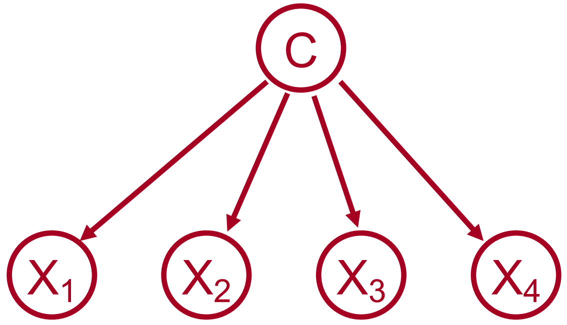
Principles of Machine Learning

Week 12: Probabilistic Machine Learning (Part 2)

Part 2A: Probabilistic Classifiers

Naïve Bayes Classifier (1)

- Simplest form of Bayesian classifier
  - A node for each variable in domain
  - **C** is **class** node
  - Other nodes are **evidence** nodes
  - Alternative view: C is Cause; others are Effects
  - **Arc** from class node to each evidence node



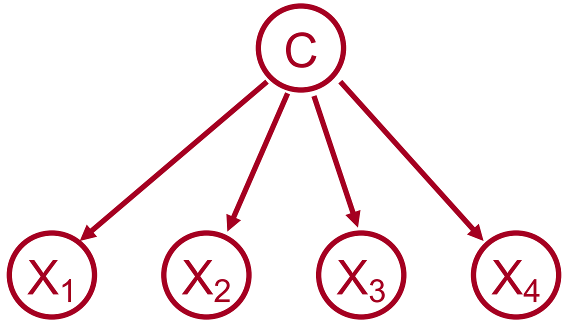
- "Naïve" assumption:
  - Evidence nodes assumed to be conditionally independent of each other, given the class node
  - Simplifies calculations, but may be incorrect

Naïve Bayes Classifier (2)

- For each arc, need set of probabilities (next slide):

  P(X_1=true | C=true) = 1 – P(X_1=false | C=true)

  P(X_1=true | C=false) = 1 – P(X_1=false | C=false)



- To classify a new instance:

  1: For each possible value of the class node,

  Calculate the probability of that class given the values of the other attributes
  (Use Bayes' Rule with Normalisation, assuming Cond. Indep.)

$$P(c_1 \wedge x_1 \wedge x_2 \wedge \ldots \wedge x_n) = P(c_1) \prod_i P(x_i | c_1)$$

  2: Normalise the probabilities of each class; select the most probable.

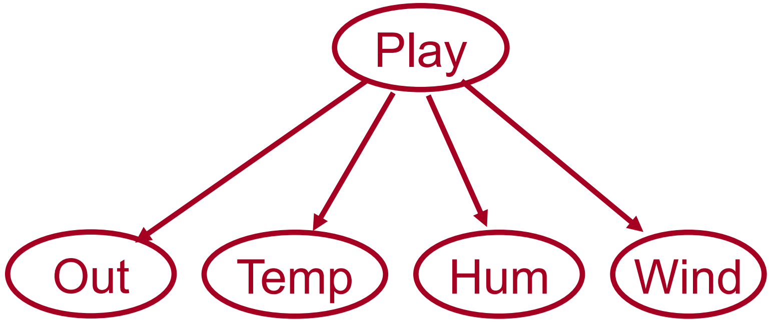
- Note: $P(c_j | x_1 \wedge \ldots \wedge x_n) = P(c_j \wedge x_1 \wedge \ldots \wedge x_n) / P(x_1 \wedge \ldots \wedge x_n)$
  $P(x_1 \wedge \ldots \wedge x_n)$ is fixed when normalising => $P(c_j | \text{--}) \propto P(c_j \wedge \text{--})$

Naïve Bayes Classifier (3)

