General -

Probabilistic – deal with Uncertainty – unknown, noise, model error (Declarative, reasoning pattern, learning methods)

Graphic – complex system (Allow Represent very complex structure) so that we can efficiently reasoning using general purpose algorithms; limit the parameterization so it is feasible to learn from data;

Model – clear **representation** of our understanding of the world (Can plug in any algorithm)

Random variables X1,…Xn =>factors in the relationship ---------- represented by **Nodes**

Joint distribution P(X1,…Xn) => Estimate

**BN** – directed graph \*(medical diagnose)

**Markov Network** – undirected graph \*(image recongnization)

Content covered –

* **Representation** – Directed, undirected, temporal, plate models
* **Inference** (Reasoning) – Exact reasoning, approximate reasoning, decision making
* **Learning (**Train from historical data) – parameters, structure, with or without complete data

Probabilistic Distribution –

X1 {low, high} | X2 {easy, hard} | X3 {A, B, C} # 3 random variables

Joint Distribution -> CPT \* all discrete variable – 2 \* 2 \* 3 = 12 possibilities (All sum to 1)

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | P(X1, X2, X3) |
| Low | Easy | A | 0.123 |
| Low | Easy | B | 0.145 |
| Low | Easy | C | … |
| Low | Hard | A | … |
| Low | Hard | B | .. |
| Low | Hard | C |  |
| High | Easy | A |  |
| High | Easy | B |  |
| High | Easy | C |  |
| High | Hard | A |  |
| High | Hard | B |  |
| High | Hard | C |  |

Conditional Probability -

If conditioned on X1 = Low ------– P(X2,X3|X1=low)

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | P(X1, X2, X3) |
| Low | Easy | A | 0.123 |
| Low | Easy | B | 0.145 |
| Low | Easy | C | … |
| Low | Hard | A | … |
| Low | Hard | B | .. |
| Low | Hard | C |  |
| High | Easy | A |  |
| High | Easy | B |  |
| High | Easy | C |  |
| High | Hard | A |  |
| High | Hard | B |  |
| High | Hard | C |  |

Only keep unmarked row where X1 = Low \* **Reduction**

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | P(X1=low, X2, X3) |
| Low | Easy | A | 0.123 |
| Low | Easy | B | 0.145 |
| Low | Easy | C | … |
| Low | Hard | A | … |
| Low | Hard | B | .. |
| Low | Hard | C |  |

P(X1=low, X2, X3) don’t sum to 1 anymore, so need to **normalize** ------ eachp() / sum(restP())) = a list of P()s sum to 1 AND it is P(X2,X3|X1=low)

Marginalization – Sum(p()) = Marginal p() for X1 = Low; Sum(p()) = Marginal p() for X1 = High

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | P(X1, X2, X3) |
| Low | Easy | A | 0.123 |
| Low | Easy | B | 0.145 |
| Low | Easy | C | … |
| Low | Hard | A | … |
| Low | Hard | B | .. |
| Low | Hard | C | .. |
| High | Easy | A | .. |
| High | Easy | B | .. |
| High | Easy | C | .. |
| High | Hard | A | .. |
| High | Hard | B | .. |
| High | Hard | C | .. |

Factors – It is a function or a table(CPT), it takes random variables (X1,…Xn)\*specific combination and output a p() \*Contains all possible combinations

\*Fundamental building blocks for defining distributions in high-dimensional spaces, set of basic operations for manipulating these probability distribution.

Factor:A \*joint distribution P(X1,X2)

|  |  |  |
| --- | --- | --- |
| X1 | X2 | P(X1,X2) |
| Low | A | 0.213 |
| High | A | 0.276 |
| Low | B | 0.541 |
| High | B | 0.135 |

Scope of the factor A is {X1, X2}

Factor:B \*Un-normalized measure P(X1,X2=A)

|  |  |  |
| --- | --- | --- |
| X1 | X2 | P(X1,X2=A) |
| Low | A | 0.213 |
| High | A | 0.276 |

Factor:C \*Conditional Probability Distribution (CPD) P(X1 | X2)

|  |  |  |
| --- | --- | --- |
|  | Low | High |
| A | 0.3 | 0.7 |
| B | 0.4 | 0.6 |

Sum to 1 by row

Factor:D \* General Factor (Weight != p())

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Weight |
| A | Low | 30 |
| A | High | 5 |
| B | Low | 1 |
| B | High | 10 |

Factor Product –

Factor1 {A,B,C} X Factor2 {B,C} = Factor3 {A,B,C} \*across rows = p()

Factor Marginalization -

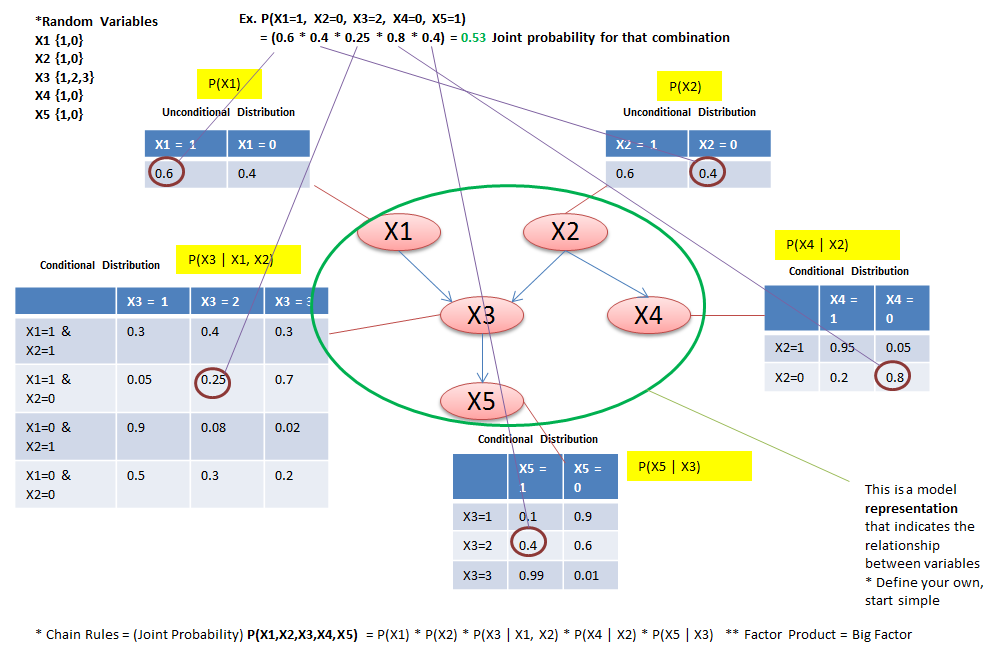
Factor1 {A,B,C} takes out B, aggregate p() for A,C = Factor2 {A,C}

Factor Reduction –

Factor1 {A,B,C} conditioned on A=1, keep row A=1 in factor1 = Factor2 \*Un-normalized factor

Bayesian Network Fundamentals –

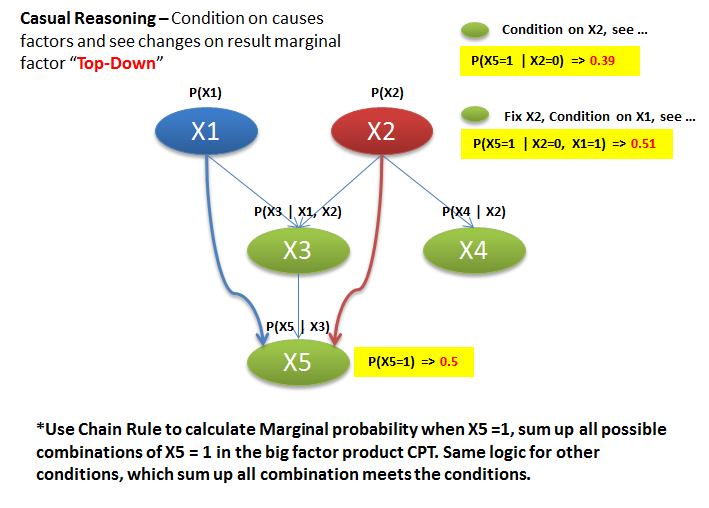
Semantics & Factorization –



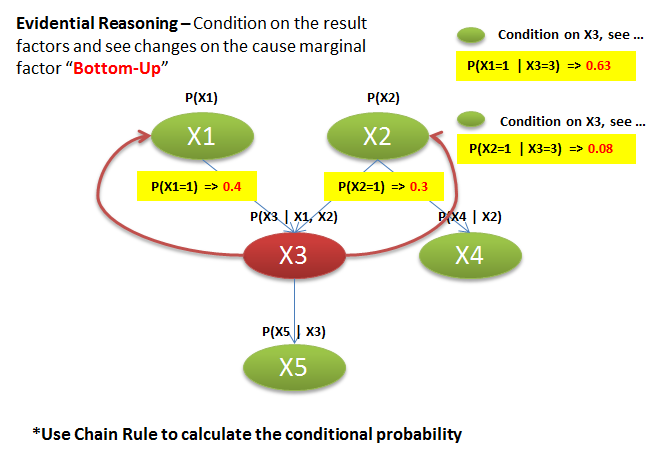
A Bayesian Network is directed acyclic (\*not a circle, can’t backward)) graph (DAG) whose nodes represents random variables (X1,…Xn). For each node Xi, a CPD – P(Xi | par(Xi)); BN represents a **joint distribution** via the **chain rule** for Bayesian networks.

