Naive Bayesian Classifier (20.2.2)

Given a vector V = {v11, v22, v31, ..., vm2, c?}
Compute:

P(c1| v11, v22, v31,…, vm2) proportional to

P(v11|v22, v31, …, vm2,c1)P(v22|v31,…,vm2,c1)….P(vm2|c1)P(c1)

equals (under assumption that Vi's are independent conditioned on C)

P(v11|c1)P(v22|c1)P(31|c1)….P(vm2|c1)P(c1)

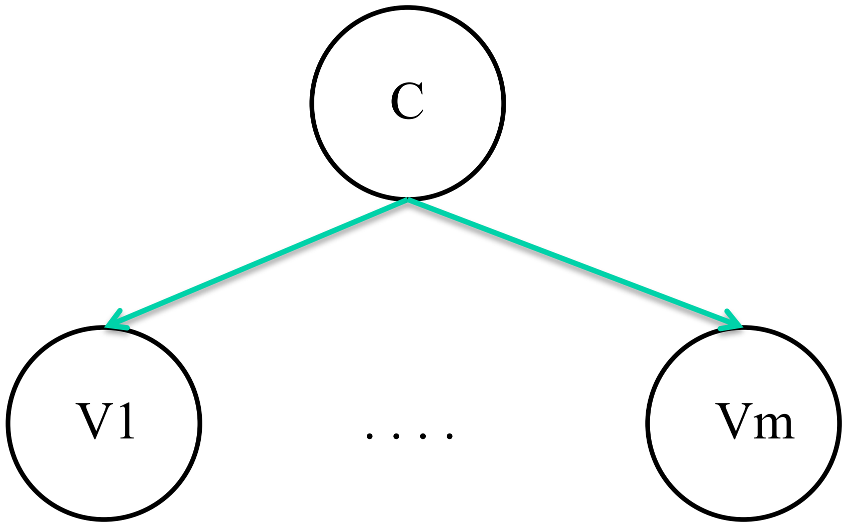
P(c2| v11, v22, v31,…, vm2) proportional to

P(v11|v22, v31, …, vm2,c2)P(v22|v31,…,vm2,c2)….P(vm2|c2)P(c2)

equals (under assumption that Vi's are independent conditioned on C)

P(v11|c2)P(v22|c2)P(31|c2)….P(vm2|c2)P(c2)

Classify V as in c1 or c2, whichever yields higher probability



P(C| V1, V2, V3,…, Vm) proportional to

P(V1, V2, V3, Vm, C) =

P(V1|C)P(V2|C)P(V3|C)….P(Vm|C)P(C)

Given a vector $V = \{1, \quad -1, \quad 0, \quad \ldots., \quad 1\}$

Compute:

P(-1| 1, -1, 0,…, 1) as P(1|-1)P(-1|-1)P(0|-1)….P(1|-1)P(-1)

P(1| 1, -1, 0,…, 1) as P(1|1)P(-1|1)P(0|1)….P(1|1)P(1)

Classify V as in c1 or c2, whichever yields higher probability

Learning a Naïve Bayesian Classifier.

View probabilities as proportions computed over training set.

P(v11|c1)P(v22|c1)P(31|c1)….P(vm2|c1)P(c1)

= [v11,c1]/[c1] * [v22,c1]/[c1] * [v31,c1]/[c1] *…* [vm2,c1]/[c1] * [c1]/[]

where [conditions] is the number of objects/rows in the training set that satisfy all the conditions. So [v11,c1] is the number of training data that are members of c1 and have V1=v11, [c1] is the number of training objects in c1, [] is the total number of training objects.

Learning in this case, is a matter of counting the number of rows in training data in which various conditions satisfied. What conditions? Each class/variable-value pair, each class, total number of rows.

$$
\begin{array}{ccc}
 & \begin{array}{ccccc} V1 & V2 & V3 & \dots\dots\dots & Vm \end{array} & \\
c1 & \left(\begin{array}{ccccc} [v11,c1] & [v21,c1] & [v31,c1] & \dots\dots\dots & [vm1,c1] \\ [v12,c1] & [v22,c1] & [v32,c1] & \dots\dots\dots & [vm2,c1] \end{array}\right) & [c1] \\
 & & \\
c2 & \left(\begin{array}{ccccc} [v11,c2] & [v21,c2] & [v31,c2] & \dots\dots\dots & [vm1,c2] \\ [v12,c2] & [v22,c2] & [v32,c2] & \dots\dots\dots & [vm2,c2] \end{array}\right) & [c2] \\
 & & \\
 & \begin{array}{ccccc} [v11] & [v21] & [v31] & & [vm1] \\ [v12] & [v22] & [v32] & & [vm2] \end{array} & [\ ]
\end{array}
$$