- Training a Naïve Bayes Classifier:
  - Arc structure is fixed
  - Just estimate arc probabilities from the training data!
  - Probabilities for one arc: "**Conditional Distribution**"
- To estimate probabilities, can simply use frequencies from training data:
  - $P(X=x_1 \mid C=c_1) = N_{x_1 c_1} / N_{c_1}$
  - N_{c_1} = count of cases where $C=c_1$
  - $N_{x_1 c_1}$ = counts of cases where $X=x_1$ and $C=c_1$
- **Laplacian Smoothing** avoids 0 probabilities
  - Effectively adds *m* 'virtual observations' with everything seen once
  - $(N_{x_1 c_1} + m) / (N_{c_1} + m|X|)$
  - Typically use m=1

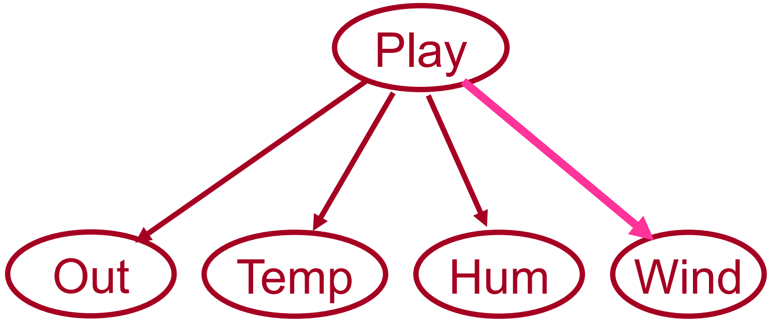
Naïve Bayes Example: Tennis (1)

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no



Naïve Bayes Example: Tennis (2)

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no



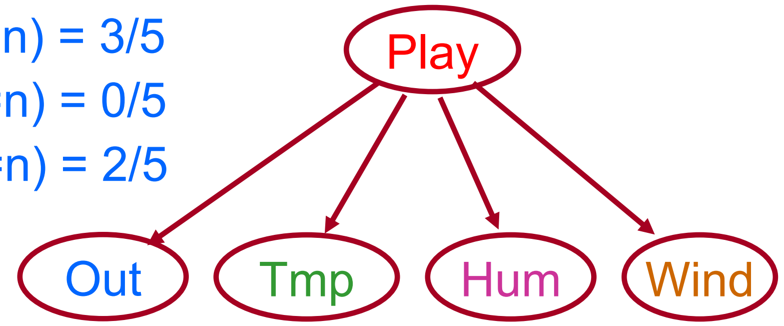
P(Wind=t | Play=y) = 3/9
P(Wind=f | Play=y) = 6/9

P(Wind=t | Play=n) = 3/5
P(Wind=f | Play=n) = 2/5

Now do the same for the other attributes …

Naïve Bayes Example: Tennis (3)

P(Out=s | Play=y) = 2/9	P(Out=s | Play=n) = 3/5
P(Out=o | Play=y) = 4/9	P(Out=o | Play=n) = 0/5
P(Out=r | Play=y) = 3/9	P(Out=r | Play=n) = 2/5



P(Tmp=h | Play=y) = 2/9	P(Tmp=h | Play=n) = 2/5
P(Tmp=m | Play=y) = 4/9	P(Tmp=m | Play=n) = 2/5
P(Tmp=c | Play=y) = 3/9	P(Tmp=c | Play=n) = 1/5

P(Hum=h | Play=y) = 3/9	P(Hum=h | Play=n) = 4/5
P(Hum=n | Play=y) = 6/9	P(Hum=n | Play=n) = 1/5

P(Wind=t | Play=y) = 3/9	P(Wind=t | Play=n) = 3/5
P(Wind=f | Play=y) = 6/9	P(Wind=f | Play=n) = 2/5

P(Play=y) = 9/14
P(Play=n) = 5/14
(prior probabilities)

Naïve Bayes Example: Tennis (4)

- Now classify new instance: sunny, cool, high, true: Play?

