Learning belief networks for classification

- Use a belief network to infer the probability distribution of some class variable
  - specifies the probability the class variable will take on each of its possible values given available evidence
- Example
  - P(Tennis = “yes” | evidence)
    P(Tennis = “no” | evidence)

Learning belief networks for classification

Versions (simplest to hardest)

1. given data & naïve Bayes (fixed structure) of network,
   learn: probabilities
2. given data & structure of the network (we come up with it),
   learn: probabilities
3. given data & node for each attribute and class variable
   learn: arcs and probabilities
4. given data & node for each attribute and class variable
   learn: hidden nodes, arcs, and probabilities

Jeeves is a valet to Bertie Wooster. On some days, Bertie likes to play tennis and asks Jeeves to lay out his tennis things and book the court. Jeeves would like to be able to predict whether Bertie will play tennis (and so be a better valet). Each morning over the last two weeks, Jeeves has recorded whether Bertie played tennis on that day and various attributes of the weather.

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Can Jeeves learn to predict Bertie's tennis playing?

Consider an accurate network for Jeeves data

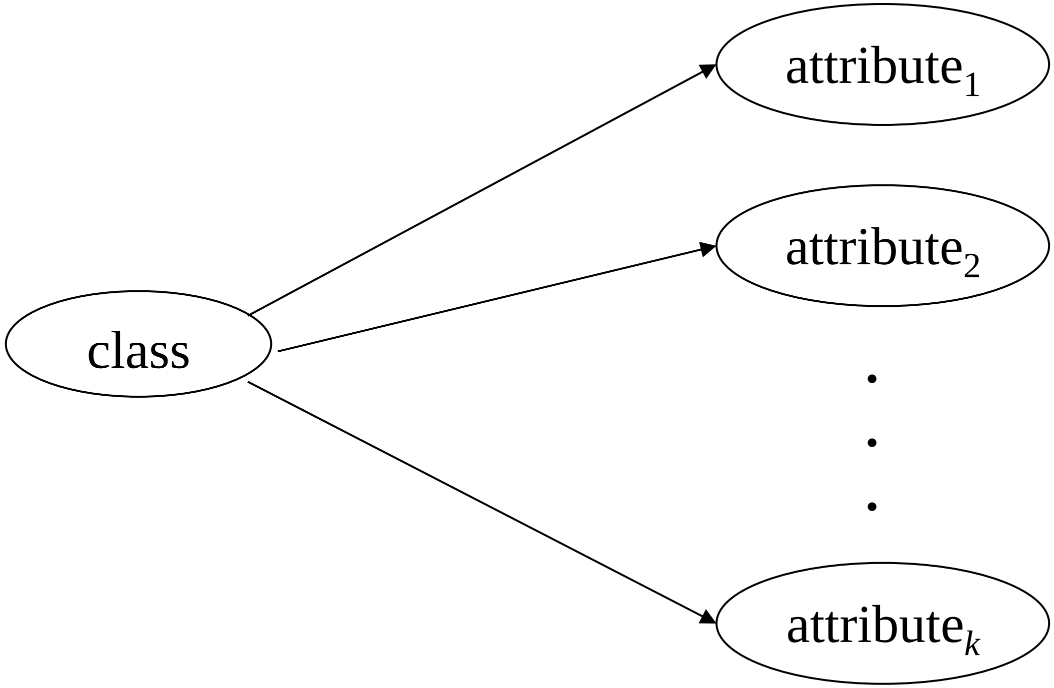
Recall ...

$$P(X_1, \ldots, X_n) = \prod P(X_i \mid \text{Parents}(X_i)) \qquad (1)$$

$$P(X_1, \ldots, X_n) = \prod P(X_i \mid X_{i-1}, \ldots, X_1) \qquad (2)$$

- Equation 1: formula for network
  Equation 2: chain rule (always true)
- Equation 1 is a <u>correct</u> representation of a domain only if each node is conditionally independent of its predecessors (in the node ordering), given its parents