Reasoning Patterns –

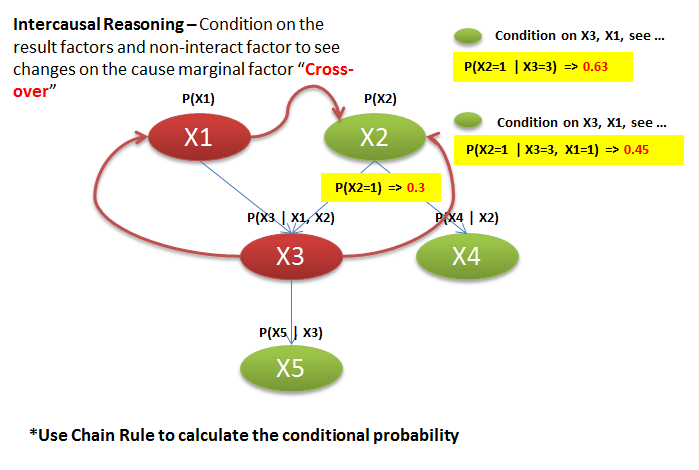
Casual Reasoning:



Evidential Reasoning:

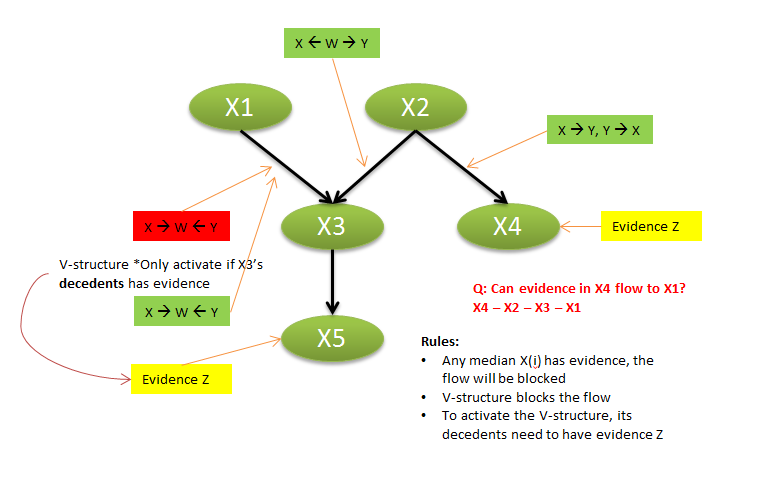


Intercausal Reasoning:



Flow of Probabilistic Influence -

“When can X influence Y?” {X 🡪 Y, Y 🡪 X, X 🡪 W 🡪 Y, Y 🡪 W 🡪 X, X 🡨 W 🡪 Y, [X 🡪 W 🡨 Y, V-structure exception]}

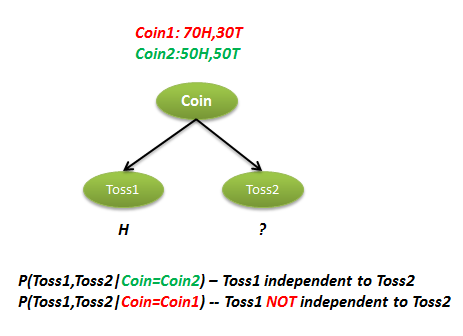


Conditional Independence -

X independent to Y 🡪 P(X,Y) = P(X)\*P(Y) ; P(X|Y) = P(X) ; P(Y|X) = P(Y)

X independent to Y conditioned on Z 🡪 P(X,Y|Z) = P(X|Z)\*P(Y|Z) ; P(X|Y,Z) = P(X|Z) ; P(Y|X,Z) = P(Y|Z)

X independent to Y conditioned on Z 🡪 P(X,Y,Z) ~ **factor.1** (X,Z) \* **factor.2**(Y,Z)



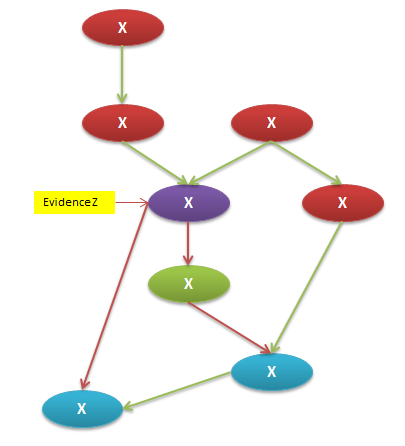
Independencies in Bayesian Networks –

**d-separated (Conditional independent)**: X and Y are d-separated in G given Z if there is no active trail in G between X and Y given Z

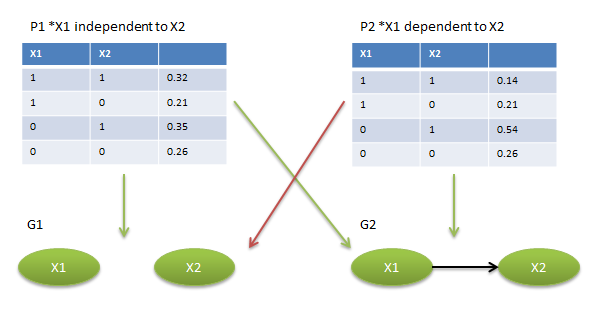
**Notation**: d-sepG(X, Y|Z)

**Theorem**: If P factorizes over G, and d-sepG(X,Y|Z) then P satisfies “X independent to Y condition on Z”

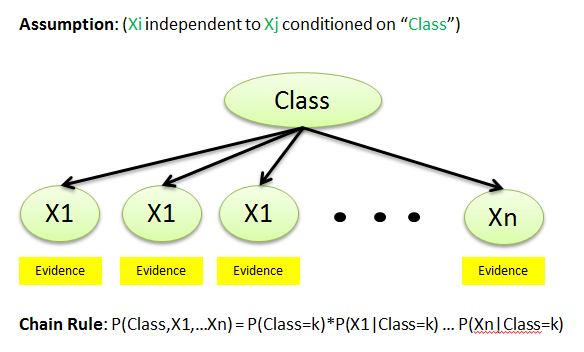
**Theorem**: Any node is d-separated from its non-descendants given its parents(Z)

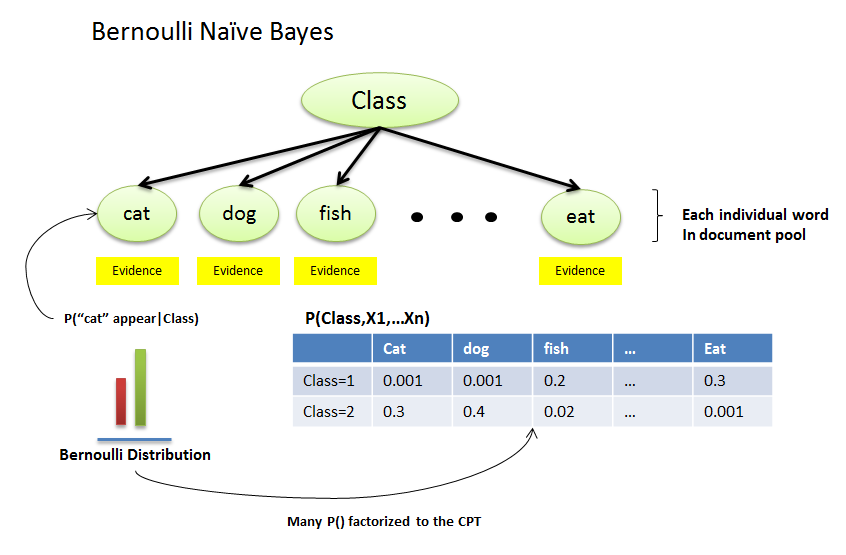


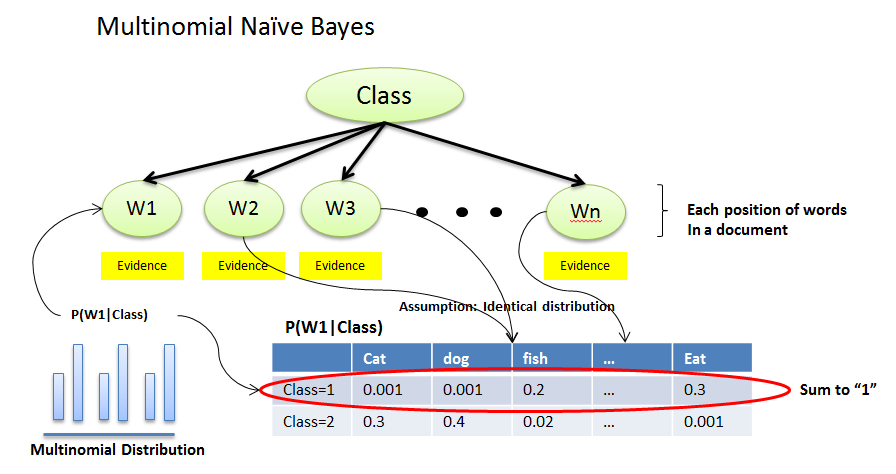
**I-Map:** It is a set of independencies**;** If P satisifies **I(G)**, we say that **G** is an I-Map (Independency map) of P; If P factorizes over G, then G is an I-map for P;



Naïve Bayes –





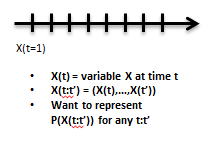


Template Models –

A template can be replicated in multiple copies (Not only used in one application, can be used in other situation as well). One of the common used called “**temporal models**” which represents models evolves with time. Share within Model. Share between Models

**Temporal models – (Kind of Dynamic Model) - DBN**

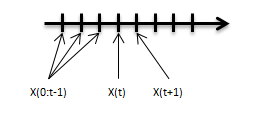
**First**, discretize time into buckets, Day1, Day2, etc.



