Ch6. Bayesian Learning 6.1 – 6.3.1, 6.7 - 6.11

S. Visa

Basics of probability

- P(T) = probability that event T occurs
- P(T|S) = probability that event T occurs given that S has occurred (conditional probability)
- Rules
  – Complement: P(~A) = 1- P(A)
  – Disjunction: P(A or B) = P(A)+P(B)-P(A and B)
  – Conjunction: P(A and B) = P(A)P(B|A)=P(B)P(A|B)
  – Th. of total probability: $P(B) = \Sigma P(B|A_i)P(A_i)$ where A_i are mutually exclusive and $\Sigma P(A_i)=1$

Introduction

- P(T|S) = probability that event T occurs given that S has occurred

- Example
  - P(U) = probab. of drawing Ace and Queen of Diamonds on 2 draws
  - P(A) = 1/52 → probab. to draw Ace of Diamond
  - P(Q|A) = 1/51 → probab. that Queen of Diamond is drawn, given that Ace of Diamond was drawn already
  - ➔ P(U) = 1/52*1/51= P(A)* P(Q|A)
  - Similarly, P(U) = P(Q)* P(A|Q) = 1/52*1/51
  - Solving eq. P(A)* P(Q|A) = P(Q) *P(A|Q)➔ Bayes Product Th.

- Bayes General Product Theorem

$$P(A \mid Q) = \frac{P(Q \mid A) * P(A)}{P(Q)}$$

- Bayes Theorem in ML

$$P(h \mid D) = \frac{P(D \mid h) * P(h)}{P(D)}$$

PART A

- Find most probable h given D and H

Bayesian learning - MAP

$$P(h \mid D) = \frac{P(D \mid h) * P(h)}{P(D)}$$

- **Posterior probability**
  – = probab. of h given data set D
  – = probab. that h is the sol. for the problem

- Posterior probab. computation uses
  – **Likelihood** of h
    - = probab. that data D occurs if h were the correct hypothesis for the pb.
  – **Prior probability** of h
    - = probab. that h is correct if D is not considered
  – **P(D)**
    - = probab. that data D occurs independently

- Learning criteria: h with largest posterior probability
  ➔ Maximum Aposteriori Hypothesis (MAP)

- Note
  – P(D) is constant ➔ ignore it
  – posterior = likelihood * prior = evidence

Bayesian Learning – ML

- Maximum likelihood (ML) hypothesis
  - = MAP when all h are equally likely
  - ➔ particular case of MAP

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Generally want the most probable hypothesis given the training data
Maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

If assume $P(h_i) = P(h_j)$ then can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

Example: using MAP

- Given: a lab test for cancer gives
  - pos. result 98% of times when cancer is present
    - P(+|cancer) = .98
    - P(-|cancer) = .02
  - neg. result 97% of times if cancer is absent
    - P(+|~cancer) = .03
    - P(-|~cancer) = .97
  - 0.8% of population has cancer (this is the prior)
    - P(cancer) = .008
    - P(~cancer) = .992
    - Most probable h = ~cancer

- **Q: given pos. result of a lab work, the patient is diagnosed cancer or ~cancer?**

- Sol:
  - P(cancer|+) = P(+|cancer)*P(cancer) = .98*.008 = .0078
  - P(~cancer|+) = P(+|~cancer)*P(~cancer) = .03*.992 = .0298
  - ➔ h_{MAP} = ~cancer
  - h_{ML}= ? (Note: can use it only when P(cancer)=P(~cancer))

- Obs: Bayesian inference dep. strongly on the prior probability

Brute force MAP/ML hypothesis learner

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

- Obs:
  - Not practical for large H
  - Standard for evaluating or justifying other learning algorithms

PART B

- Find most probable **classification** for x given D and H

Bayes optimal classifier

- Pb. considered until now
  - Given D and H, find most probable h
- Now consider this pb:
  - Given D, H, and a new instance x, what is most probable classification of x?
  - Answer: most probable classif. = h(x), where h is most probable for D (MAP)
  - **NO!!!**
  - **BEST classifier takes advantage of ALL h!**
  - Ex.
    - 3 possible h: $P(h_1|D)=.4$, $P(h_2|D)=P(h_3|D)=.3$ ➔ **h_1 is MAP**
    - Assume for x: $h_1(x)=+$, $h_2(x)=h_3(x)=-$
    - What is most probable classification of x?

