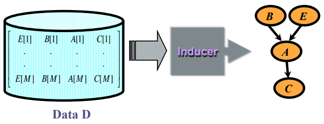
Bayesian Learning

(Ch. 20.1–20.2)

Knowledge-Based Agents

(Ch. 7)



Material from Dr. Marie desJardin,

Quick Bookkeeping

- Today:
  - Tail end of machine learning (for now)
  - Knowledge-based agents and knowledge representation
- Next time:
  - Propositional logic
  - Logical inference
- After that: planning, planning, more planning

Bayesian Learning

- Bayesian probability: the view of probability as a measure of belief, as opposed to being a frequency.
  - Does not mean that past statistics are ignored
  - Statistics of what has happened in the past is the knowledge that is conditioned on and used to update belief.
- **Models** are mathematical formulations of observed events
- **Parameters** are factors in the models affecting observations

Mackworth & Poole Ch. 6

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

Naïve Bayes

- Make the simplest possible independence assumption: Each attribute is independent of the values of the other attributes, given the class variable
  - In restaurants: Cuisine is independent of Patrons, *given* a decision to stay
- Embodied in a belief network where:
  - The features are the nodes
  - Target variable (the classification) has no parents
  - The classification is the only parent of each input feature
- This requires:
  - Probability distributions $P(C)$ for target variable C
  - $P(F_i \mid C)$ for each input feature F_i

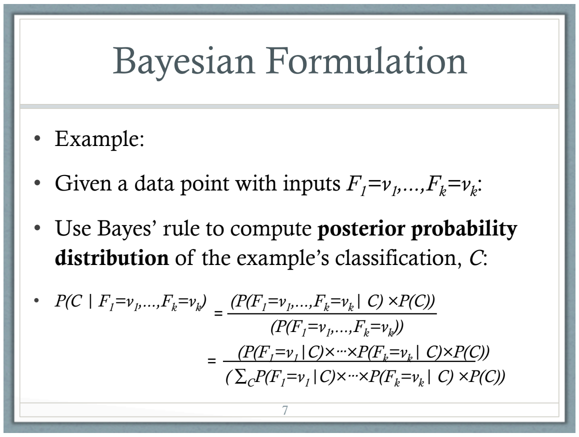
Bayesian Formulation

- **For each example, predict C by conditioning on observed input features and by querying the classification**
- The probability of class C given $F_1, \ldots, F_n$
  $\mathbf{p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)}$
- Denominator: normalizing constant to make probabilities sum to 1, which we call $\boldsymbol{\alpha}$

  $\mathbf{p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)}$
- Denominator does not depend on class
- Therefore, not needed to determine the most likely class

Bayesian Formulation

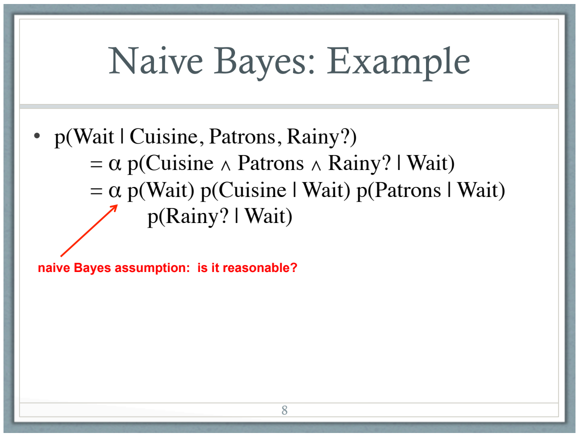
- The probability of class C given $F_1, \ldots, F_n$
  $\mathbf{p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)}$
  $\mathbf{= \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)}$
- Assumption: each feature is conditionally independent of the other features given C. Then:
  $p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, \Pi_i\, p(F_i \mid C)$
- We can estimate each of these conditional probabilities from the observed counts in the training data:
  $p(F_i \mid C) = N(F_i \wedge C) / \mathbf{N}(C)$



Bayesian Formulation

- Example:
- Given a data point with inputs $F_1=v_1,\ldots,F_k=v_k$:
- Use Bayes' rule to compute **posterior probability distribution** of the example's classification, C:
- $P(C \mid F_1=v_1,\ldots,F_k=v_k) = \frac{(P(F_1=v_1,\ldots,F_k=v_k \mid C) \times P(C))}{(P(F_1=v_1,\ldots,F_k=v_k))}$

