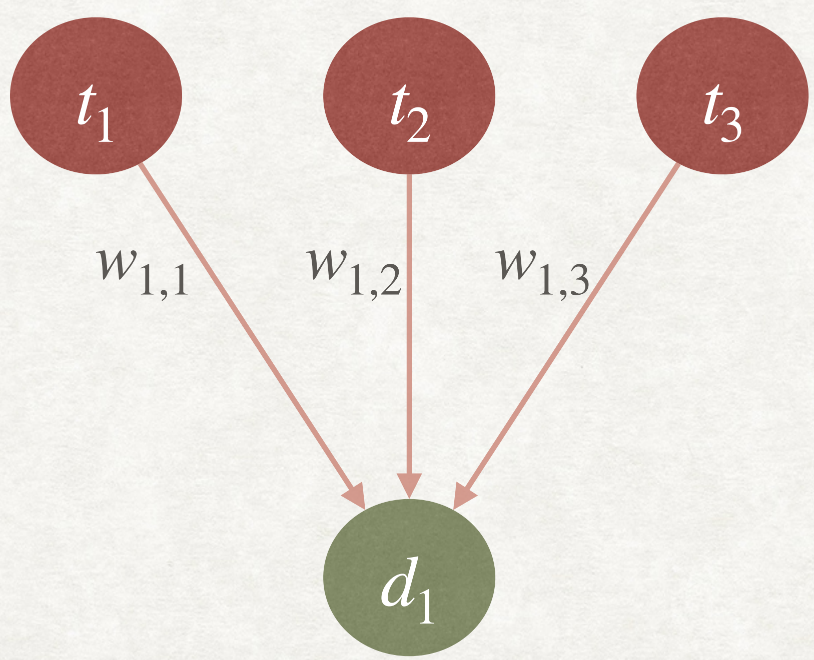
With α a normalising constant

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:



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

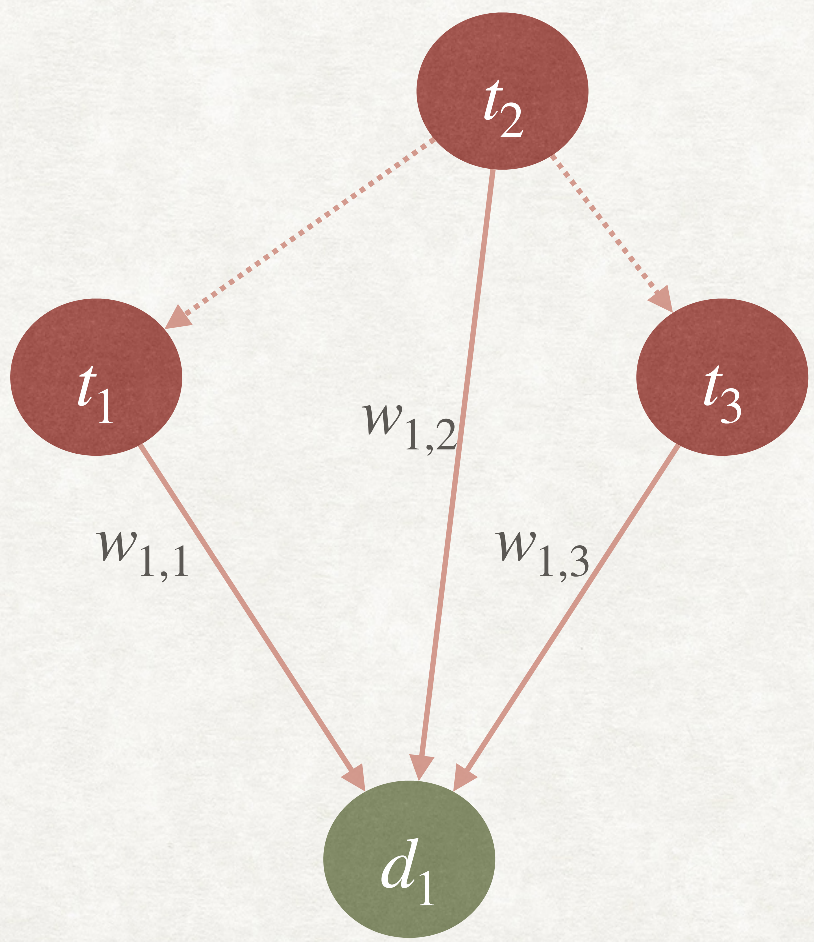
In general:

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

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.

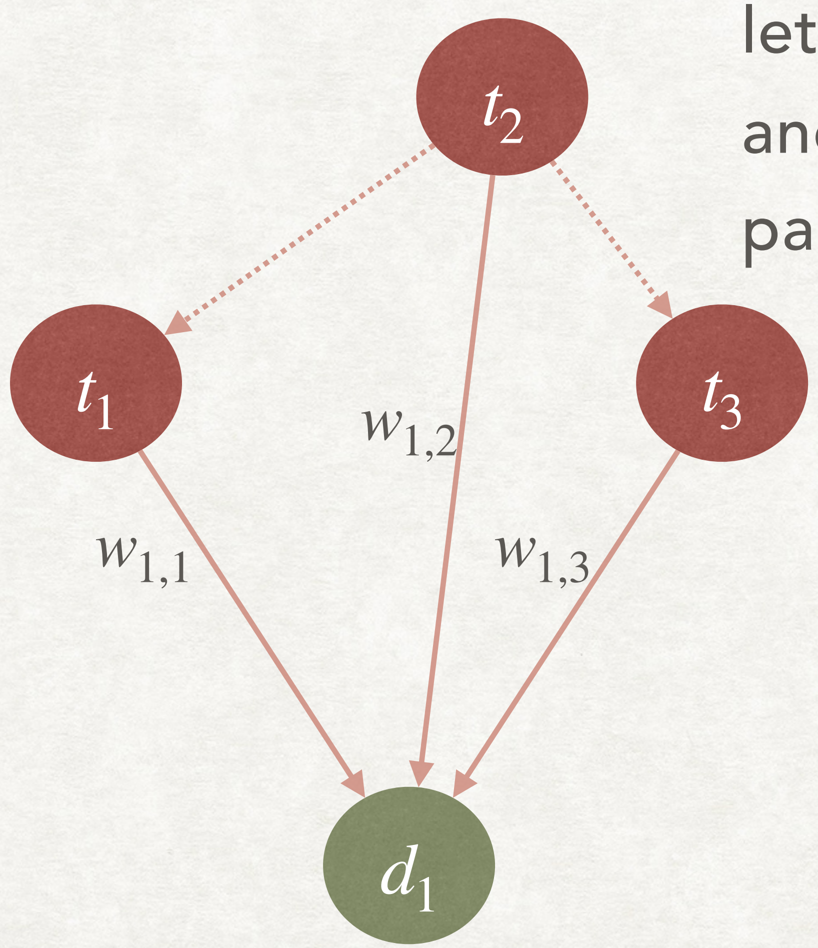
For root nodes we already have:

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".