Bayes optimal classifier

- Most probable classif. of x is obt. by comb. the predictions of all h, weighted by their posteriors (see formula, where V is the set of classes)

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D)$$

- In our ex:

$$P(h_1|D) = .4,\ P(-|h_1) = 0,\ P(+|h_1) = 1$$
$$P(h_2|D) = .3,\ P(-|h_2) = 1,\ P(+|h_2) = 0$$
$$P(h_3|D) = .3,\ P(-|h_3) = 1,\ P(+|h_3) = 0$$

therefore

$$\sum_{h_i \in H} P(+|h_i) P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i) P(h_i|D) = .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D) = -$$

- OBS. MAP h_1 predicts + but Bayes optimal h (which uses theorem of total probability) predicts -

OBS. on Bayes optimal classifier

- NO classifier based on D and H can exceed the performance of the Bayes optimal classifier

- Bayes optimal is optimal but expensive → uses all h in H

- Drawback – choose an H s.t. one CAN compute all posteriors $P(h_i|D)$ ➔ this might reduce choice of H so severely that other techniques using more powerful H can do better job

- Even so, offers a performance target for all Bayesian classifiers

Gibbs classifier (‘91)

- Randomly selects an h according to posteriors P(h|D) (in practice use P(h))
- Prediction made by h(x)

- Surprisingly: $E(err_{Gibbs}) <= 2*E(err_{BayesOptimal})$
- Apply Gibbs to VS with uniform distribution
  - Pick any h
  - Expected err is no worse than twice Bayes Optimal
  - Not very good but you get a prediction model at 0 cost

Naïve Bayes classifier

- Simple, v. practical, widely used
  - Diagnosis
  - Text documents classification
- Based on Bayes rule + assumption of conditional independence
  - Assumption often violated in practice
  - Even, then, it usually works well
- Applies to learn MAP h for conj. of discrete attributes

Classification using Bayes rule

- Given attribute values, what is most probable class (value) of target variable?

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

Bayes rule

$$\begin{aligned} v_{MAP} &= \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\ &= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j) \end{aligned}$$

- Pb.: large data set needed to estimate $P(a_1 \ldots a_n | v_j)$

Naïve Bayes classifier

- Naïve Bayes assumption: attributes are independent, given the class
  - ➔ $P(a_1 \ldots a_n|v_j)=P(a_1|v_j)\ P(a_2|v_j)\ \ldots\ P(a_n|v_j)$
- Under this assumption ➔ v_{MAP} is:

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

Naive Bayes algorithm

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Naive Bayes: estimation

- Estimate probability from sample proportion
  - $P(v) = count(v)/N$
  - $P(A|B)= count(A \text{ and } B)/count(B)$

- Ex.: N = 100 with 70+ and 30-
  - $P(+)=0.7$ and $P(-)=0.3$
  - Among 70 pos. ex., 35 with a_1=SUNNY ➔ $P(a_1=SUNNY|+)=0.5$

Training examples for PlayTennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naïve Bayes: example

- Consider new instance
  <Outlook=sun, Temp=cool, Humid=high, Wind=strong>
- Use NB to classify it: 'yes' or 'no'?
- Compute $v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$
- P(yes)=?, P(no)=?
- P(sun|yes)=? P(cool|yes)=? P(high|yes)=? P(strong|yes)=?
- P(sun|no)=? …

Naïve Bayes: example

Consider *PlayTennis* again, and new instance

$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong$

Want to compute:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$

$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$

$$\rightarrow v_{NB} = n$$

Naïve Bayes: subtleties

- Estimating probabilities is the major challenge
- Conditional independence assumption is often violated
- …but it works surprisingly well anyway
- What if attribute a_i never observed for class v_j (due to small tr. set)?
  - ➔ estimate $P(a_i|v_j)$ as 0 because count(a_i and v_j)=0
  - Effect too strong ➔ gives 0 to candidacy of v_j
  - Sol.: use m-estimate smoothing

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

Ex.: m-estimate smoothing

- 70+, 30-
- P(a1=s|+) = 0/70 = 0
- Using m-estimate ➔ P(a1=s|+) = $\frac{0+10*\frac{1}{3}}{70+10}=0.04$
  - 10 = no of virtual ex.
  - 1/3 = there are 3 possible values for a1 having uniform distribution

Ex.: m-estimate smoothing

- P(a1=s|+) = .31 (is TRUE probability; ~ 2 out of 6 +ves have a1=s)
- Assume that in tr. data only 1 ex. (out of 6 ex.) in +ve class has a1=s
- ➔ estimate of P(a1=s|+) from tr. data is 1/6=.17 (instead of .31!!!!!)