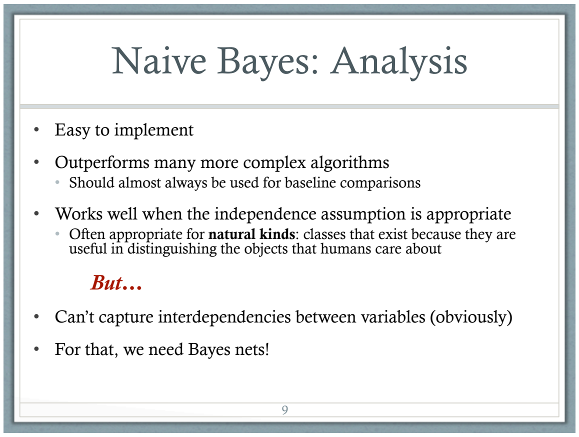
$$= \frac{(P(F_1=v_1 \mid C)\times\cdots\times P(F_k=v_k \mid C)\times P(C))}{(\sum_C P(F_1=v_1 \mid C)\times\cdots\times P(F_k=v_k \mid C) \times P(C))}$$



Naive Bayes: Example

- p(Wait | Cuisine, Patrons, Rainy?)
  = α p(Cuisine ∧ Patrons ∧ Rainy? | Wait)
  = α p(Wait) p(Cuisine | Wait) p(Patrons | Wait)
  p(Rainy? | Wait)

naive Bayes assumption: is it reasonable?

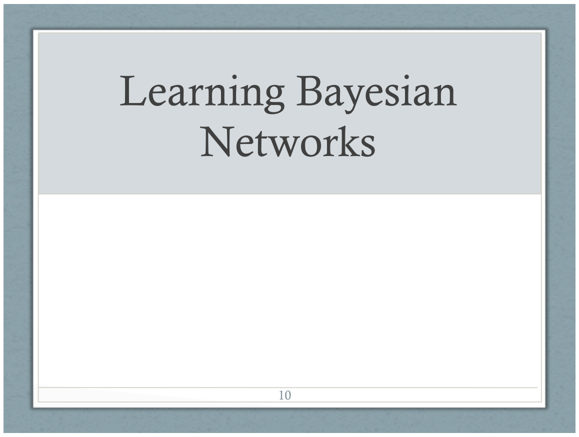


Naive Bayes: Analysis

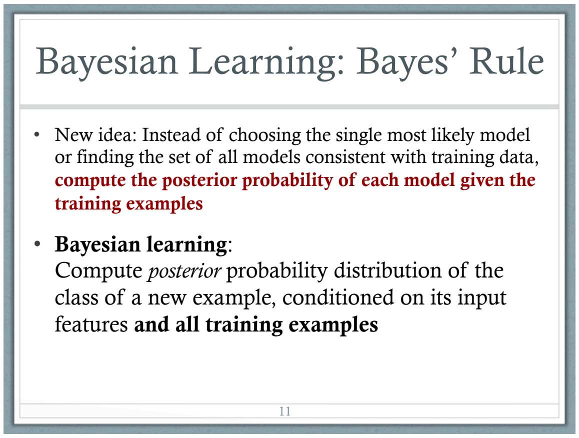
- Easy to implement
- Outperforms many more complex algorithms
  - Should almost always be used for baseline comparisons
- Works well when the independence assumption is appropriate
  - Often appropriate for **natural kinds**: classes that exist because they are useful in distinguishing the objects that humans care about

But…

- Can't capture interdependencies between variables (obviously)
- For that, we need Bayes nets!

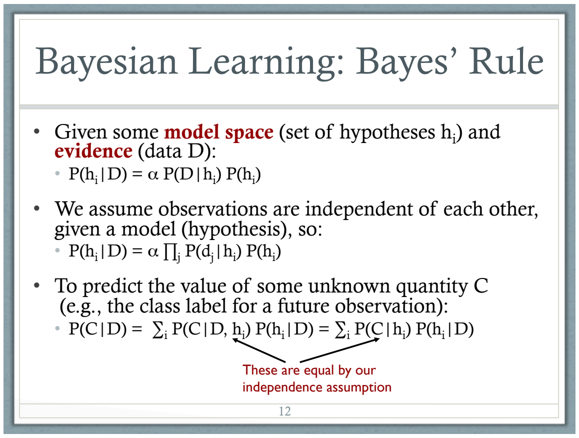


Learning Bayesian Networks



Bayesian Learning: Bayes' Rule

- New idea: Instead of choosing the single most likely model or finding the set of all models consistent with training data, **compute the posterior probability of each model given the training examples**
- **Bayesian learning**:
  Compute *posterior* probability distribution of the class of a new example, conditioned on its input features **and all training examples**