Consider an (multidimensional) array implementation of int, and estimate P(vij|ck) as ([vij,ck]+1) / ([ck]+2), and P(ck) as ([ck]+1) / ([]+2)

Number of classes

Number of Vi values

“various tricks are used” to avoid probabilities of 0

Unsupervised Performance Task: Pattern Completion

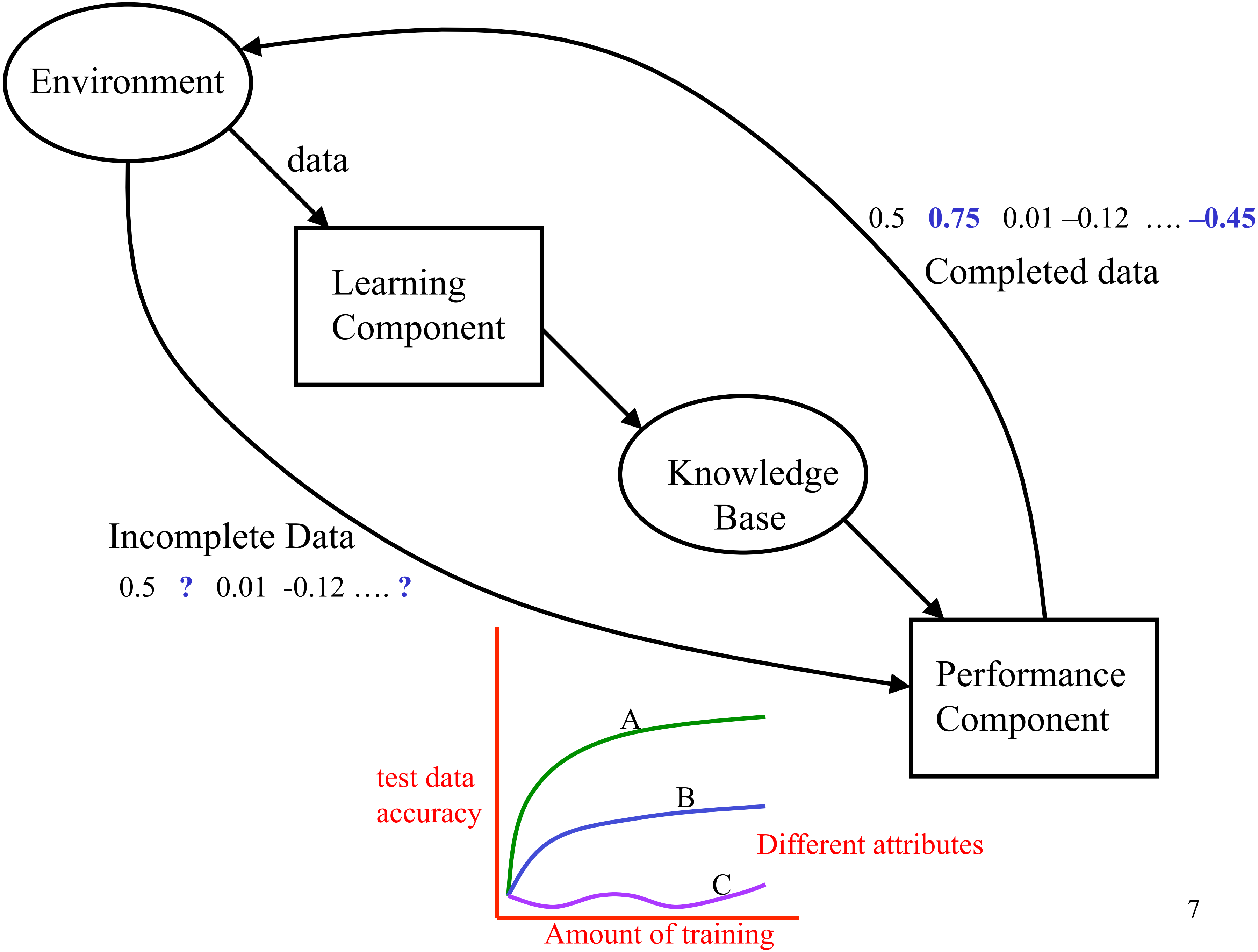
0.5 **?** 0.01 -0.12 …. **?** ~~C’ C”~~

↓

Knowledge
Base

↓

0.5 **0.75** 0.01 –0.12 …. **–0.45** ~~c’1 c”2~~


Environment
data
Learning Component
Knowledge Base
Performance Component
0.5 0.75 0.01 –0.12 …. –0.45
Completed data
Incomplete Data
0.5 ? 0.01 -0.12 …. ?
test data accuracy
A
B
C
Different attributes
Amount of training