- To deal with distortion of probab. when dealing with small tr. sets → use m-estimate
- ➔ P(a1=s|+) = $\dfrac{1+50*\frac{1}{3}}{6+50} = 0.294$
  - 50 = no of virtual ex.
  - 1/3 = there are 3 possible values for a1 having uniform distribution

Obs. on m-estimate smoothing

- In previous ex., one can use more than 50 virtual ex. to get even closer to .31
- BUT, actual probab. value (here .31) is unknown!
- M-estimate only improves the **estimate of an unknown probability when dealing with small data sample**
- m = 0 ➔ m-estimate = n_c/n (=1/6 → original estimate!)
- m→∞ ➔ m-estimate = p (=1/3 → prior estimate P(a1=s|+))
- Instead of using formula → pick a value for P(a1=s|+) from intervals $[n_c/n,p)$ or $(p,n_c/n]$ (whichever is non-empty)
- With no additional info → pick $(n_c/n + p)/2)$ as compromise between observed probab. and assumed prior probab.

Naïve Bayes classifier for text

- Ex.
  - Learn which new articles are of interest
  - Learn to classify web pages by topic

- NB works well
  - How to apply NB?
  - How do we represent ex.?
  - What are the attributes?

Representation for text classification

- Attributes = word positions
  - i.e. attribute i = i-th word in text
  - Values for attribute = word that occurs there
  - doc=(a_1=w_1,…,a_n=w_n)
  - Can chose other repres.: attr=specific word, value=its freq. in text

- Assumption: probab. of having a specific word is independent of position
  - $P(a_i=w_k|v_j)=P(a_m=w_k|v_j)=P(w_k|v_j)$
  - $P(doc|v_j)=P(a_1=w_1, a_2=w_2,\dots, a_n=w_n|v_j)=$

    $=P(w_1|v_j)^{freq(w1)} \dots P(w_n|v_j)^{freq(wn)}$

Twenty newsgroups (Jochims' 96)

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

- 20 classes
- 1000 docs for each class
- 2/3 → training; 1/3 → test
- Used 100 most frequent words
- Remove
  - the, and, of, ...
  - any word occurring fewer than 3 times
- Resulting vocabulary ~ 38,500 words
- Random guessing → 5% accuracy

Algorithm

LEARN_NAIVE_BAYES_TEXT(*Examples*, V)