How to **represent**?

Assumption #1: Markov Assumption – conditional independence assumption

( **X(t+1)** independent to **X(0:t-1)** | **X(t)** )



**P(X(0:t))**  = P(X(0)) \* ( P(X(t+1) | X(t)) … All ts ) Only depends on previous t, simplified

**Disadvantage**: ignore changing velocity, too strong assumption.

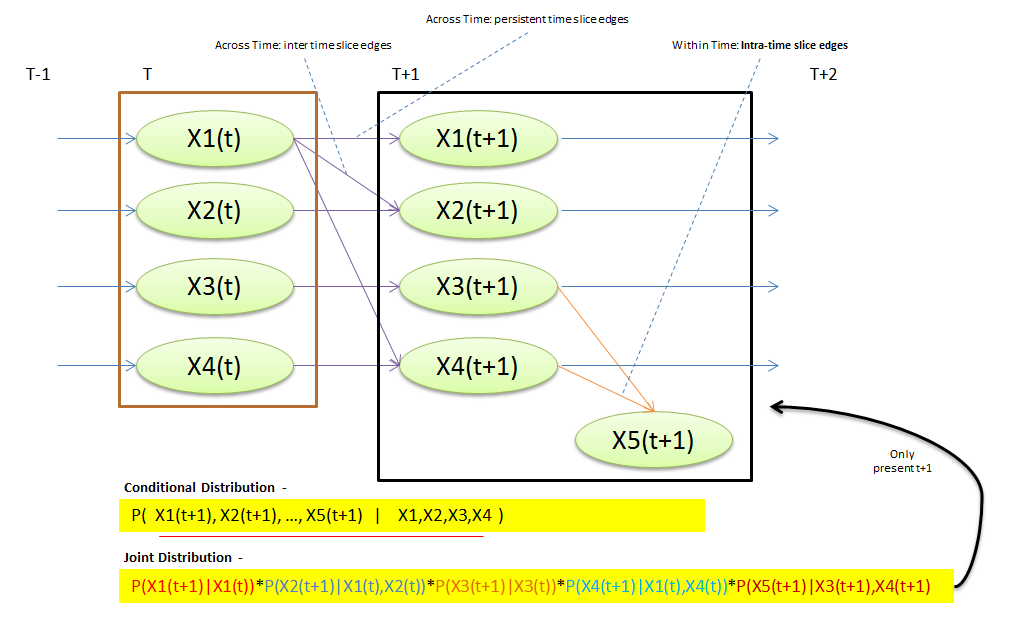
**Solution** -> Add more features (Model enrichment – better model represent reality), like X(velocity) to address it. **OR** Semi-Markov Model

Assumption #2: Time Invariance –

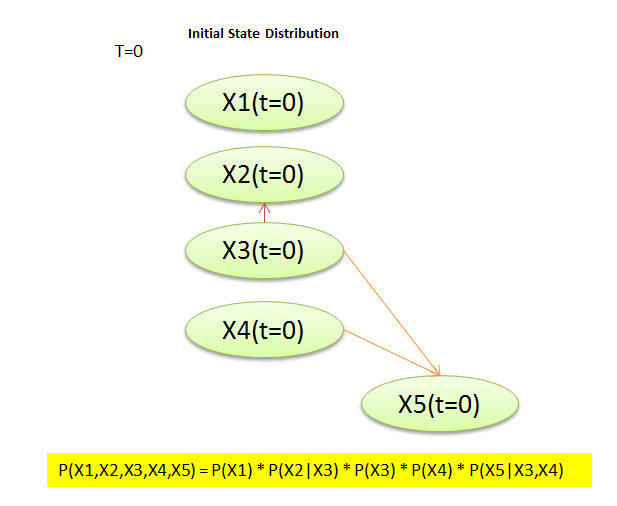
Template Probability Model P(X’|X), \* From T1 to T2, or T6 to T7, use the same model (The dynamic system is the same, not depend on time:t) – **improve** -> Add more features (Model enrichment – better model represent reality

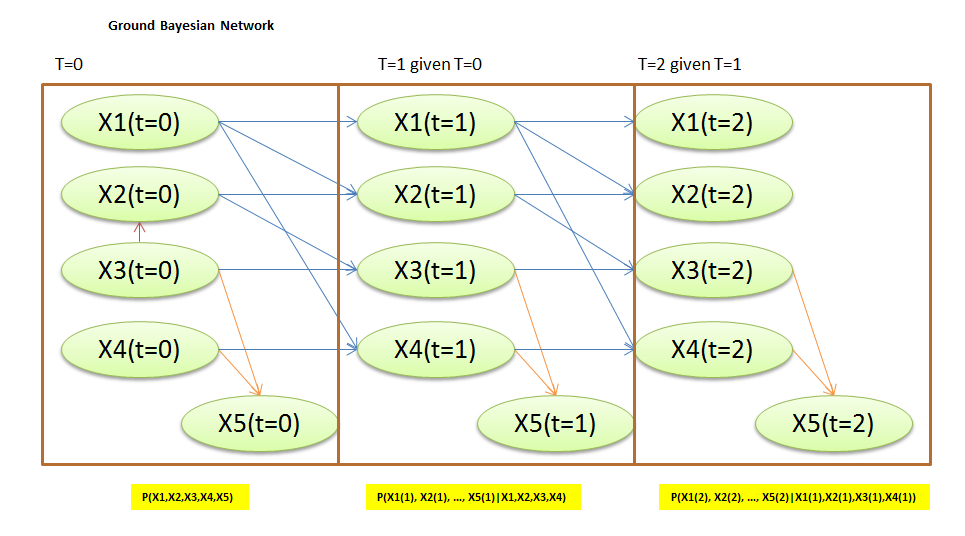
For all t: P(X(t+1) | X(t)) = P(X’ | X)

Template Transition Model –



Above is the simple “2 Time Slices Bayesian Network” – **2TBN (template)**



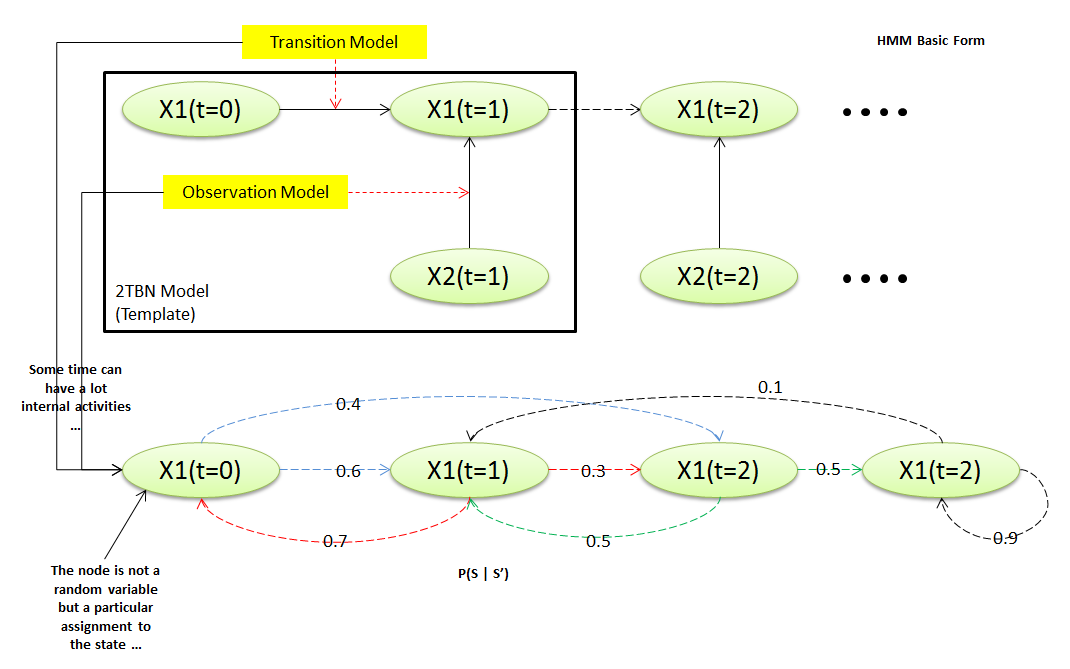


Above **ground network** - use the **template model** combine with initial state to roll over n periods.

**DBNS** are a compact representation for encoding structured distributions over arbitrarily long temporal trajectories. But the assumptions {**Markov assumption**; **Time invariance**} is strong, may need **add more features** to relax the model and make it more realistic.