Example: Unsupervised rule induction of Association Rules (market-basket analysis)

In a nutshell: run "brute force" rule discovery for all possible consequents, not simply single variable values (e.g., V1=v12), but consequents that are conjunctions of variable values (e.g., V1=v12 & V4=v42 & V5=v51).

Retain rules $A \rightarrow C$ such that $\underbrace{P(A \,\&\, C)}_{\textit{support}} \geq T1$ and $\underbrace{P(C|A)}_{\textit{confidence}} \geq T2$. These thresholds enable pruning of the search space (A and C are themselves conjunctions).

Problem: a plethora of rules, most uninteresting, are produced.

Solutions: Organize/prune rules by

a) Interestingness (e.g., $A \rightarrow C$ interesting if $P(A, C) \gg P(A)P(C)$ or $\ll P(A)P(C)$)

b) confidence (a confidence interval around coverage and/or accuracy)

c) support for top-level goal

A Priori Algorithm

Prune conjunctions of features that fall below threshold support

T1 = 20%

[ ]

[soap] 10%   [towels] 5%   [soda] 25%   milk 30%   [bread] 40%

A Priori Algorithm

Prune conjunctions of features that fall below threshold support

T1 = 20%

[ ]

~~[soap] 10%~~ ~~[towels] 5%~~ [soda] 25% milk 30% [bread] 40%

soda, milk 18% soda, bread 14% milk, bread 22%

A Priori Algorithm

Prune conjunctions of features that fall below threshold support

T1 = 20%

[ ]

~~[soap] 10%~~ ~~[towels] 5%~~ [soda] 25% milk 30% [bread] 40% . . .

~~soda, milk 18%~~ ~~soda, bread 14%~~ milk, bread 22%

A Priori Algorithm

Prune conjunctions of features that fall below threshold support

T1 = 20%

[ ]

~~[soap] 10%~~ ~~[towels] 5%~~ [soda] 25% milk 30% [bread] 40% . . .

~~soda, milk 18%~~ ~~soda, bread 14%~~ milk, bread 22% . . .

milk, bread, PB 20% milk, bread, spam 5%

A Priori Algorithm

Prune conjunctions of features that fall below threshold support

T1 = 20%

[ ]

~~[soap] 10%~~ ~~[towels] 5%~~ [soda] 25% milk 30% [bread] 40% . . . PB

~~soda, milk 18%~~ ~~soda, bread 14%~~ milk, bread 22% . . .

milk, bread, PB 20% ~~milk, bread, spam 5%~~

And search for implication rules within each frequent item set

e.g., milk → bread P(bread | milk), and bread → milk P(milk | bread)

[bread, milk] ; bread → milk (40%, 80%)

[wheat medium bread, low fat milk]; wheat medium bread → low fat milk (16%, 85%)