1. collect all words and other tokens that occur in Examples

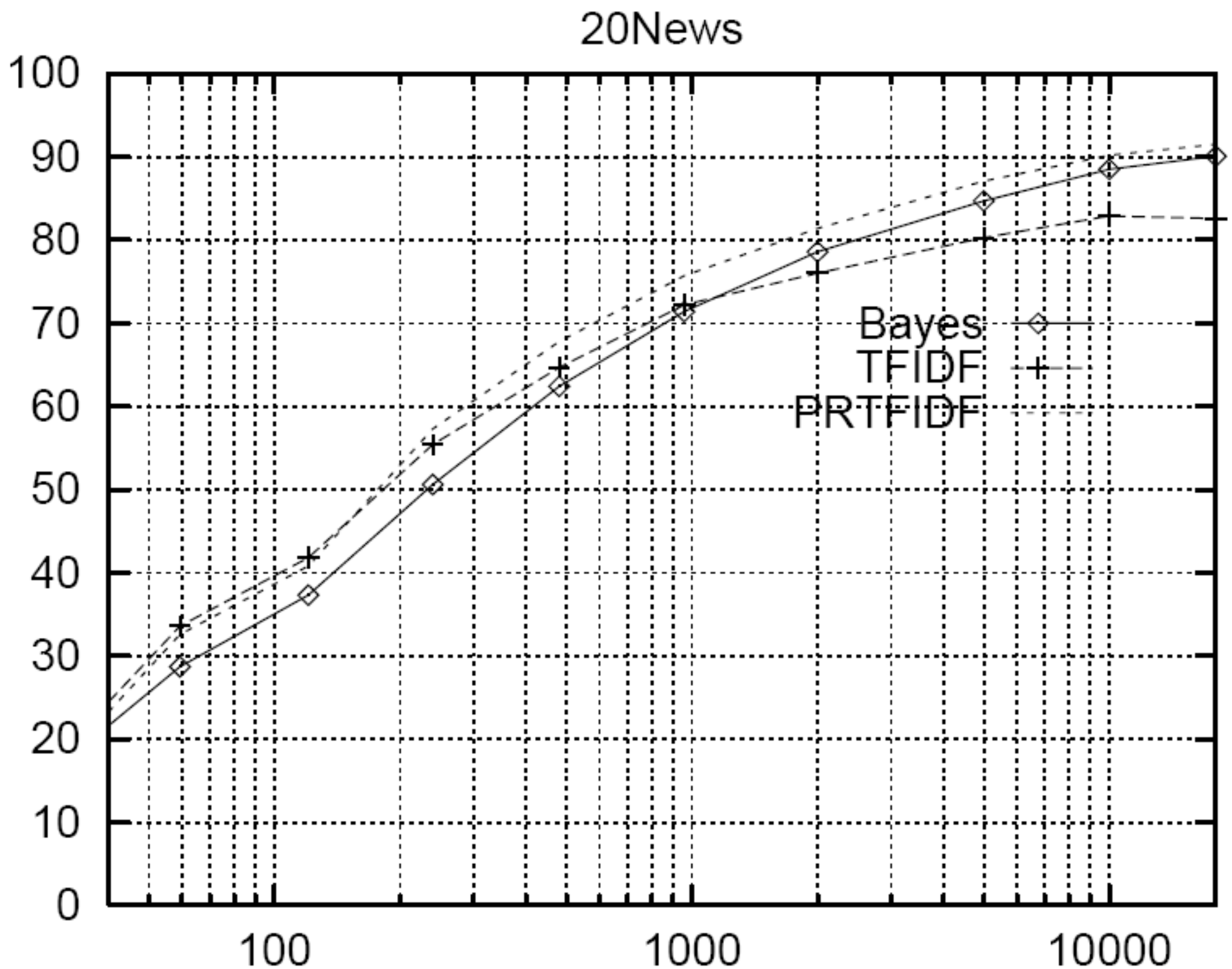
- $Vocabulary \leftarrow$ all distinct words and other tokens in $Examples$

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
  - for each word w_k in $Vocabulary$
    * $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    * $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Learning curve for 20 newsgroups



Accuracy vs. Training set size (1/3 withheld for test)

Bayesian Belief Networks

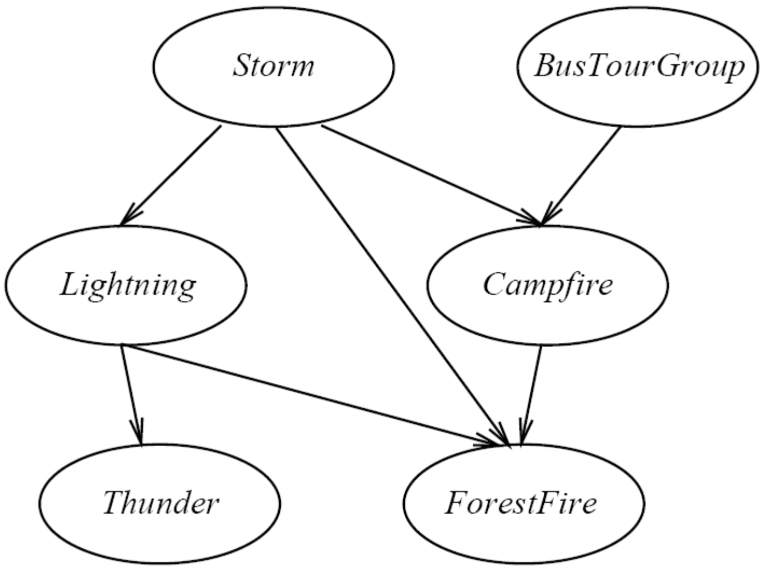
- Consider two extremes
  - Bayes Optimal Classifier – get correct joint probability distribution
    - ➔ optimal classifier
    - But infeasible in practice (too much data needed)
  - Naïve Bayes
    - Much more feasible
    - But strong (and restrictive) assumption of cond. independence

- Something in between?
  - = make some independence assumptions but only where reasonable?
  - ➔ BBN describe conditional independence among subsets of variables
  - BBN is a compromise between BOC and NB