Template Models – **Hidden Markov Models - (HMM)**

HMMs can be viewed as a subclass of DBNs; HMMs seem unstructured at the level of random variables; HMM structure typically manifests in sparsity and repeated elements within the transition matrix; Widely used in a wide variety of applications for **modeling sequences.**

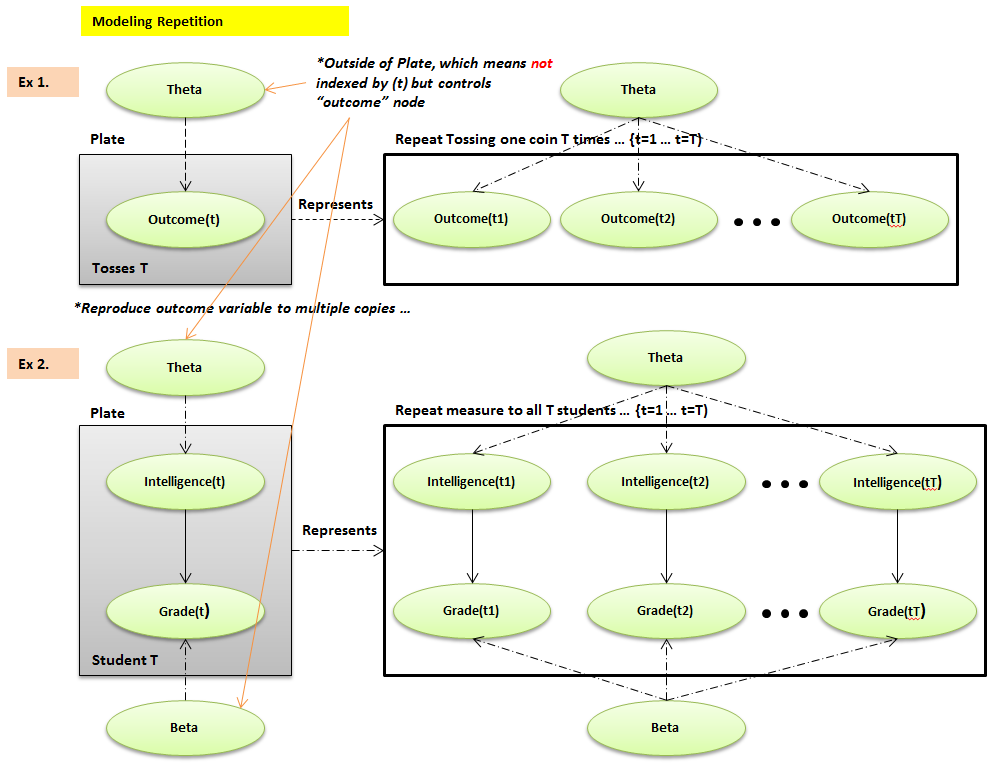


Template Models – **Plate Model**

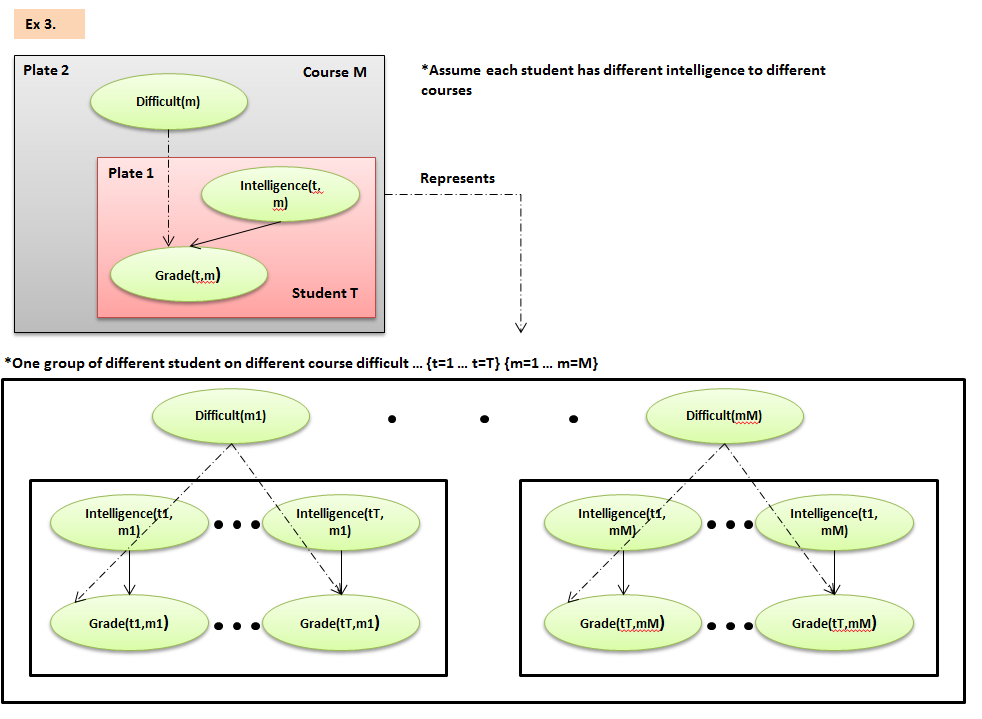
Multiple objects of the same type (plate Model); Example, template variable “Grade(m\*t)” and template parent “Intelligence, Difficult”; **CPD = P(Grade | Intelligence, Diffcult {m X t}); Ground Network:** extend to as many students or courses as you want.

Template for an initite set of BNs, each induced by a different set of domain object(Student, course)

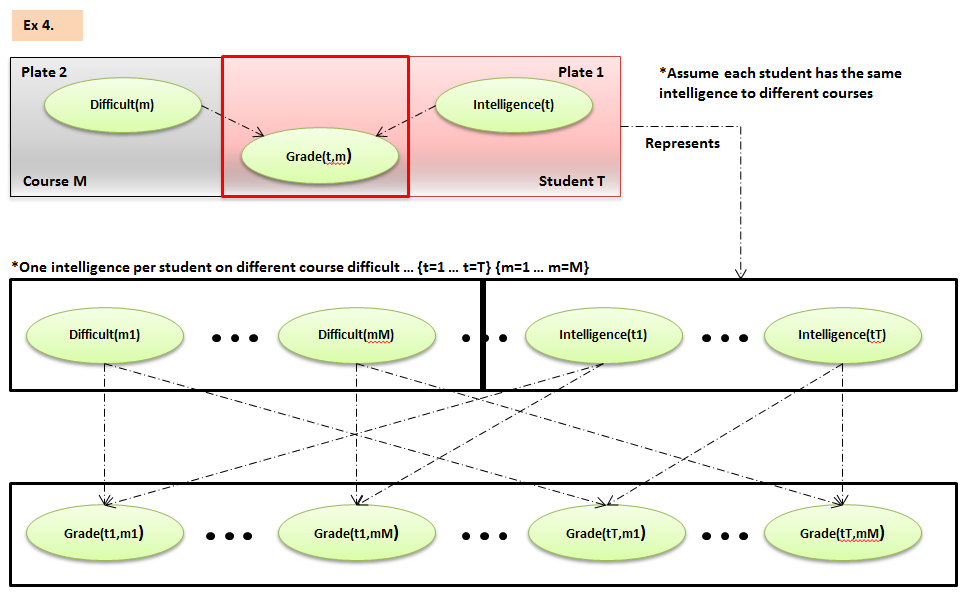
Parameters and structure are reused within a BN and across different BNs (one object, between objects)



**Nested Plates – How overlap different plates (Like nested for loop)**

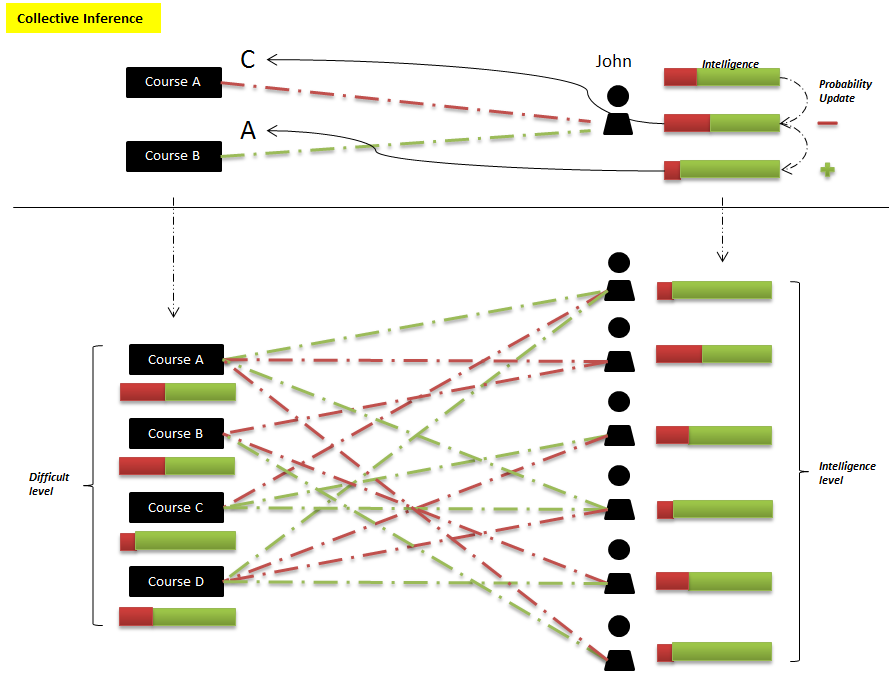
****

**Overlapping Plates**

****

**Collective Inference – Why it is useful?**

Models encode **correlations** across multiple objects,allowing collective inference.