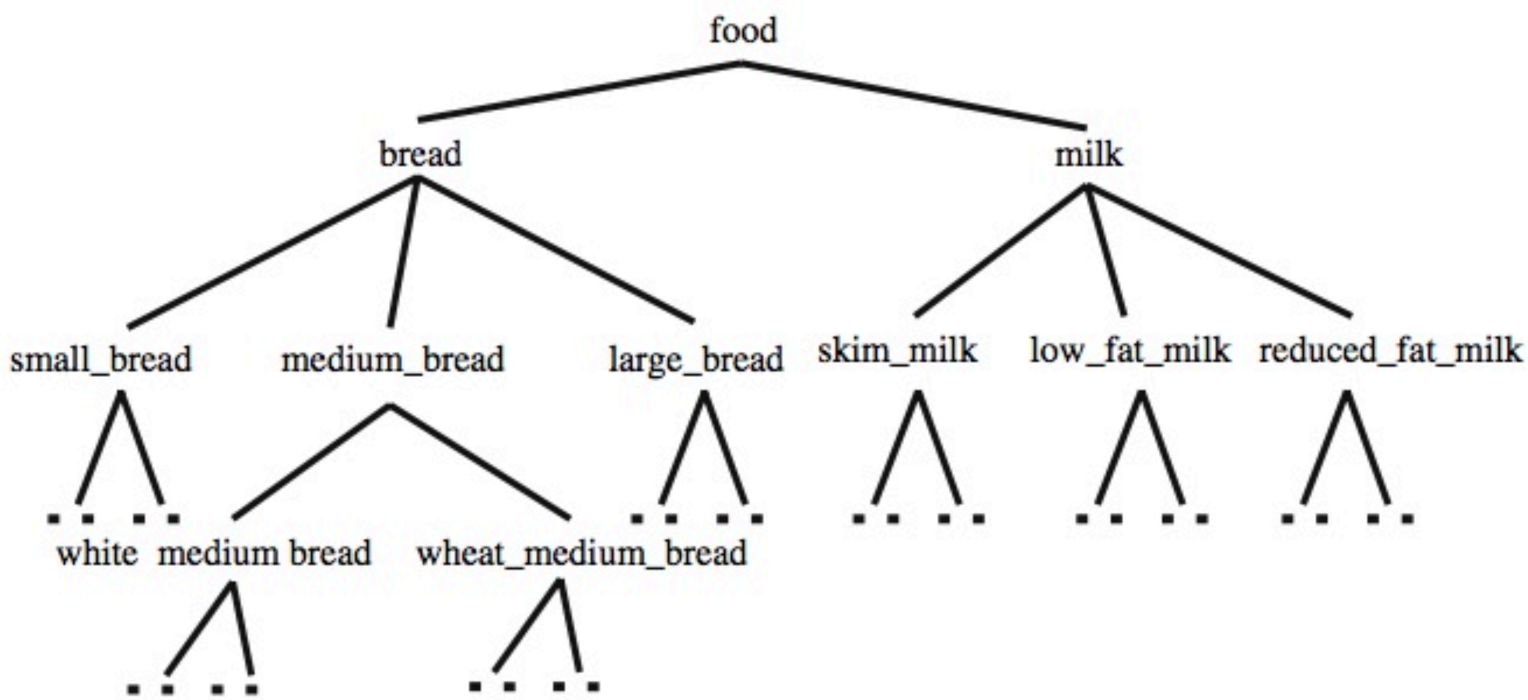


Figure 10.11. Example of a concept hierarchy concerning food.

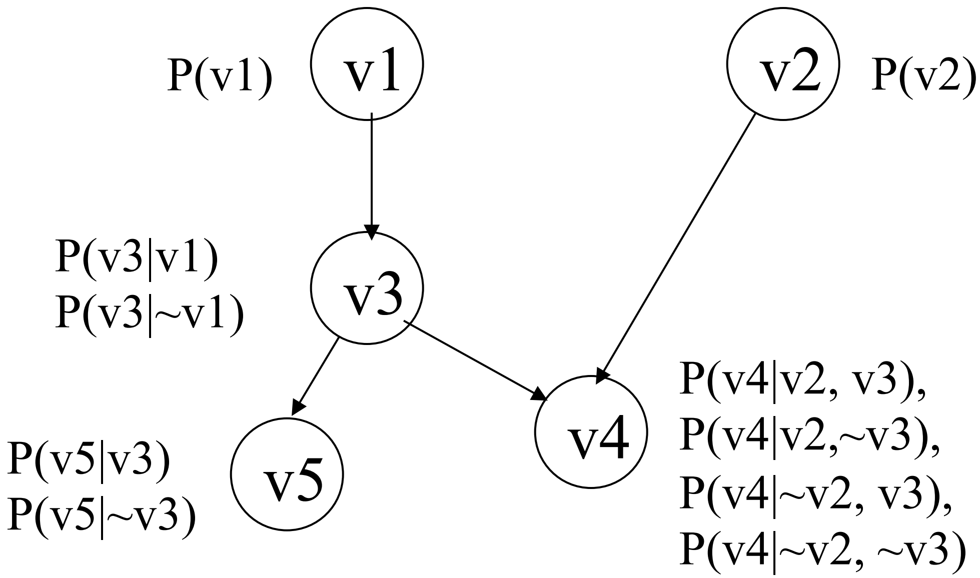
Pairing weak supervision with unsupervised learning to derive a characteristic learner

e.g., Biswas and Kinnebrew

Separate high learners with low learners

Learn association rules for each group

Example (Empirical, Unsupervised): Learning Bayesian Networks



Components of a Bayesian Network: a topology (graph) that qualitatively indicates displays the conditional independencies, and probability tables at each node