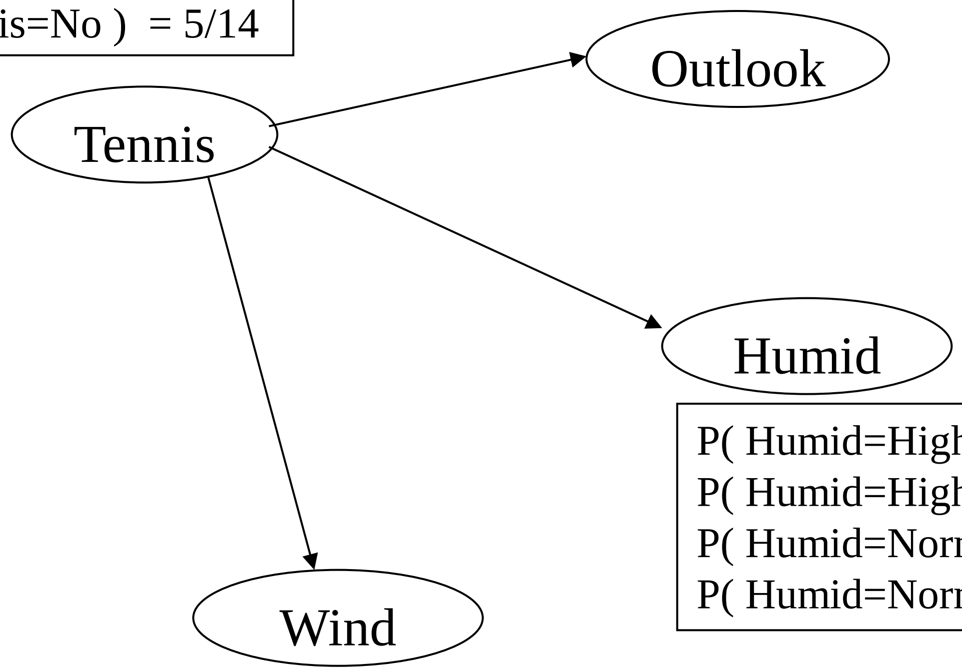
Naïve Bayes classifier



Jeeves' naïve Bayesian network classifier

P(Tennis=Yes) = 9/14
P(Tennis=No) = 5/14

P(Outlook=Sunny | Tennis=Yes) = 2/9
P(Outlook=Sunny | Tennis=No) = 3/5
P(Outlook=Overcast | Tennis=Yes) = 4/9
P(Outlook=Overcast | Tennis=No) = 0
P(Outlook=Rain | Tennis=Yes) = 3/9
P(Outlook=Rain | Tennis=No) = 2/5



P(Humid=High | Tennis=Yes) = 3/9
P(Humid=High | Tennis=No) = 4/5
P(Humid=Normal | Tennis=Yes) = 6/9
P(Humid=Normal | Tennis=No) = 1/5

P(Wind=Weak | Tennis=Yes) = 6/9
P(Wind=Weak | Tennis=No) = 2/5
P(Wind=Strong | Tennis=Yes) = 3/9
P(Wind=Strong | Tennis=No) = 3/5

Is this network a good model?

- Consider the conditional independence assumptions implict in the naïve Bayes classifier
- Using Equation (1)

  P(Tennis, Outlook, Humid, Wind)
  = P(Tennis) P(Outlook | Tennis)
  P(Humid | Tennis)
  P(Wind | Tennis)

- Using Equation (2)

  P(Tennis, Outlook, Humid, Wind)
  = P(Tennis) P(Outlook | Tennis)
  P(Humid | Tennis, Outlook)
  P(Wind | Tennis, Outlook, Humid)

Is this network a good model?

- Are these valid conditional independence assumptions?

  P(Humid | Tennis) = P(Humid | Tennis, Outlook) ?

  P(Wind | Tennis) = P(Wind | Tennis, Outlook, Humid) ?

- Using our knowledge of weather? No
- Consider, empirically, one example

  P(Humid = High | Tennis = Yes) = 3/9

vs.