****

Structured CPDs – Conditional Probability Distribution

**Tabular CPD** can’t handle many parents’ nodes … Not suitable for real world application

**General CPD** – **P(X | Y1,…,Yk)** specifies distribution over X for each assignment Y1,…, Yk 🡪 We can use any ***function*** to specify a factor over the scope - (X, Y1, …, Yk) such that sum up all factors for all assignments will get “1”.

|  |  |
| --- | --- |
| **CPD Models** | **Description** |
| Deterministic CPDs | Deterministic function |
| Tree-structured CPDs |  |
| Logistic CPDs & generalizations |  |
| Noisy OR / AND |  |
| Linear Gaussians & generalizations |  |

Context-specific Independence -

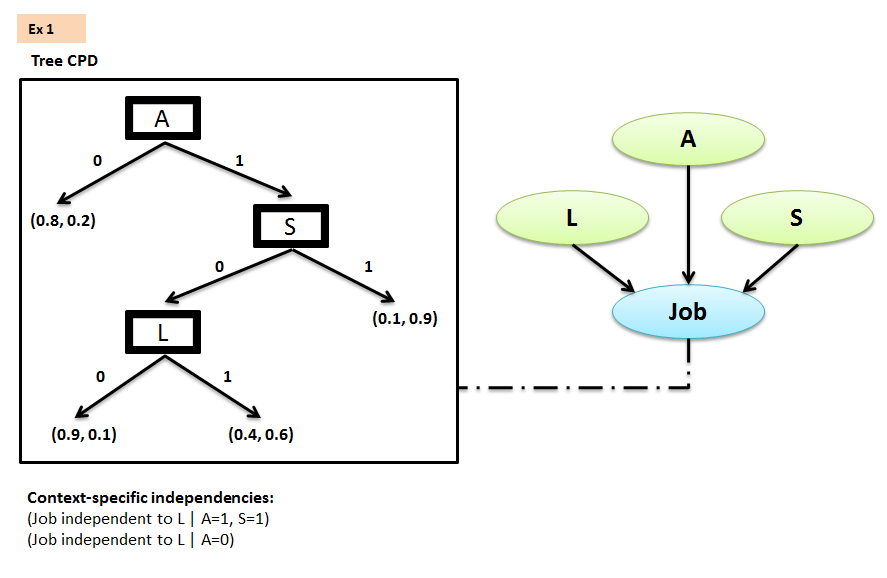
**X** independent to **Y** conditioned on **Z, C** (Depends on which condition **C**) (**C** is a condition of **Z**)

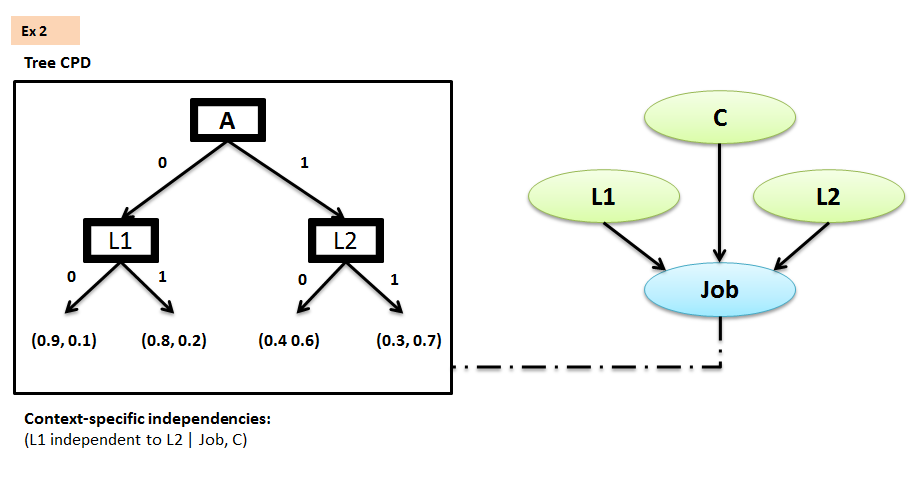
\*Only Z=1, **X** independent to **Y**

**Tree-Structured CPDs -**

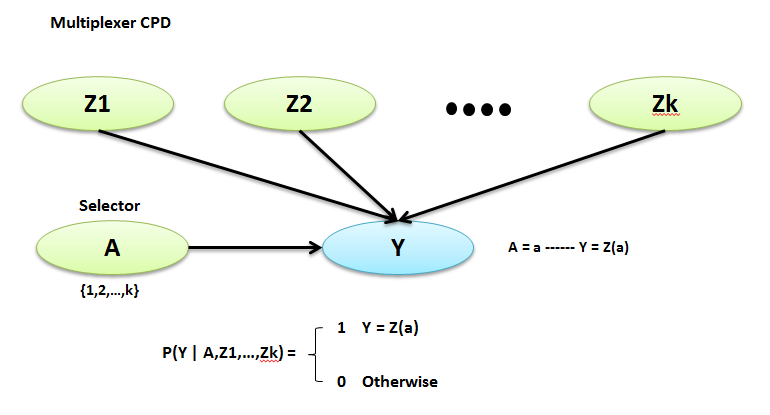
Compact CPD representation that captures context-specific dependencies

Relevant in applications – Hardware configuration, Medical settings, Dependence on agent’s action, perceptual ambiguity



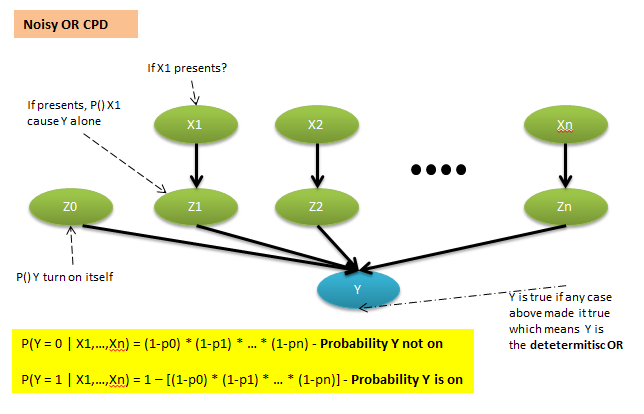


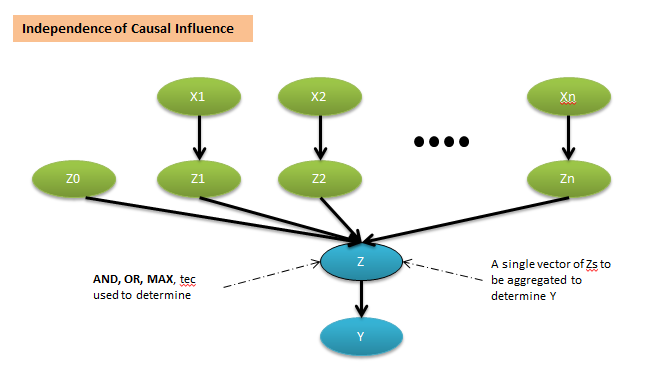
Multiplexer CPD –

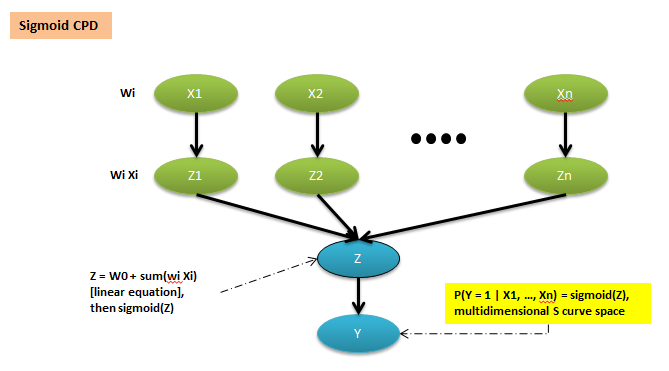


Noisy OR / AND CPD - **Independence of Causal Influence –**

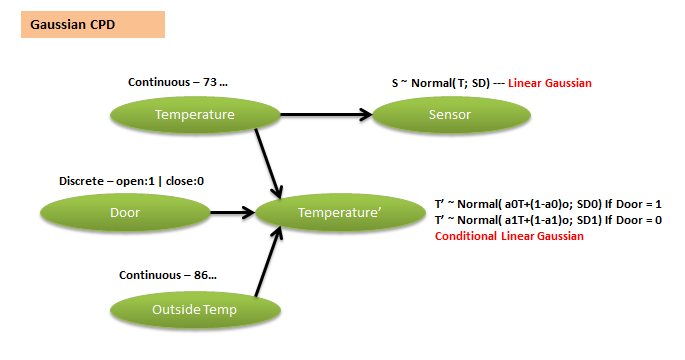
A lots of variables evenly contribute to a node