Semantics of graphical component: for each variable, v, v is independent of all of its non-descendents conditioned on its parents

A Bayesian Network is a graphical representation of a joint probability distribution with (conditional) independence relationships made explicit

Recall the chain rule:

Assume Vi a binary valued variable (T or F)

P(v1 and v2 and ~v3 and v4 and ~v5)

A factorization ordering

$$= \underbrace{\underbrace{\underbrace{\underbrace{P(v1)P(v2|v1)}_{P(v1,v2)}P(\sim v3|v1,v2)}_{P(v1,v2,\sim v3)}P(v4|v1,v2,\sim v3)}_{P(v1,v2,\sim v3,v4)}P(\sim v5|v1,v2,\sim v3,v4)}_{P(v1,v2,\sim v3,v4,\sim v5)}$$

P(v1 and v2 and ~v3 and v4 and ~v5)

An alternative ordering

$$= P(v4)P(v2|v4)P(\sim v3|v4,v2)P(v1|v4,v2,\sim v3)P(\sim v5|v4,v2,\sim v3,v1)$$

$P(v1 \text{ and } v2 \text{ and } \sim v3 \text{ and } v4 \text{ and } \sim v5)$

A factorization ordering

$= P(v1)P(v2|v1)P(\sim v3|v1,v2)P(v4|v1,v2,\sim v3)P(\sim v5|v1,v2,\sim v3,v4)$

Assume the following conditional independencies:

$P(v1)$

v2 independent of v1

$P(v2|v1) = P(v2)$ and $P(v2|\sim v1) = P(v2)$, $P(\sim v2|v1) = P(\sim v2)$, $P(\sim v2|\sim v1) = P(\sim v2)$

$P(\sim v3|v1,v2) = P(\sim v3|v1)$

v3 independent of v2 conditioned on v1

and $P(\sim v3|v1,\sim v2) = P(\sim v3|v1)$, $P(\sim v3|\sim v1,v2) = P(\sim v3|\sim v1)$, $P(\sim v3|\sim v1,\sim v2) = P(\sim v3|\sim v1)$,
$P(v3|v1,v2) = P(v3|v1)$, $P(v3|v1,\sim v2) = P(v3|v1)$, $P(v3|\sim v1,v2) = P(v3|\sim v1)$,
$P(v3|\sim v1,\sim v2) = P(v3|\sim v1)$

$P(v4|v1,v2,\sim v3) = P(v4|v2, \sim v3)$ and ……

$P(\sim v5|v1,v2,\sim v3,v4) = P(\sim v5|\sim v3)$ and …..

P(v1 and v2 and ~v3 and v4 and ~v5)

= P(v1)P(v2|v1)P(~v3|v1,v2)P(v4|v1,v2,~v3)P(~v5|v1,v2,~v3,v4)

= P(v1)P(v2)P(~v3|v1)P(v4|v2,~v3)P(~v5|~v3)

How many probabilities need be stored?

P(v1), P(~v1) 2 probabilities (actually only one, since P(~v1) = 1 – P(v1))

2 probabilities (or 1) instead of 4 (or 2)

P(v2|v1) = P(v2) and P(v2|~v1) = P(v2), P(~v2|v1) = P(~v2), P(~v2|~v1) = P(~v2)

P(~v3|v1) = 1 – P(v3|v1)

P(~v3|v1,v2) = P(~v3|v1) 4 probabilities (or 2) instead of 8 (or 4)

and P(~v3|v1,~v2) = P(~v3|v1), P(~v3|~v1,v2) = P(~v3|~v1), P(~v3|~v1,~v2) = P(~v3|~v1),
P(v3|v1,v2) = P(v3|v1), P(v3|v1,~v2) = P(v3|v1), P(v3|~v1,v2) = P(v3|~v1),
P(v3|~v1,~v2) = P(v3|~v1)

8 probabilities (or 4) instead of 16 (or 8)

P(v4|v1,v2,~v3) = P(v4|v2, ~v3) and ……

4 probabilities (or 2) instead of 32 (or 16)

P(~v5|v1,v2,~v3,v4) = P(~v5|~v3) and …..