Bayesian Learning: Bayes' Rule

- Given some **model space** (set of hypotheses h_i) and **evidence** (data D):
  - $P(h_i \mid D) = \alpha P(D \mid h_i) P(h_i)$
- We assume observations are independent of each other, given a model (hypothesis), so:
  - $P(h_i \mid D) = \alpha \prod_j P(d_j \mid h_i) P(h_i)$
- To predict the value of some unknown quantity C (e.g., the class label for a future observation):
  - $P(C \mid D) = \sum_i P(C \mid D, h_i) P(h_i \mid D) = \sum_i P(C \mid h_i) P(h_i \mid D)$

These are equal by our independence assumption

Example

- New example has inputs ***X*=*x*** and target features (class variables) ***Y***
- ***e*** is the set of training examples
- Goal: compute **P(*Y*|*X*=*x*∧*e*)**
  - The probability distribution of target variables given the inputs and the examples
- A **model** is assumed to have generated the examples; *M* is set of models
- Then: $P(Y|x\wedge \mathbf{e}) = \sum_{m\in M} P(Y \wedge m \ | x\wedge \mathbf{e})$
  $= \sum_{m\in M} P(Y \mid m \wedge x\wedge \mathbf{e}) \times P(m|x\wedge \mathbf{e})$
  $= \sum_{m\in M} P(Y \mid m \wedge x) \times P(m|\mathbf{e})$
- Bayes' rule: $P(m|\mathbf{e}) = (P(\mathbf{e}|m)\times P(m))/(P(\mathbf{e}))$
- So, **weight of each model** depends on how well it predicts the data and its prior probability

Details: http://artint.info/html/ArtInt_196.html

Bayesian Learning, 3 Ways

- **BMA (Bayesian Model Averaging)**
  - Don't just choose one hypothesis; instead, make predictions based on the weighted average of all hypotheses (or some set of best hypotheses)
- **MAP (Maximum *A Posteriori*) hypothesis**
  - Choose hypothesis with highest *a posteriori* probability, given data
  - **Maximize p(h_i | D)**
  - Generally easier than Bayesian learning
  - Closer to Bayesian prediction as more data arrives
- **MLE (Maximum Likelihood Estimate)**
  - Assume all hypotheses are equally likely *a priori*; best hypothesis maximizes the **likelihood** (i.e., probability of data given hypothesis)
  - **Maximize p(D | h_i)**

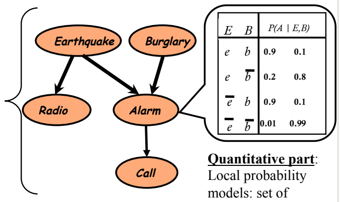
Bayesian Learning

- **BMA (Bayesian Model Averaging) –** average predictions of hypotheses
- **MAP (Maximum *A Posteriori*) hypothesis –** Maximize p(h_i | D)
- **MLE (Maximum Likelihood Estimate) –** Maximize p(D | h_i)
- **MDL (Minimum Description Length) principle:** Use some encoding to model the **complexity** of the hypothesis, and the fit of the data to the hypothesis, then **minimize** the overall description of h_i + D

Quick Review: Bayes Nets

Qualitative part:
statistical independence statements (causality!)

- Directed acyclic graph (DAG)
  - Nodes - **random variables of interest** (exhaustive, mutually exclusive states)
  - Edges - direct (causal) influence



E	B	P(A | E,B)	
e	*b*	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	*b*	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

Quantitative part:
Local probability models: set of conditional probability distributions.

Slide © 1998, Nir Friedman, U.C. Berkeley, and Moises Goldszmidt, SRI International. All rights reserved.

Example: Coin Toss

- **Models** mathematically formulate observed events
- **Parameters** are factors in the models affecting outcomes
- **Toin Coss Example**
  - **Fairness of coin** is the parameter, θ ;
  - **Outcome** of the events is data, D
    - E.g. heads = 72, tails = 28
  - Given an outcome (D), what is the probability this coin is fair (θ =0.5)?
  - Bayes' rule: P(θ |D) = (P(D| θ) × P(θ))/P(D)

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

Example: Coin Toss

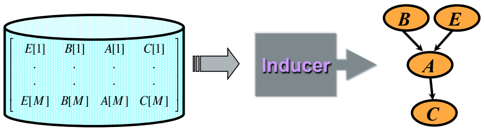
- Bayes : P(θ |D) = (P(D| θ) × P(θ))/P(D)
- **P(θ)** ss of coin befor
  - Car
- **P(D|** distri
  - Prol umber of flips
- **P(D)**
  - Det ues of θ , weighted by how strongly we believe in those particular values of θ
- **P(θ |D) is the posterior**: belief of our parameters after observing the evidence

The point: If we had multiple hypotheses about the fairness of the coin, but didn't know for sure, then this tells us the probability of seeing a certain sequence of flips for each possible fairness.