  Play is y or n. Evaluate probability of each given data:

  P(Play=y ∧ Out=s ∧ Tmp=c ∧ Hum=h ∧ Wind=t)

  = P(Play=y) × P(Out=s | Play=y) × P(Tmp=c | Play=y) × P(Hum=h | Play=y) × P(Wind=t | Play=y)

  = 9/14 × 2/9 × 3/9 × 3/9 × 3/9 **= 1/189**

  P(Play=n ∧ Out=s ∧ Tmp=c ∧ Hum=h ∧ Wind=t)

  = 5/14 × 3/5 × 1/5 × 4/5 × 3/5 **= 18/875**

  Normalise:

  P(Play=y | *data*) = 125/611 = **20.5%**

  P(Play=n | *data*) = 486/611 = **79.5%**

$$P(c_1 \wedge x_1 \wedge x_2 \wedge \ldots \wedge x_n) = P(c_1)\prod_i P(x_i \mid c_1)$$

- Conclusion: more likely NOT to play tennis today.

Part 2- 2

Bayesian Spam Filter (1)

Classify messages as spam/ham: Naïve Bayes often used

"Bag of words" representation of messages: Each word in a message is a feature

You are consulting for PhotoGram, a web service for sharing photos with short captions. They wish to automatically identify postings that are spam from the caption text. They have the following small set of captions that are spam and legitimate:

Spam:	*Legitimate:*
"click this link"	"puppy sleeping today"
"weight drugs link"	"good luck puppy"
"drugs news here"	"good sleeping"
	"news meeting today"

Using a Naïve Bayes classifier, compute the probability of the following two messages being spam: (1) "weight drugs news"; (2) "puppy news". Show all steps in your computation and explain any assumptions you make.

Bayesian Spam Filter (2)

SPAM	$\neg$SPAM
click this link	puppy sleeping today
weight drugs link	good luck puppy
drugs news here	good sleeping
	news meeting today

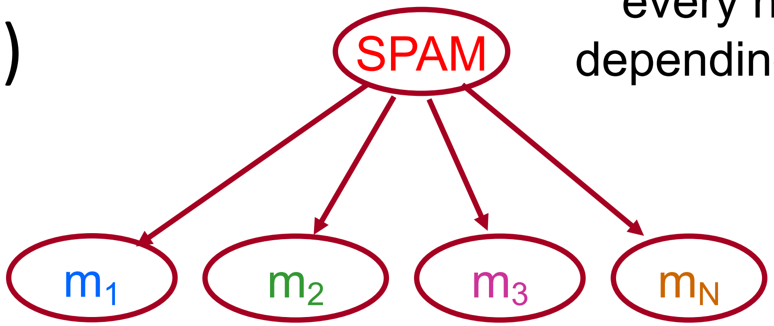
P(SPAM) = ?　　P($\neg$SPAM) = ?　　(no smoothing)

P("news"|SPAM) = ?

P("news"|$\neg$SPAM) = ?

Bayesian Spam Filter (2)

- For a message M = (m_1 m_2 m_3 …m_N), want to compute P(SPAM | M)



[New classifier created for every new message, depending on words in it]

- Compute:
  P'(SPAM | M) = P(SPAM) P(m_1|SPAM) P(m_2|SPAM) …
  P'(HAM | M) = P(HAM) P(m_1|HAM) P(m_2|HAM) …
  And normalise
- Examples:
  1. M is “puppy news”
  2. M is “weight drug news”

How do the calculations change if we use Laplacian smoothing? See spreadsheet.

Bayesian Networks: Syntax (1)