For a *particular factorization ordering*, construct a Bayesian network as follows:

v1 a "root"

P(v1), P(~v1)   v1   P(v1) = 0.75

P(~v1) = 0.25 = 1 – P(v1)

---

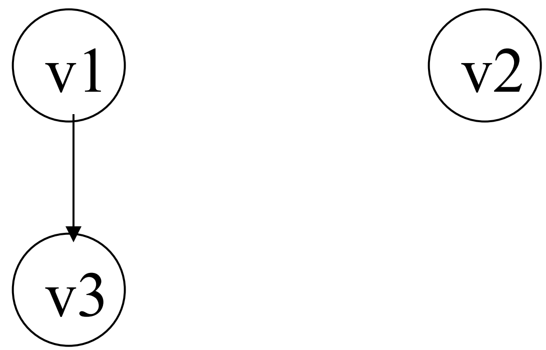
v2 is second variable in ordering. If v2 independent of a subset of its predecessors (possibly the empty set) in ordering conditioned on a disjoint subset of predecessors (including possibly all its predecessors), then the latter subset is its parents, else if latter subset is empty then v2 is a "root"

Since P(v2|v1) = P(v2) ….

P(v1)   v1   v2   P(v2)

v3 is third variable in ordering. Since P(v3|v1,v2) = P(v3|v1), …:

P(v1)

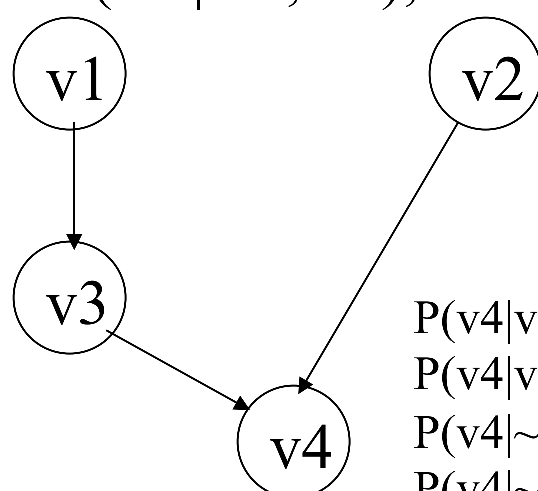


P(v2)

P(v3|v1)
P(v3|~v1)
P(~v3|v1) = 1 – P(v3|v1)
P(~v3|~v1) = 1 – P(v3|~v1)

Since P(v4| v1, v2, v3) = P(v4 | v2, v3), …

P(v1)

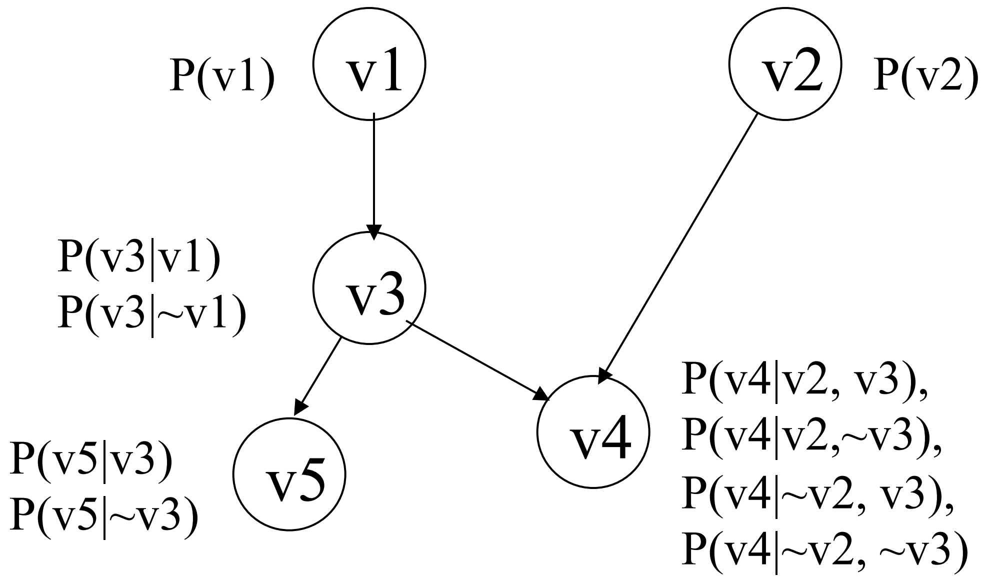


P(v2)