P(Humid = High | Tennis = Yes, Outlook = Sunny) = 0/2

P(Humid = High | Tennis = Yes, Outlook = Overcast) = 2/4

P(Humid = High | Tennis = Yes, Outlook = Rain) = 1/3

Assumption appears not to hold empirically (but: would need more data and would decide this using a statistical test)

Jeeves would like to evaluate the classifier he has come up with for predicting whether Bertie will play tennis. Each morning over the next two weeks, Jeeves records the following data.

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Mild	High	Strong	No
2	Rain	Hot	Normal	Strong	No
3	Rain	Cool	High	Strong	No
4	Overcast	Hot	High	Strong	Yes
5	Overcast	Cool	Normal	Weak	Yes
6	Rain	Hot	High	Weak	Yes
7	Overcast	Mild	Normal	Weak	Yes
8	Overcast	Cool	High	Weak	Yes
9	Rain	Cool	High	Weak	Yes
10	Rain	Mild	Normal	Strong	No
11	Overcast	Mild	High	Weak	Yes
12	Sunny	Mild	Normal	Weak	Yes
13	Sunny	Cool	High	Strong	No
14	Sunny	Cool	High	Weak	No

How well does Jeeves predict Bertie's tennis playing?

Example

- Suppose:

  Outlook = Sunny, Humidity = High, Wind = Strong

  Tennis?

P(Tennis = Yes) P(Outlook = Sunny | Tennis = Yes)
P(Humidity = High | Tennis = Yes)
P(Wind = Strong | Tennis = Yes)

= (9/14)(2/9)(3/9)(3/9) = 0.01587

P(Tennis = No) P(Outlook = Sunny | Tennis = No)
P(Humidity = High | Tennis = No)
P(Wind = Strong | Tennis = No)

= (5/14)(3/5)(4/5)(3/5) = 0.1029

Jeeves would like to evaluate the classifier he has come up with for predicting whether Bertie will play tennis. Each morning over the next two weeks, Jeeves records the following data.

Day	Outlook	Humidity	Wind	Tennis?	Prediction
1	Sunny	High	Strong	No	No
2	Rain	Normal	Strong	No	**Yes**
3	Rain	High	Strong	No	No
4	Overcast	High	Strong	Yes	Yes
5	Overcast	Normal	Weak	Yes	Yes
6	Rain	High	Weak	Yes	Yes
7	Overcast	Normal	Weak	Yes	Yes
8	Overcast	High	Weak	Yes	Yes
9	Rain	High	Weak	Yes	Yes
10	Rain	Normal	Strong	No	**Yes**
11	Overcast	High	Weak	Yes	Yes
12	Sunny	Normal	Weak	Yes	Yes
13	Sunny	High	Strong	No	No
14	Sunny	High	Weak	No	No

How well does Jeeves predict Bertie's tennis playing?

Handling missing values

- Suppose:

  Outlook = ?, Humidity = High, Wind = Strong

  Tennis?

  Just omit the attribute Outlook in calculation

  P(Tennis = Yes) P(Humidity = High | Tennis = Yes)
  P(Wind = Strong | Tennis = Yes)

  = (9/14)(3/9)(3/9) = 0.071

  P(Tennis = No) P(Humidity = High | Tennis = No)
  P(Wind = Strong | Tennis = No)

  = (5/14)(4/5)(3/5) = 0.171

Learning probabilities

- We estimated, for example,

$$P(\text{Outlook} = \text{Sunny} \mid \text{Tennis} = \text{Yes}) = \frac{n_a}{n} = \frac{2}{9}$$

where $n = 9$ is number of instances for which Tennis=Yes

$n_a = 2$ is number of instances for which Outlook=Sunny using just the instances where Tennis=Yes

- This will be a poor estimate when n_a is small or zero

Common solutions

- *m*-estimate of probability

$$\frac{n_a + mp}{n + m}$$

  p = prior estimate of probability

  m = "equivalent sample size weighting"

- E.g., for P(Outlook = Overcast | Tennis = No)

  Assume *p* = 1/3 (equally likely to be Overcast, Sunny, Rain)