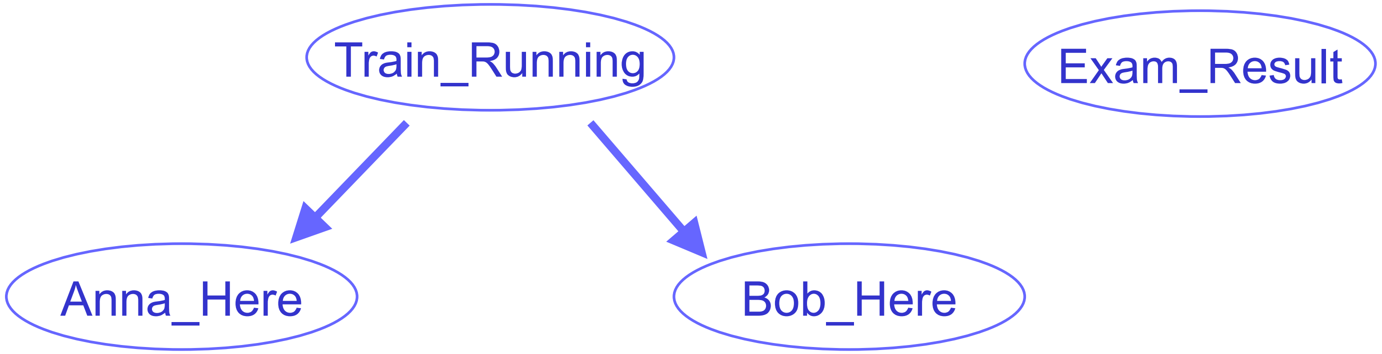
- Graphical notation for conditional independence assertions
  - Allows compact specification of full joint distribution
  - Consists of **Topology** + **Probabilities**
- Topology (Structure):
  - One node for every variable in domain
  - Arcs between nodes, forming a **directed acyclic graph** (DAG): Roughly speaking, arc X→Y means "X **directly influences** Y"
- Probabilities:
  - A **local conditional distribution** for each node given its parents: **$P(X_i \mid Parents(X_i))$**
  - Represented as Conditional Probability Table (**CPT**) giving the distribution over **X_i** for each combination of its parent values

Bayesian Networks: Syntax (2)

- Key point:

  Topology of network describes conditional independence assertions

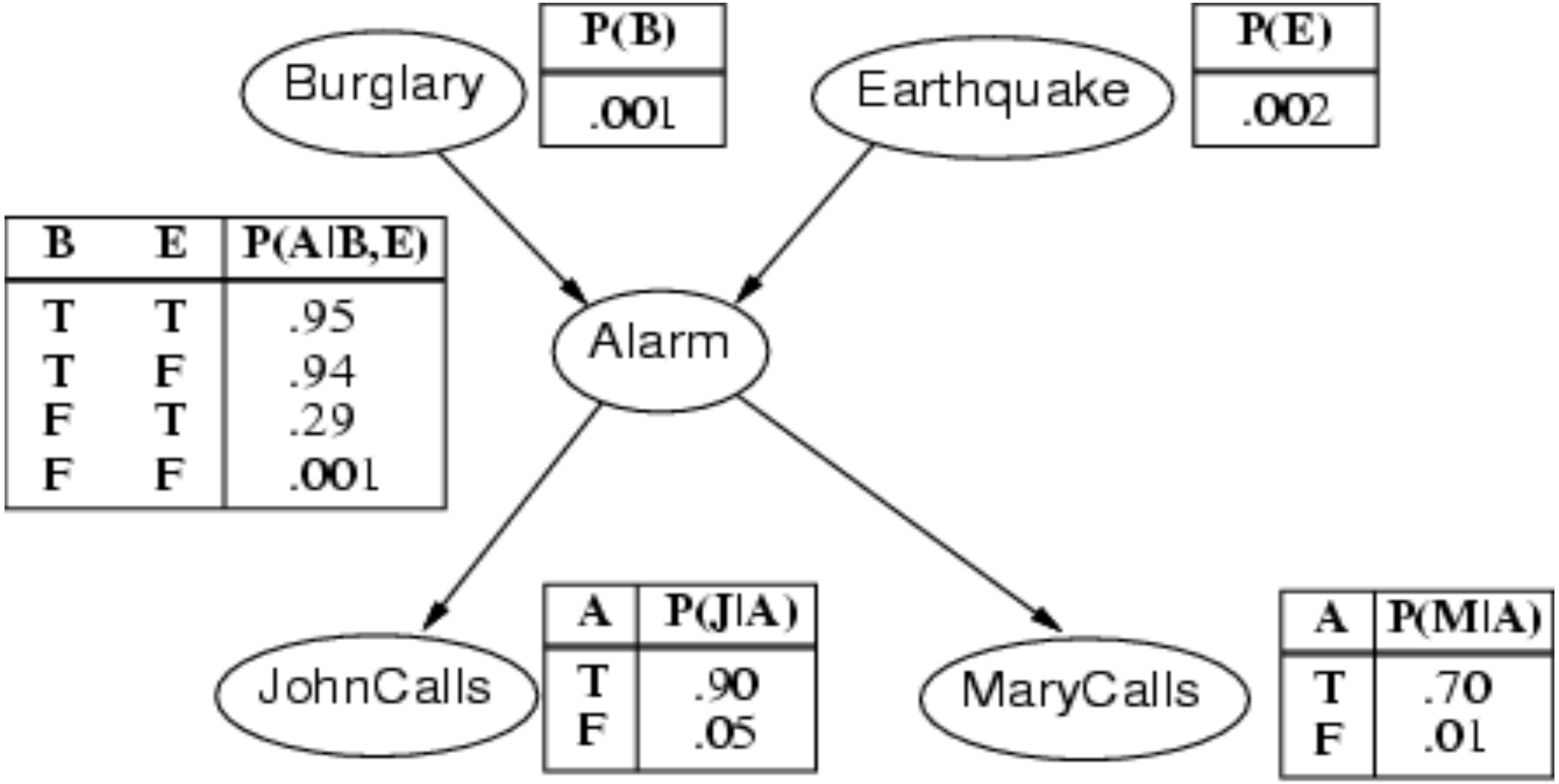
- Example:



ExamResult is independent of the other variables

Anna_Here and **Bob_Here** are conditionally independent given **Train_Running**

Bayesian Networks: Earthquake Example

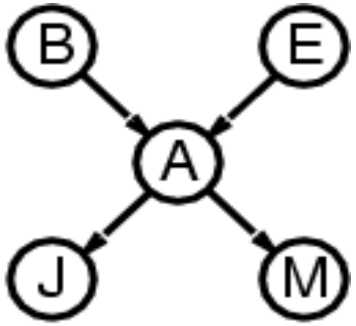


[Russell & Norvig]

Bayesian Networks: Semantics

- The full joint distribution is the product of the local conditional distributions:

$$P(x_1 \wedge x_2 \wedge \ldots \wedge x_n) = \prod_{i}^{n} P(x_i \mid Parents(X_1))$$



- For example:

  What is the probability that alarm has activated, but neither burglary nor earthquake has occurred, and both John and Mary call?

  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

  $= P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$

  $= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 =$ **0.000628**

Bayesian Networks: Compactness

- A CPT for a node X_i with k parents has 2^k rows
  - One for each combination of parent values
  - Assuming Binary variables
  - Each row requires one probability value, for **X_i=true** (probability for **X_i=false** is just 1 – prob. for **X_i=true**)
  - If each node has at most **k** parents, the complete network requires the order of **$(n \cdot 2^k)$** numbers
- Much more compact than full joint distribution:
  - Grows **linearly with n** (assuming a fixed max. number of parents), compared to the full joint distribution which grows **exponentially**
- For burglary network:
  - Have 1 + 1 + 4 + 2 + 2 = **10** numbers
  - Full joint distribution would have 2^5-1 = **31** numbers

Part 2 - 4

Constructing a Bayesian Network Manually

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For i = 1 to n
   i. Add $\mathbf{X_i}$ to the network
   ii. Its potential parents are its **predecessors** in the ordering: Only add arc to $\mathbf{X_i}$ if a potential parent **directly influences** it

- Note 1: Using Conditional Independence, rule is:

  Select parents such that $\mathbf{P(X_i \mid Parents(X_i)) = P(X_i \mid X_1, \ldots X_{i-1})}$

- Note 2: the ordering of variables will determine how the network can be structured