P(v3|v1)
P(v3|~v1)

P(v4|v2, v3), P(~v4|v2,v3) = 1-P(v4|v2,v3
P(v4|v2,~v3), …
P(v4|~v2, v3), …
P(v4|~v2, ~v3), …

Since P(v5|v1,v2, v3, v4) = P(v5|v3),…:



Components of a Bayesian Network: a topology (graph) that qualitatively indicates displays the conditional independencies, and probability tables at each node

Semantics of graphical component: for each variable, v, v is independent of all of its non-descendents conditioned on its parents

Where does knowledge of conditional independence come from?

a) From data. Consider congressional voting records. Suppose that we have data on House votes (and political party). Suppose variables are ordered
Party, Immigration, StarWars, ….

Party P(Republican) = 0.52 (226/435 Republicans
209/435 Democrats)

To determine relationship between Party and Immigration, we count

Actual Counts

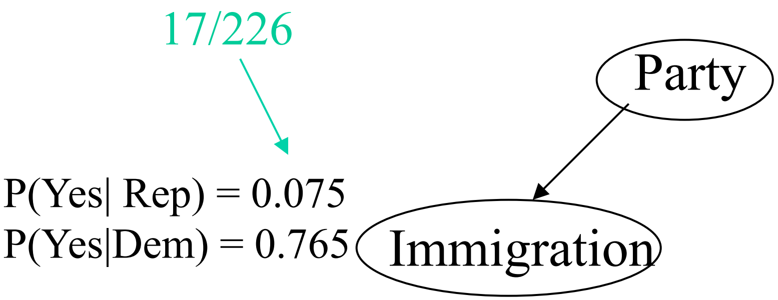
	Immigration Yes	Immigration No
Republican	17	209
Democrat	160	49

Predicted Counts (if Immigration and Party independent)

	Yes	No
Republican	92	134
Democrat	85	124

Very different distributions – conclude **dependent**

P(Rep)*P(Yes) * 435
= 0.52 * (17+160)/435 * 435



P(Republican) = 0.52 (226/435 Republicans
209/435 Democrats)

Actual Counts

	Immigration	
	Yes	No
Republican	17	209
Democrat	160	49

Consider StarWars
- Is StarWars independent of Party and Immigration?
  - (i.e., is P(StarWars|Party, Immigration) approx equal P(StarWars)
    - for all combinations of variable values?)
  - if yes, then stop and make StarWars a "root", else continue
- Is StarWars independent of Immigration conditioned on Party?
  - if yes, then stop and make StarWars a child of Party, else continue
- Is StarWars independent of Party conditioned on Immigration?
  - if yes, then stop and make StarWars a child of Immigration, else continue
- Make StarWars a child of both Party and Immigration

17/226

Party P(Republican) = 0.52 (226/435 Republicans
209/435 Democrats)

P(Yes| Rep) = 0.075
P(Yes|Dem) = 0.765 Immigration

	Actual Counts Immigration Yes	 No	Actual Counts StarWars Yes	 No
Republican	17	209	219	7
Democrat	160	49	24	185

Consider StarWars
Is StarWars independent of Party and Immigration?

Actual Counts

	Immigration Yes, StarWars Yes	Immigration Yes, StarWars No	Immigration No, StarWars Yes	Immigration No, StarWars No
Republican	14	3	205	4
Democrat	8	152	16	33

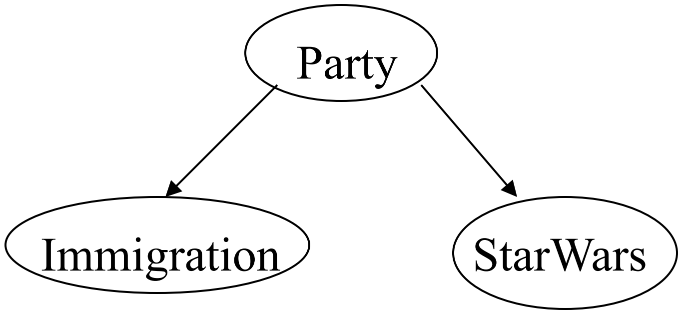
P(Rep & Imm=Y)P(SW=Y)435

Predicted Counts

	Immigration Yes, StarWars Yes	Immigration Yes, StarWars No	Immigration No, StarWars Yes	Immigration No, StarWars No
Republican	9.5	7.5	117	92
Democrat	89	71	27	22

