The Bayesian Network Model

Probability basics:

✓ *Bayes rule* of conditional probability:

- The probability of event A, given event B is

  $P(A \mid B) = P(A \wedge B) / P(B)$

  $= P(A) \cdot P(B \mid A) / P(B)$

✓ The *chain* rule:

- By applying *Bayes rule* twice:

  $P(A \wedge B \wedge C) = P(A \mid B \wedge C) \cdot P(B \mid C) \cdot P(C)$

✓ *Probabilistic independence*:

- *Definition:* A and B are independent if $P(A \wedge B) = P(A) \cdot P(B)$.
- It follows that if A and B are independent, then $P(A \mid B) = P(A)$.

✓ *Conditional independence*:

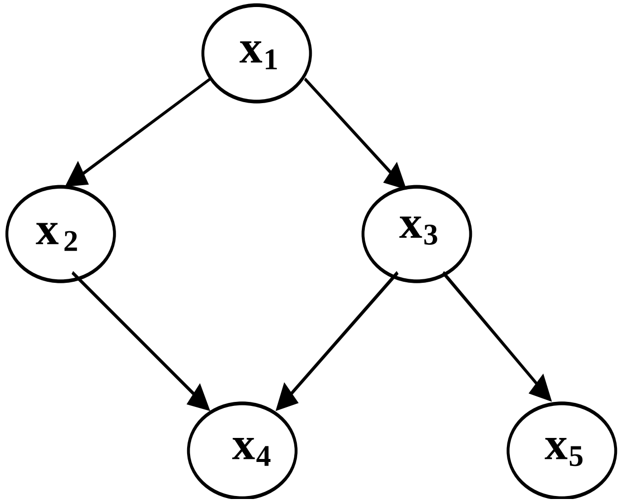
- $P(A \wedge B \mid C) = P(A \mid C) \cdot P(B \mid C)$.
- *Note*: $A \wedge B$ will be denoted A,B.

Bayesian Networks

- ✓ *Bayesian network*: directed acyclic graph (DAG) for illustrating *causal relationships* among variables. In a Bayesian network:
  - Nodes represent random variables.
  - An edge from node Y (parent) to node X (child) represents a dependence between these variables.
  - Each node X is associated with *conditional probability* $P(X \mid Y_1, \ldots, Y_n)$, expressing the *strength* of the dependence of X on its parents $Y_1, \ldots, Y_{n.}$
  - A node does not depend on any nodes but its parents; i.e., if X is parent of Y and Y is parent on Z, then $P(Z \mid X,Y) = P(Z \mid Y)$.

Bayesian Networks *(cont.)*

✓ Example of a Bayesian network:



- $P(X_1)$ is *prior* probability.
- Example of conditional probability:
  Assume X_2 may have two values: *lo*, *hi*,
  assume X_3 may have two values *yes*, *no*,
  and X_4 may have three values: 10, 20, 30.
  Then $P(X_4 \mid X_2, X_3)$ is expressed in a table such as

	lo,yes	*lo,no*	*hi,yes*	*hi,no*
10	0.4	0.5	0.3	0.5
20	0.3	0.2	0.5	0.1
30	0.3	0.3	0.2	0.4

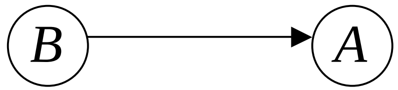
- The *joint* probability $P(X_1, X_2, X_3, X_4, X_5) =$
  $P(X_1) \cdot P(X_2 \mid X_1) \cdot P(X_3 \mid X_1) \cdot P(X_4 \mid X_2, X_3) \cdot P(X_5 \mid X_3)$

Bayesian Networks *(cont.)*

- ✓ *Purpose*: Compute other probabilities. For example,
  - *Prediction*: Given $P(X_1=a)$ (the probability that random variable X_1 attains a certain value), we could calculate the probability $P(X_4=b)$ (the probability that random variable X_4 attains a certain value).
  - *Diagnostics*: Given $P(X_4=b)$ (the probability that random variable X_4 attains a certain value), we can calculate the probability $P(X_1=a)$ (the probability that random variable X_1 attains a certain value),

Bayesian Networks *(cont.)*

✓ Simple example of a Bayesian network:



- B: There is a burglary.
- A: The alarm goes off.
- The prior probability of a burglary is known: $P(B) = 0.0001$.
- The conditional probability of an alarm given a burglary is known:

$P(A \mid B) =$

	Burglary	No burglary	Marginal Probability
Alarm	0.95	0.01	0.01
No alarm	0.05	0.99	0.99

- The probability of the alarm going off is (marginalization):

$$P(A) = 0.95 \cdot 0.0001 + 0.01 \cdot 0.9999 = 0.01$$

- We can compute the posterior probability that there is a burglary if the alarm goes off:

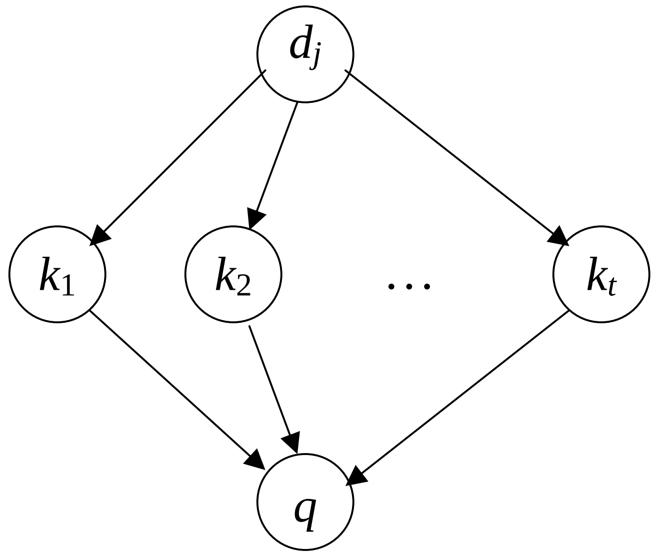
$$P(B \mid A) = P(A \mid B) \cdot P(B) \,/\, P(A) = 0.95 \cdot 0.0001 \,/\, 0.01 = 0.0095$$

(about 95 times higher than the prior probability of a burglary).

Bayesian Networks for IR

Bayesian networks for information retrieval:

- ✓ A node for every term k_i, document d_j, and query q.
- ✓ Two types of edges:
  - Edge from document d_j to term k_i: Term k_i appears in (is relevant to) document d_j.
  - Edge from term k_i to query q: Term k_i appears in (is relevant to) query q.
- ✓ A three level network: documents, terms and queries.
- ✓ $P(q, d_j)$: The probability of a match between a query q and a document d_j (used for *ranking*).



Bayesian Networks for IR *(cont.)*

- ✓ Calculating ranking: $P(q,d_j) = \sum_{\forall \vec{k}} P((q,d_j) \mid k_1,\ldots,k_t) \cdot P(k_1,\ldots,k_t)$

$$= \sum_{\forall \vec{k}} P(q,d_j,k_1,\ldots k_t) =$$

$$= \sum_{\forall \vec{k}} P(q \mid (d_j,k_1,\ldots,k_t)) \cdot P(d_j,k_1,\ldots,k_t)$$

$$= \sum_{\forall \vec{k}} P(q \mid k_1,\ldots,k_t) \cdot P(k_1,\ldots,k_t \mid d_j) \cdot P(d_j)$$

- Arguments applied in this derivation:
  - Basic conditioning: When B_i are disjoint and exhaust all the possibilities then $P(A)=\Sigma\, P(A \mid B_i) \cdot P(B_i)$.
  - Bayes rule (3 times).
  - A node does not depend on a grandparent: $P(q \mid d, k_1, \ldots, k_t) = P(q \mid k_1, \ldots, k_t)$.

Bayesian Networks for IR *(cont.)*

- ✓ Assumption of term independence:

$$P(k_1, \ldots k_t \mid d_j) = \prod_{i \mid k_i = 1} P(k_i \mid d_j) \cdot \prod_{i \mid k_i = 0} (1 - P(k_i \mid d_j))$$

  - The first product is for the terms k_i that appear (1) in $k_1, \ldots, k_t$.
  - The second product is for the terms k_i that do not appear (0) in $k_1, \ldots, k_t$.

- ✓ Altogether,

$$P(q, d_j) = P(d_j) \cdot \sum_{\forall \vec{k}} P(q \mid k_1, \ldots, k_t) \cdot \prod_{i \mid k_i = 1} P(k_i \mid d_j) \cdot \prod_{i \mid k_i = 0} (1 - P(k_i \mid d_j))$$

Bayesian Networks for IR *(cont.)*

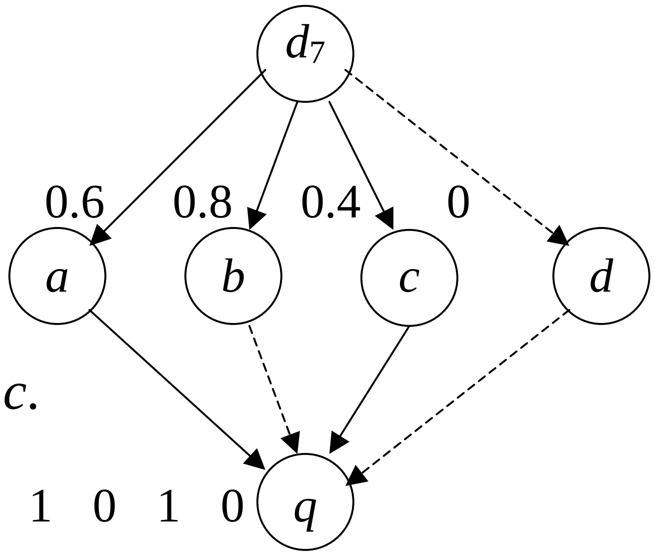
- ✓ We provide
  - The *prior* probability $P(d_j)$
  - The *conditional* probabilities $P(k_i \mid d_j)$
  - The *posterior* probabilities $P(q \mid k_1, \ldots, k_t)$
- ✓ We then derive
  - The final *ranking* $P(q, d_j)$