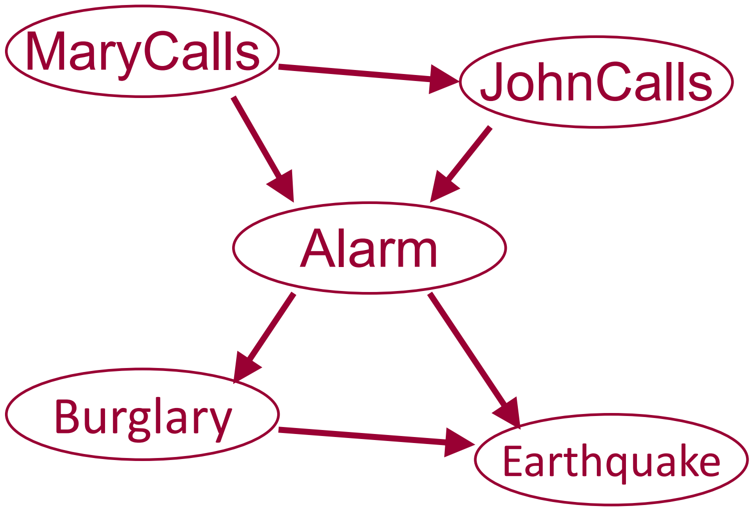
  Think about which variables **cause** which

  Add the **root causes** first, then the variables they directly influence, and so on

  Last added are those with **no** direct causal influence on any others

Constructing a BN: Example

Node Ordering: M, J, A, B, E



First add MaryCalls: no parents.

Add JohnCalls: If Mary calls, alarm likely to have activated, so more likely that John calls => add arc

Add **Alarm**: If Mary and John call, more likely that alarm has activated than if one or neither call => add arcs from both

Add **Burglary**: If we know alarm state, then **JohnCalls** or **MaryCalls** adds no more info: **P(B | A,J,M) = P(B | M)** => arc from Alarm only

Add **Earthquake**: If Alarm on, more likely there was an earthquake, unless there was a Burglary to explain the alarm => add arcs from both.

Learning a BN from Data (1)

- Two sub-tasks in learning a BN:
  - Learn the structure
  - Estimate the probabilities
  - Decomposable: can do separately
- Several approaches to structure learning
  - Bad news: finding optimum network is NP-Hard!
  - Typically combine quality score with search heuristics
- Examples of scores:
  - Minimum description length
  - Probability of network given data
- Examples of search procedures:
  - Genetic, hill-climbing, conditional independence tests

Learning a BN from Data (2)

- K2 [Cooper & Herksovits, 1992]

  Basis: Which of two structures more likely, given DB?

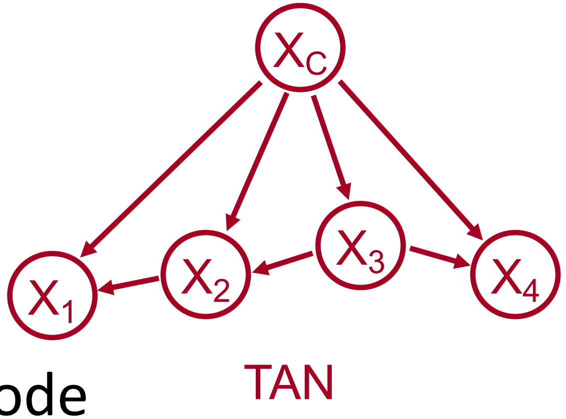
  i.e. calculate $P(B_{S_i}|D) \,/\, P(B_{S_j}|D)$

  equivalent to $P(B_{s_i},D) \,/\, P(B_{S_j},D)$

- Others assume a restricted form of network

  TAN: Tree Augmented Naïve Bayes
  [Friedman et al '97]