Bayesian Belief Networks

- Def. BBN is directed acyclic graph (nodes + arcs) + conditional probability table for each node
- Represent the joint probability distribution of the variables (=all cond. probab. among variables)
- Use the concept of **conditional independence**
  - $P(A1|A2,V) = P(A1|V)$
    - **A1 and A2 are conditional independent given V**
    - = even though A1 and A2 may influence each other, the fact that V is true, completely explains that
    - E.g. Campfire is cond. indep. of Lightning given Storm

Bayesian Belief Networks

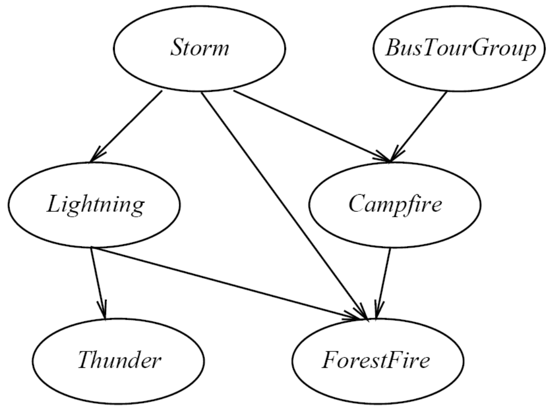


	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

Network represents a set of conditional independence assertions:

- Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors.
- Directed acyclic graph

Cond. indep. and joint probab.



	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

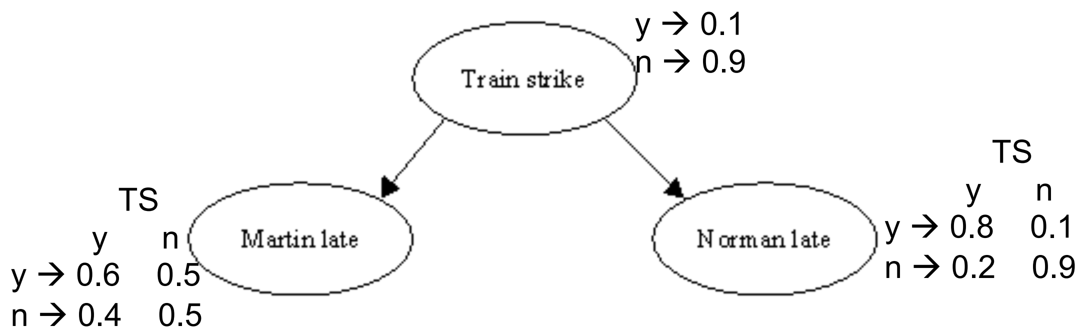
Campfire

- Node v is **cond. indep.** of node na (not an ancestor of v) given its immediate ancestors a1,…,an
  - P(v|na,a1,…,an)=P(v|a1,…,an)
  - P(ForestFire|Thunder, Storm, Lightening, Campfire) = P(ForestFire|Storm, Lightening, Campfire)

- Chain rule of probability describes the joint probability of a set of variables
  - $P(x_1,\dots,x_n) = \prod_i P(x_i|x_1,\dots,x_{i-1})$
  - $P(x_1,x_2,x_3) = P(x_1)\, P(x_2|x_1)\, P(x_3|x_1,x_2)$

- In BBN - probab. of immediate ancestors of node xi completely det. the **joint probab**. distrib. for xi
  - $P(x1,\dots,xn) = \prod_i P(xi|parents(xi))$
  - Ex. P(S,B,L,C,T,F) = ?

BBN example



- A) Compute unconditional (marginal) probability
  - P(NL=y) = P(NL=y | TS=y) * P(TS=y) + P(NL=y | TS=n) * P(TS=n) = 0.17
  - P(ML=y) = ? (0.51)

- B) Revising probabilities when propagating evidence
  - We know **TS = y**
    - P(NL=y) = P(NL=y | TS=y) * P(TS=y) + 0 = 0.8 * 1 + 0 = 0.8
    - P(ML=y) = ? (0.6)

  - We know **NL = y**
    - P(TS=y) = ?
      - = P(TS=y | NL=y) = [P(NL=y | TS=y) * P(TS=y)] / P(NL=y) = 0.8 * 0.1 / 0.17 = 0.47
      - Obs. The evidence NL=y increased the probab. that TS=y!!!
    - P(ML=y) = ?
      - = P(ML=y | TS=y) * P(TS=y) + P(ML=y | TS=n) * P(TS=n) = 0.6 * 0.47 + 0.5 *0.53 = 0.55
      - Obs. The evidence NL=y propagated to ML and slightly increased the probab. that ML=y!!!