different – not independent

Further tests might indicate

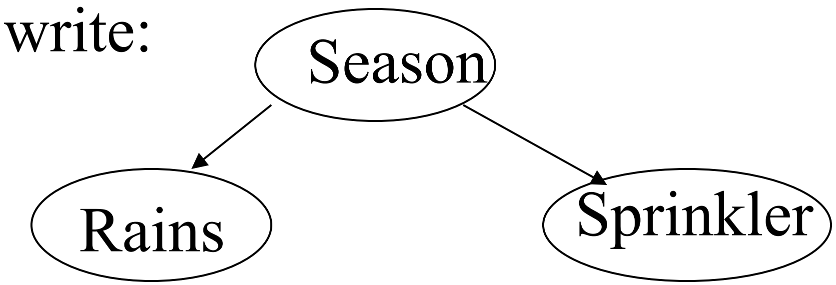


i.e., Immigration and StarWars are independent conditioned on Party

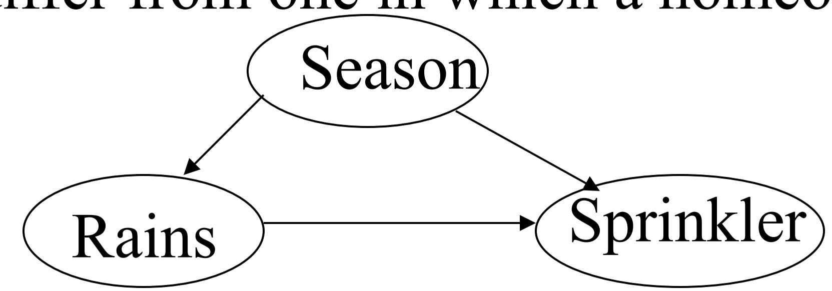
Where does knowledge of conditional independence come from?

b) "First principles"

For example, suppose that the grounds keeper sets sprinkler timers to a fixed schedule that depends on the season (Summer, Winter, Spring, Fall), and suppose that the probability that it rains or not is dependent on season. We might write:



This model might differ from one in which a homeowner manually turns on a sprinkler



Example (Empirical, Unsupervised): Clustering

Given data (vectors of variable values)

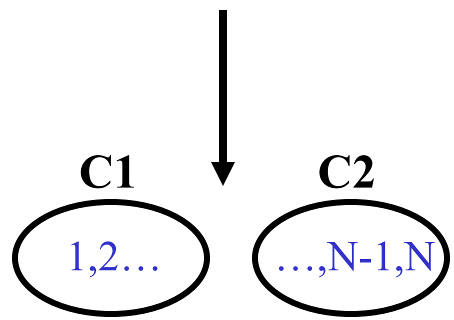
Compute a partition (clusters) of the vectors, such that vectors within a cluster tend to be similar, and vectors across clusters tend to be dissimilar

For example,

	V1	V2	V3	V4	……………	VM
1	0.3	0.7	0.1	-0.2	………….	-0.5
2	0.4	0.8	0.01	0.1	………….	-0.4
		……………………				
N-1	-0.3	0.1	1.01	0.8	………….	1.3
N	-0.5	0.03	1.1	0.9	………….	0.9

Cluster summary representations (e.g., the centroid)

	V1	V2	V3	V4 ……………	VM
1	0.3	0.7	0.1	-0.2 ………….	-0.5
2	0.4	0.8	0.01	0.1 ………….	-0.4
		……………………			
N-1	-0.3	0.1	1.01	0.8 ………….	1.3
N	-0.5	0.03	1.1	0.9 ………….	0.9



0.35 0.75 0.05 -0.05 …. –0.45
(centroid for C1)

-0.4 0.05 1.05 0.85 …. 1.1
(centroid for C2)

Using summary representations for inference

0.5 **?** 0.01 -0.12 …. **?**

C1 **C2**

1,2… …,N-1,N

0.35 0.75 0.05 -0.05 …. –0.45
(centroid for C1)

-0.4 0.05 1.05 0.85 …. 1.1
(centroid for C2)

0.5 **0.75** 0.01 –0.12 …. **–0.45**

K-means

```
Clustering K-Means (Data, K) {
    ClusterCentroids = K randomly selected vectors from Data
    for each d in Data
        assign d to cluster with closest centroid
    do {
        compute new cluster centroids
        for each d in Data
            assign d to cluster with closest centroid
    } while NOT termination condition
}
```

"closest": Euclidean distance