  Max. of 1 dependency between children of class node

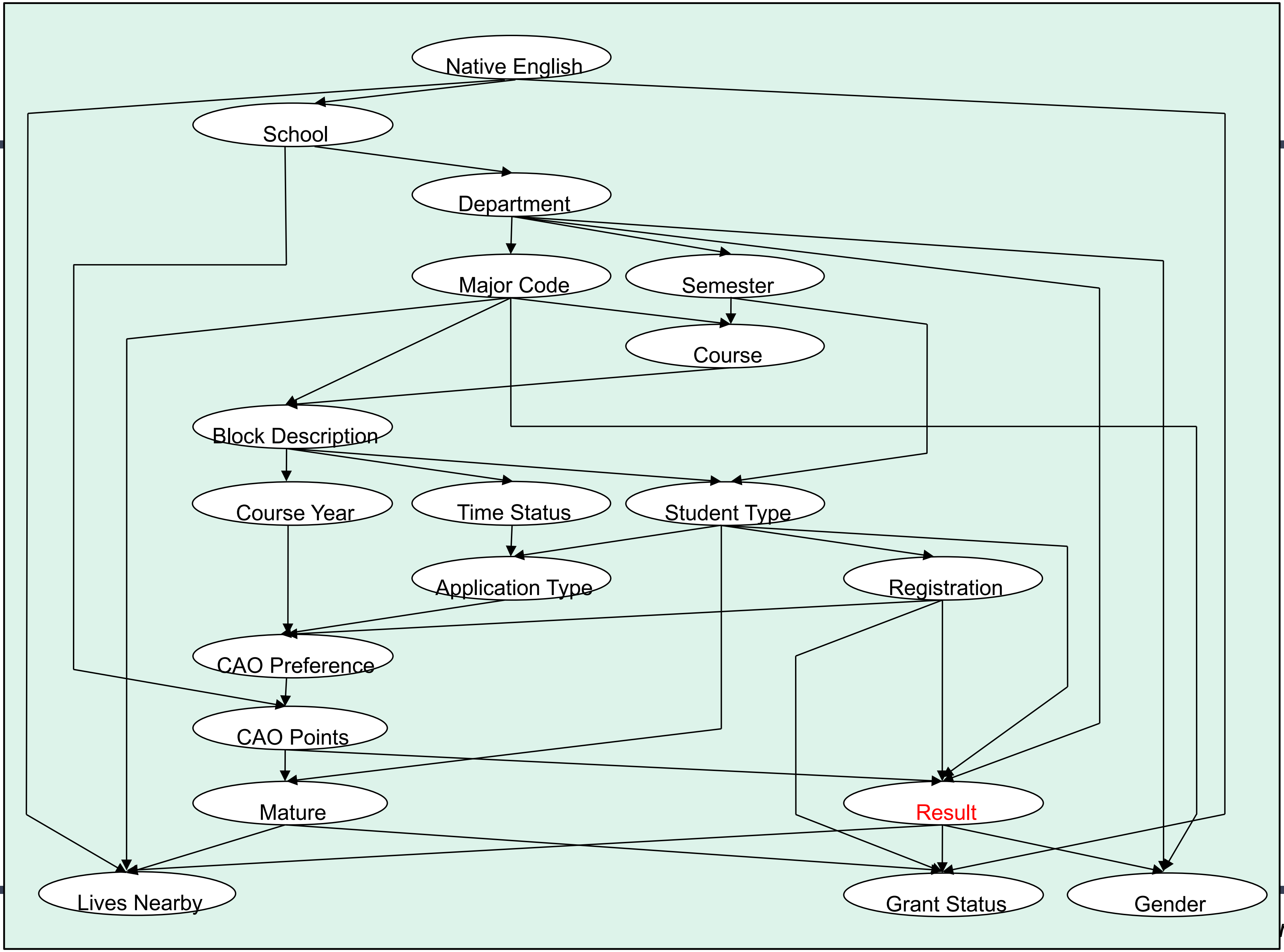


- After learning structure, learn parameters from data

  Very similar to learning Naïve Bayes parameters.

Data Exploration with BNs

- BNs help identify correlations in data (pos. or neg.)
  - Rather than just pairwise correlations, multiple ones considered simultaneously
  - Absence of arc: No correlation
- Inductively learn BN from data
  - Examine it to explore relationships
- Example: Madden, Lyons & Kavanagh, AICS 2008
  - Analyse student records with BNs, Decision Trees & Rules
  - Evidence-based understanding of how a variety of factors affect students' examination performance
  - BN learned with Minimum description length (MDL) score and hill-climbing search
  - Note: Data from another college, not NUI Galway


Native English
School
Department
Major Code
Semester
Course
Block Description
Course Year
Time Status
Student Type
Application Type
Registration
CAO Preference
CAO Points
Mature
Result
Lives Nearby
Grant Status
Gender

Data Exploration with BN: Example

- Several 'obvious' relationships:

  E.g. School → Dept → Major Code → Course

- Others less obvious but logical:

  E.g. CAO Preference → CAO Points

- Markov Blanket of a node:

  Its parents, its children and its children's parents

  Nodes outside MB do not affect it

- Markov blanket of Result node:

  CAO Points, Department, Major Code,
  Grant Status, Lives Nearby, Native English, Registration, Mature, Gender,
  Student Type

Classification using a BN

- Construct BN by induction from training DB

  Class variable not special

- To classify a new case:

  Generalised version of Naïve Bayes classification

  Assume value of X_c unknown, all others known

  For each possible value of X_c calculate joint probability of that instantiation of all variables:

$$P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n) = \prod_{i=1}^{n} P(X_n = x_n | \Pi_n = \pi_n)$$

  Normalise resulting probabilities

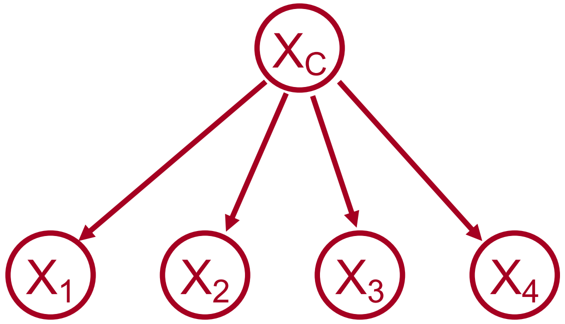
  Multiply by cost matrix if required

- Note:

  Only need to consider nodes in **Markov Blanket** of X_c

Restricted Bayesian Classifiers (1)

- **Naïve Bayes**
  (saw already)



- Assumptions
  1. All variables relevant to classification (tolerates irrelevant)
  2. Other vars conditionally independent of each other
  3. Direction of influence is from class var to others
     (i.e. class var is root cause)
- Relax Assumption 1 (and 2, weakly)

  Use subset of variables

  Selective Naïve Bayes [Langley & Sage, 1994]

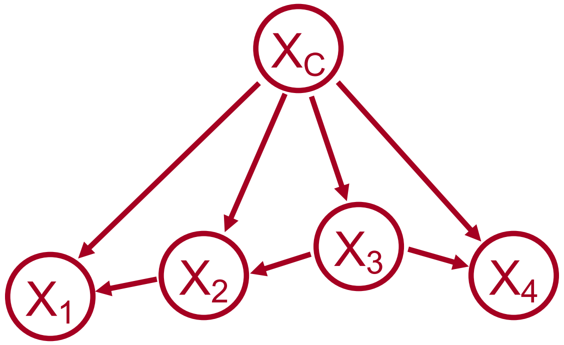
Restricted Bayesian Classifiers (2)

- Relax Assumption 2:

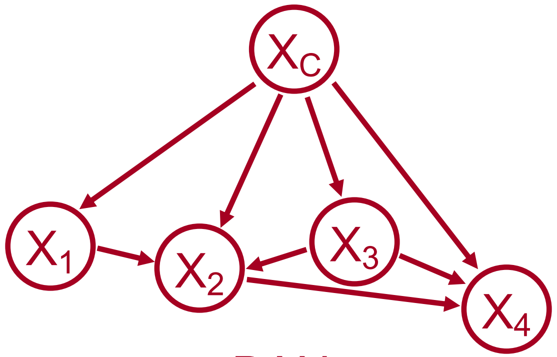
  Additional dependencies between variables

  Tree Augmented Naïve Bayes (TAN)
  [Friedman et al 1997]

  Bayesian Network Augmented Naïve Bayes (BAN)
  [Cheng & Greiner 2001]



TAN



BAN

Learning Objectives Review

You should now be able to …

- Discuss the motivation for handling uncertainty in ML
- Distinguish between prior and conditional probability
- Demonstrate understanding of how to use the axioms of probability and Bayes' rule
- Describe and apply the Naïve Bayes classifier to inductive learning problems
- Show how Bayesian Networks represent influence and independence of variables
- Discuss how BNs can be used for classification & data exploration.