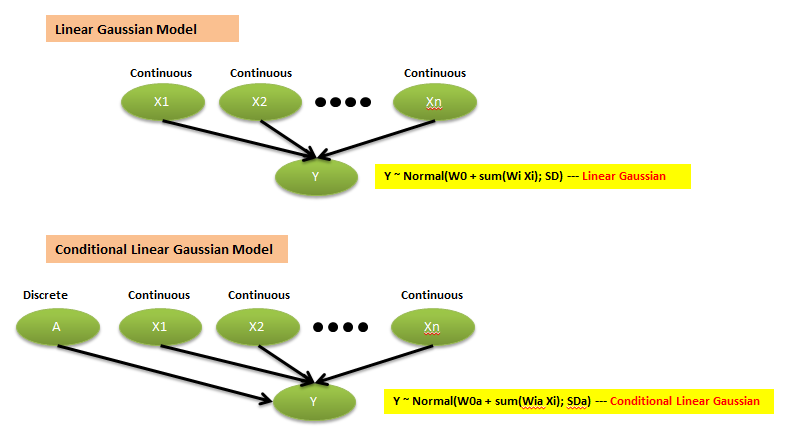






Linear Gaussians CPD - **Continuous Variable**

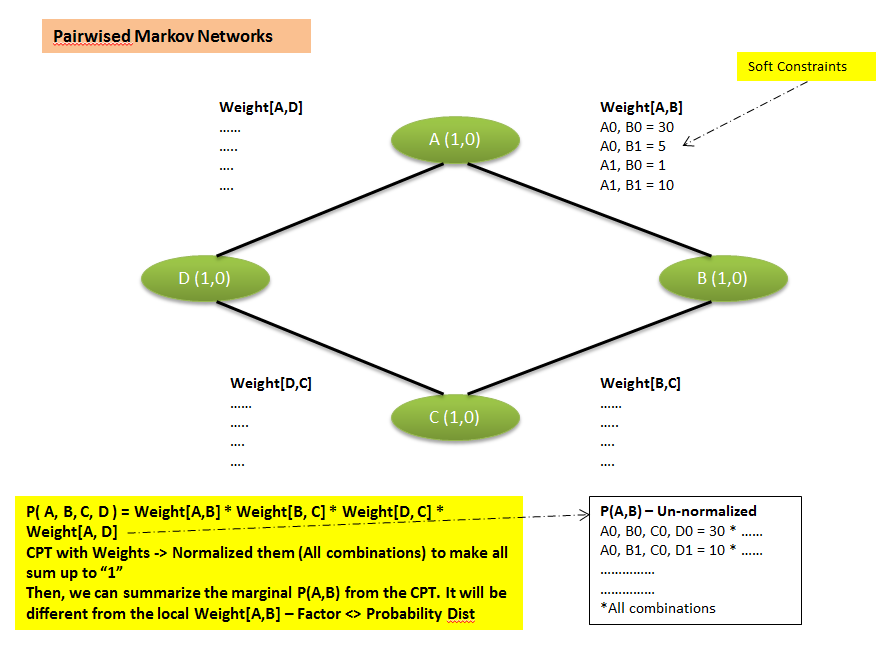


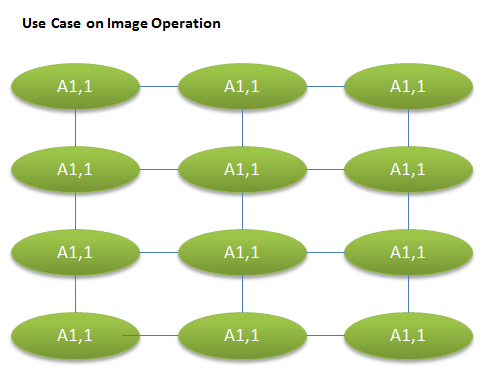


**Markov Networks** – Pairwise Markov Networks

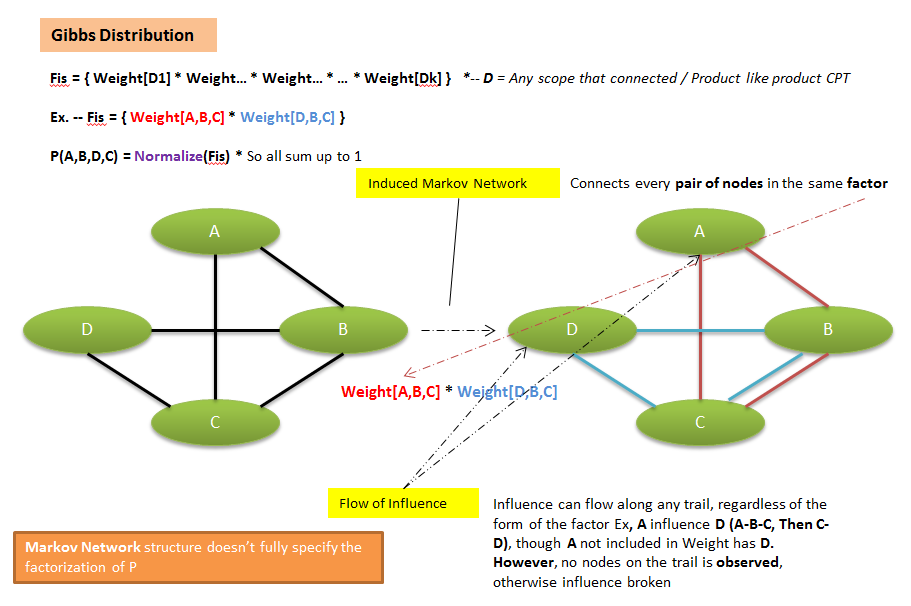
Markov Network (Undirected Graphs)

A pairwised Markov Network is an undirected graph whose nodes are X1,…, Xn and each **edge** = **Xi – Xj** is associated with a **factor =** (potential**) Weight(i) (Xi – Xj)**



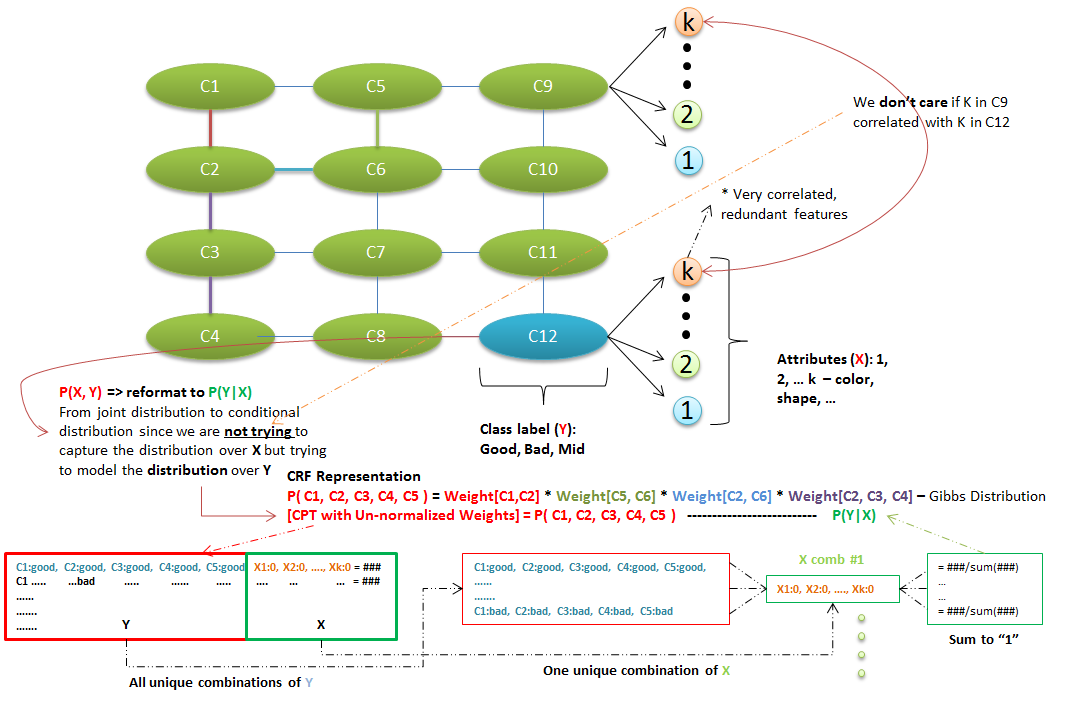


**Markov Networks** – General Gibbs Distribution



**Markov Networks** – Conditional Random Fields (**CRF**)

It is for task specific prediction = [**X**: observation **~ Y**:Prediction]

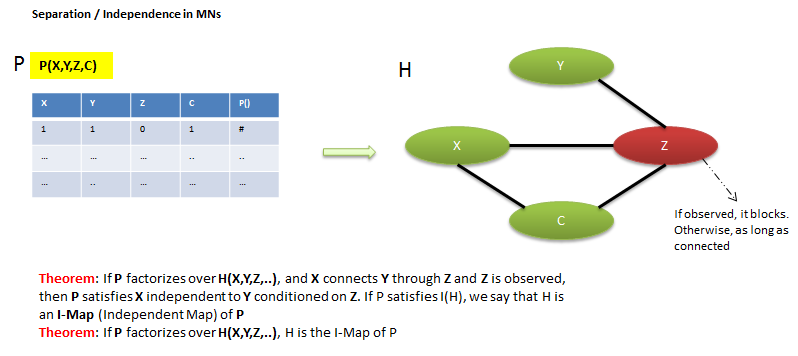


Can apply a layer of another classifier like SVM on the top of predicted labels from CRF to improve performance; Tends to over fit on features. CRF like Gibbs distribution but normalized differently. We don’t need to model the variables we don’t care about. Allows models with highly expressive features without worrying about wrong independencies.