Bayesian Networks for IR *(cont.)*

- ✓ The *prior* probability $P(d_j)$ is the probability of a document; either
  - *Uniform distribution:* $P(d_j) = 1/n$ (where n is the size of the collection).
  - *Normalized:* $P(d_j) = 1/|d_j|$ (adjust by the *norm*, as in the vector model).
- ✓ The *conditional* probability $P(k_i \mid d_j)$ is the relevance of term k_i to document d_j; either
  - *A binary value*: 1 if k_i appears in d_j, 0 otherwise (as in the Boolean model).
  - *A weight:* based on the term frequency $f_{i,j}$ (as in the vector model).
- ✓ The *posterior* probability $P(q \mid k_1, \ldots, k_t)$ is the relevance of term k_i to query q; either
  - A *binary value*: 1 if the binary vector $k_1, \ldots, k_t$ corresponds exactly to the query terms, 0 otherwise.
  - *A weight*: based on the inverse document frequency idf_i (as in the vector model).

Example

- ✓ We shall consider only the case of *uniform* priors, *weighted* conditionals, and *binary* posteriors.
- ✓ Example
  - A total of 10 documents (n = 10).
  - A total of 4 terms (t = 4): a, b, c, d.
  - A specific document d_7 has these terms
    $P(a \mid d_7) = 0.6$, $P(b \mid d_7) = 0.8$, $P(c \mid d_7) = 0.4$,
    $P(d \mid d_7) = 0$.
  - A (Boolean) query q specifies these terms: a, c.
  - The prior probability is $P(d_7) = 0.1$.
  - The posterior probabilities:
    $P(q \mid (1,0,1,0)) = 1$ (the other 15 posteriors are 0).
  - We can now calculate the ranking $P(q, d_7)$.



Example *(cont.)*

- Example (cont.)
  - The summation is over 16 possible term vectors, but the only vector with a non-zero posterior probability is 1,0,1,0.
  - The contribution of the terms: $0.6 \cdot (1 - 0.8) \cdot 0.4 \cdot (1 - 0) = 0.048$.
    - For desired terms (such as *a* and *c*), the stronger their weight in the document, the higher the final ranking!
    - For undesired terms (such as *b* and *d*), the stronger their weight in the document, the lower the ranking!
  - The final relevance (ranking) of d_7 to q is
    - $P(q, d_7)$. $= 0.1 \cdot 0.048 = 0.0048$.
  - Assume now another document d_8 with term weights *exactly* as given in q: $P(a \mid d_8) = 1$, $P(b \mid d_8) = 0$, $P(c \mid d_8) = 1$, $P(d \mid d_8) = 0$.

    Then the contribution of the terms is maximal: $1 \cdot (1 - 0) \cdot 1 \cdot (1 - 0) = 1$.

    And the final ranking is
    - $P(q, d_8) = 0.1 \cdot 1 = 0.1$.
  - The uniform prior $1/n$ may be ignored as it affects all rankings equally.

Example *(cont.)*

- ✓ Until now we assumed queries are simple conjunctions of terms.
  - In the example, $q = (a \wedge c)$.
- ✓ Assume now queries are *disjunctions* of such conjuncts.
  - For example, $q = (a \wedge c) \vee (a \wedge b)$.
- ✓ The posterior probability $P(q \mid k_1, \ldots, k_t)$ would be defined as 1 for any vector that corresponds to a conjunct, and 0 otherwise.
  - In this example, $P(q \mid (1,0,1,0)) = 1$ and $P(q \mid (1,1,0,0)) = 1$ (the other 14 posteriors are 0).
  - This results in two non-zero components:
    - $0.6 \cdot (1 - 0.8) \cdot 0.4 \cdot (1 - 0) = 0.048$
    - $0.6 \cdot 0.8 \cdot (1 - 0.4) \cdot (1 - 0) = 0.288$
  - And the overall ranking of d_7 with respect to this new query:
    - $P(q, d_7) = 0.1 \cdot (0.048 + 0.288) = 0.0336$