Independencies in Markov Network and Bayesian Networks

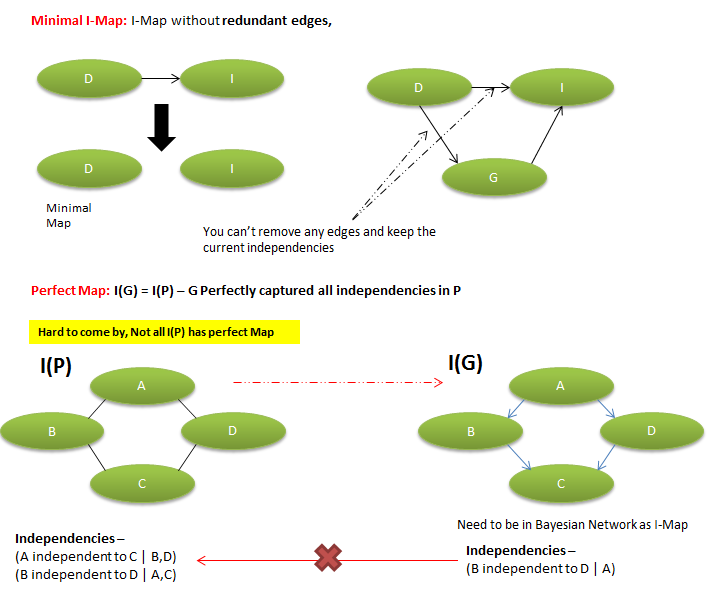
Independencies – Markov Networks

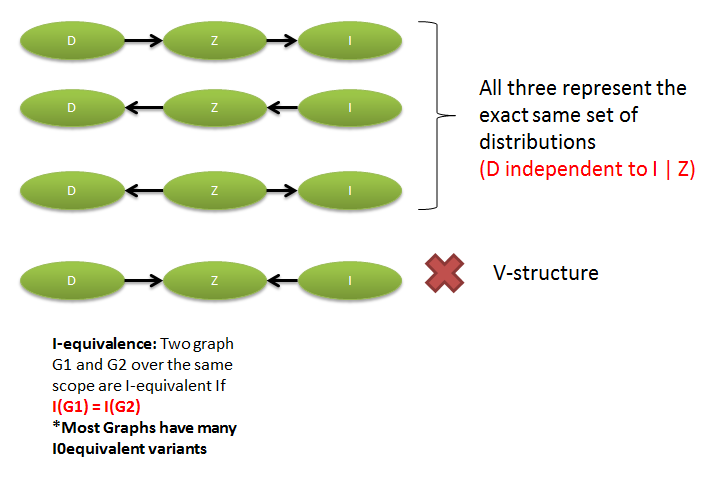
Separation in MNs –



If the graph encodes **more independencies**, it is **sparser** (**fewer parameters**) and **more informative, we preferred.** A **minimal I-Map** may fail to capture a lot of structure even if present and even if its representable. A **perfect Map** is good but may not exists.

Converting BNs 🡪 MNs loses independencies in V-structure; Converting MNs 🡪 BNs must add triangulating edges to loops (A-B-C-D) to connect

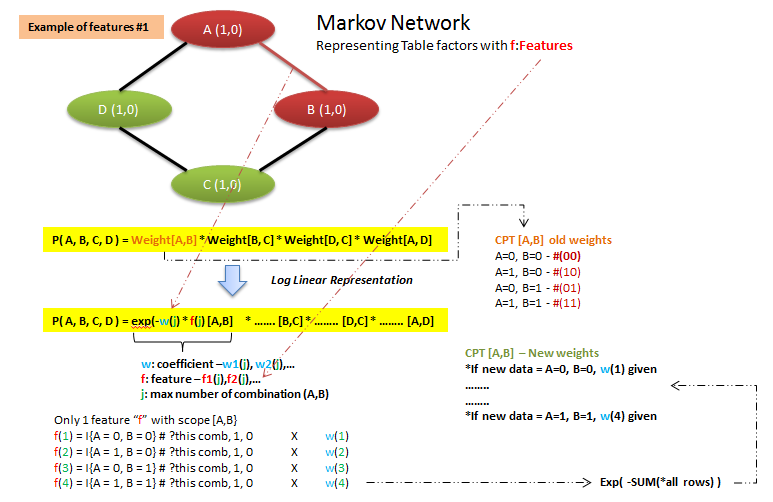


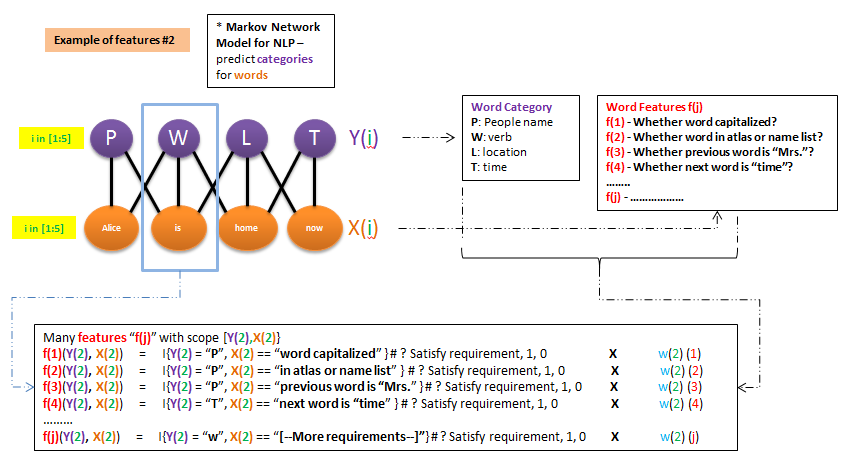


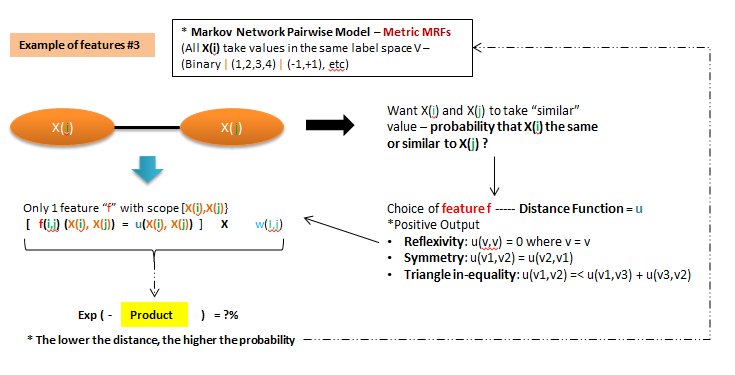
Local Structure in Markov Networks

Local Structure in Markov Network – Log-Linear Models

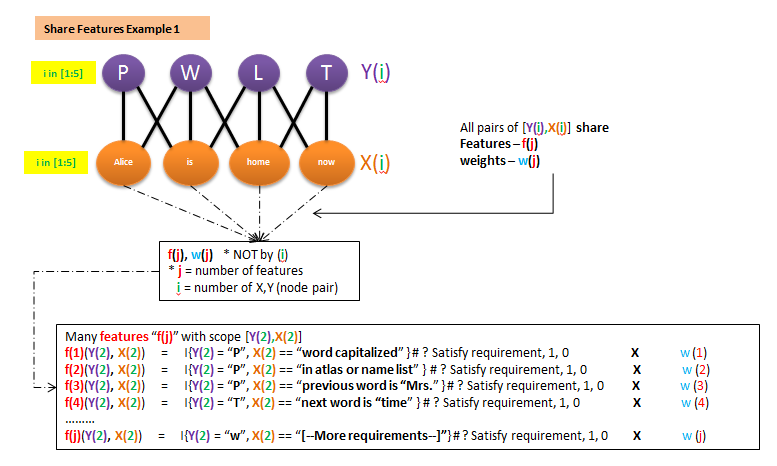
Integrate local structure into undirected Model –





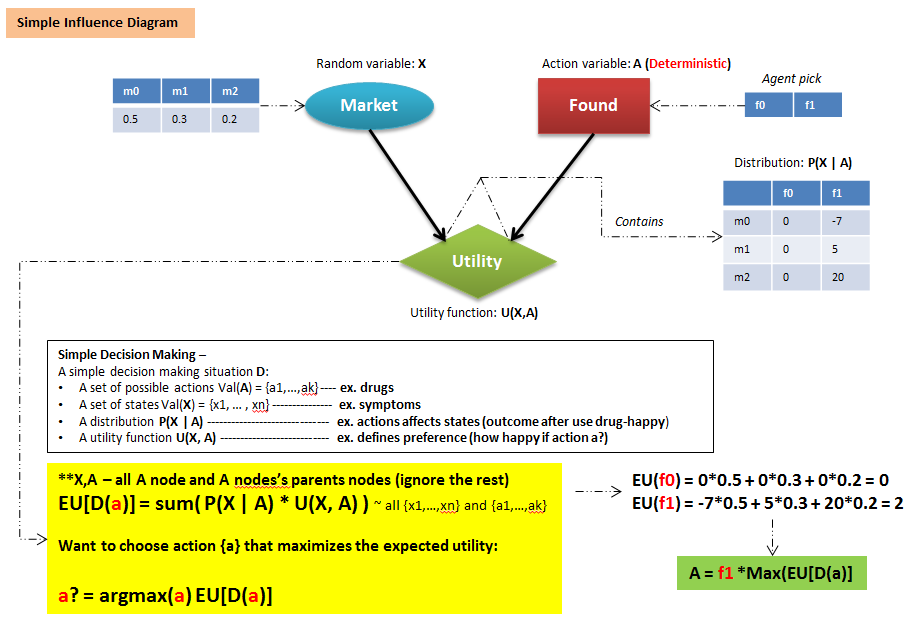


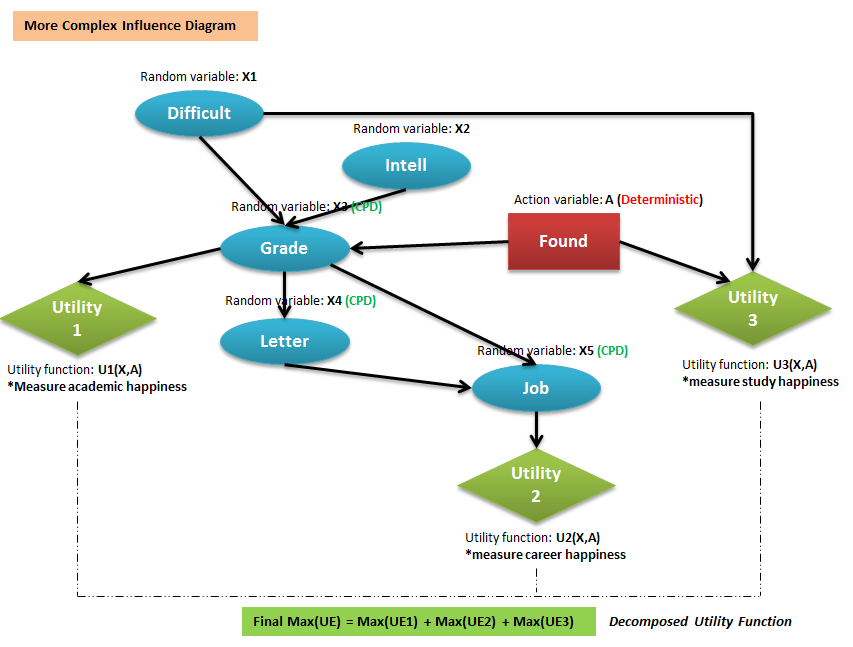
Local Structure in Markov Network – **Shared Features** in Log-linear Model

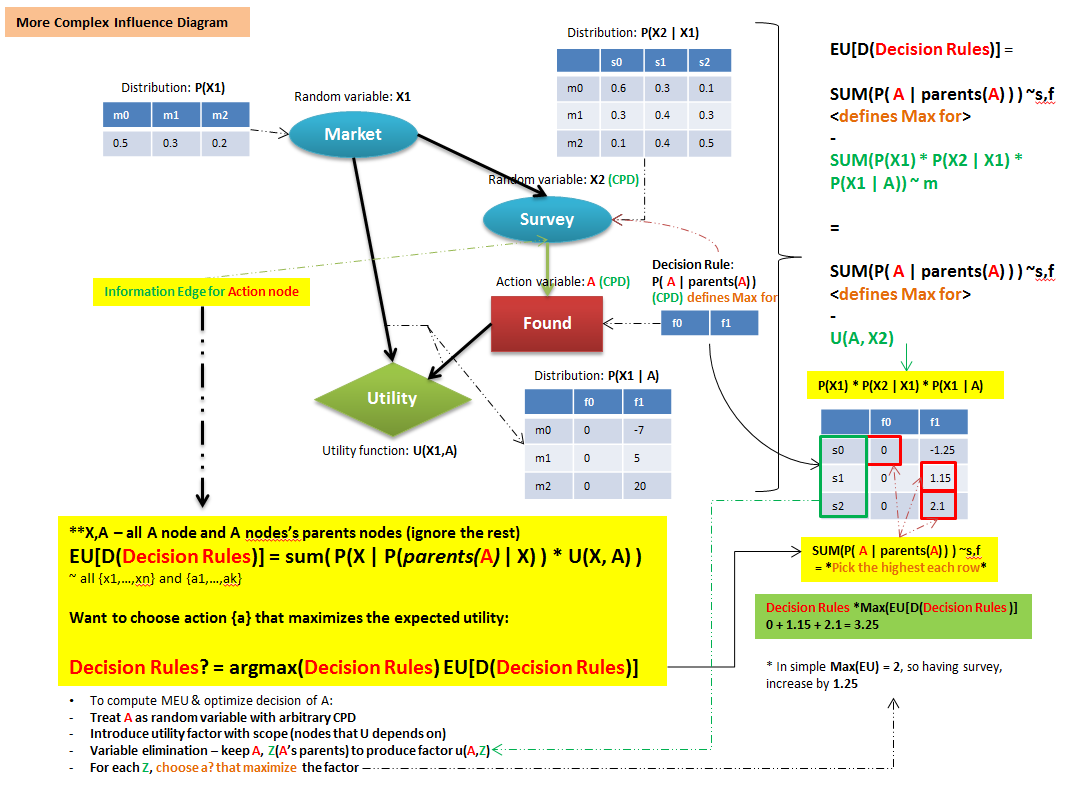


Decision Theory

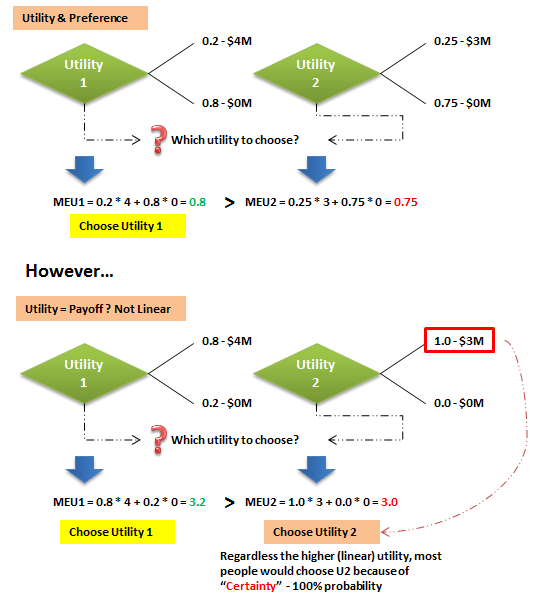
Decision Theory – Maximum Expected Utility

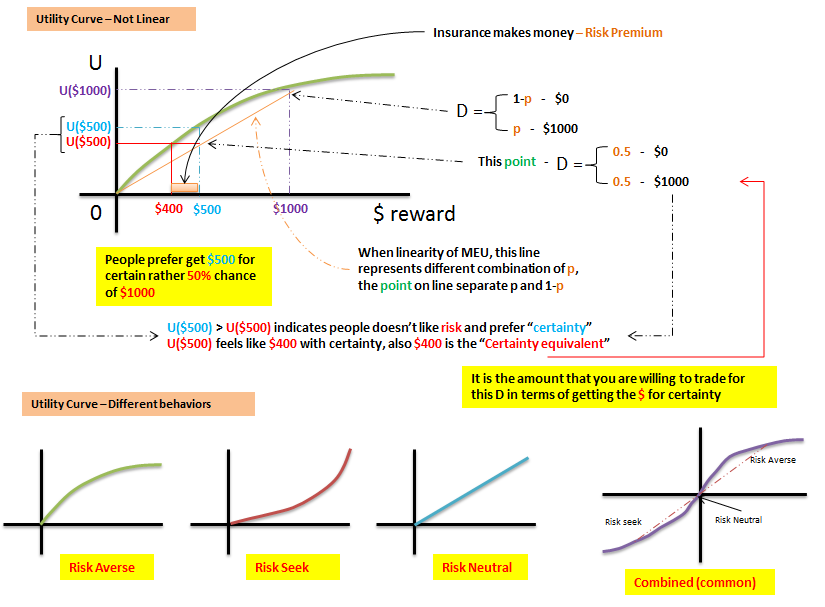


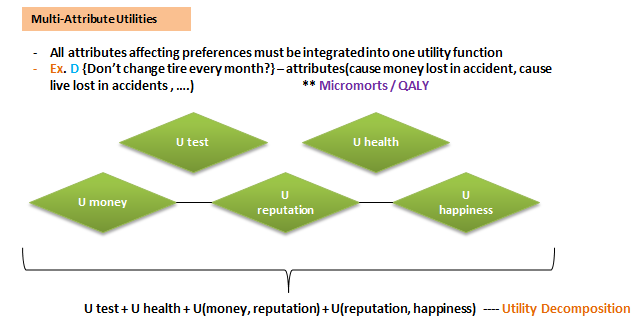




Decision Theory –Utility Function

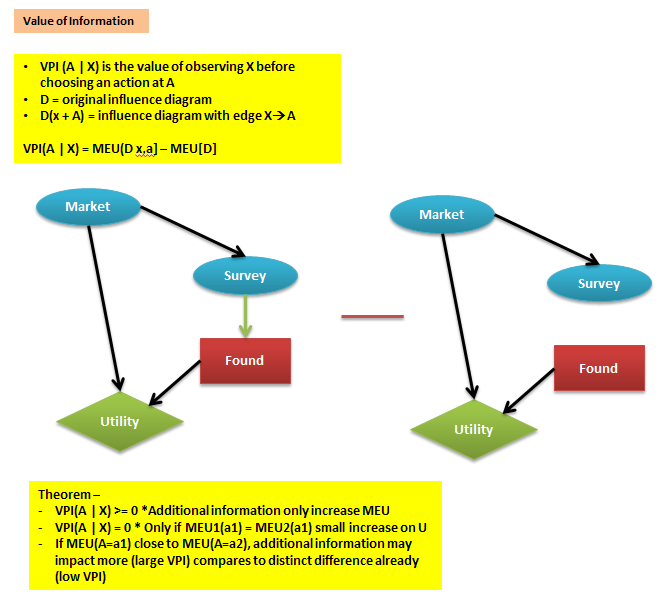






Decision Theory –Value of Perfect Information (VPI)

What features or attributes I should add to the model to improve the performance? Which is worth which is not?



Knowledge Engineering -

How to build a graphic model?