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

Learning Bayesian Networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1],\ldots,\boldsymbol{x}[\boldsymbol{M}]\}$
- Find B that best matches ***D***
  - model selection
  - parameter estimation



Data D

Parameter Estimation

- Assume known structure
- Goal: estimate BN param
  - entries in local probability models, P(X | Paren
- A good parameterization **q** is **likely** to gen te observed data:

$$\boldsymbol{L}(\theta : \boldsymbol{D}) = \boldsymbol{P}(\boldsymbol{D} \mid \theta) = \prod_{\boldsymbol{m}} \boldsymbol{P}(\boldsymbol{x}[\boldsymbol{m}] \mid \theta)$$

- Maximum Likelihood Estimation (MLE) Principle: Choose $\mathbf{q}^*$ to maximize *L*

i.i.d. samples
independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent

Parameter Estimation II

- The likelihood **decomposes** according to the structure of the network
  → we get a separate estimation task for each parameter
- The MLE (maximum likelihood estimate) solution:
  - for each value *x* of a node *X*
  - and each instantiation ***u*** of *Parents(X)*

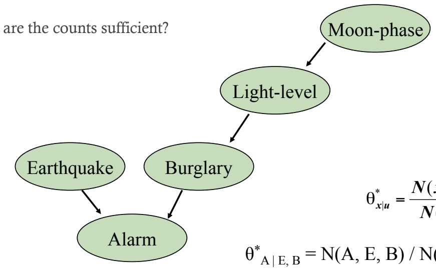
$$\theta^*_{\boldsymbol{x}|\boldsymbol{u}} = \frac{\boldsymbol{N}(\boldsymbol{x},\boldsymbol{u})}{\boldsymbol{N}(\boldsymbol{u})}$$

sufficient statistics

  - Just need to collect the counts for every combination of parents and children observed in the data
  - MLE is equivalent to an assumption of a uniform prior over parameter values

Sufficient Statistics: Example

Why are the counts sufficient?



$$\theta^*_{\boldsymbol{x}|\boldsymbol{u}} = \frac{\boldsymbol{N}(\boldsymbol{x},\boldsymbol{u})}{\boldsymbol{N}(\boldsymbol{u})}$$

$\theta^*_{A \mid E,\, B}$ = N(A, E, B) / N(E, B)

Model Selection

Goal: Select the best network structure, given the data

Input:
- Training data
- Scoring function

Output:
- A network that maximizes the score

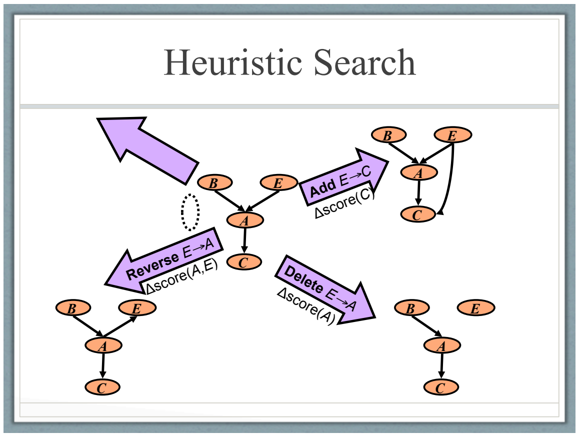
- This is NP-hard!

Structure Selection: Scoring

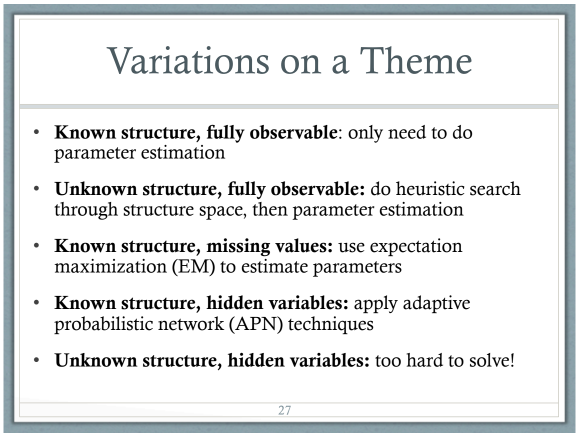
- Bayesian: prior over parameters and structure
- Find balance between model complexity and fit to data
- Score (G:D) = log P(G|D) α log [$\underbrace{\text{P(D|G)}}_{}$ P(G)]

Marginal likelihood ***Prior***

- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

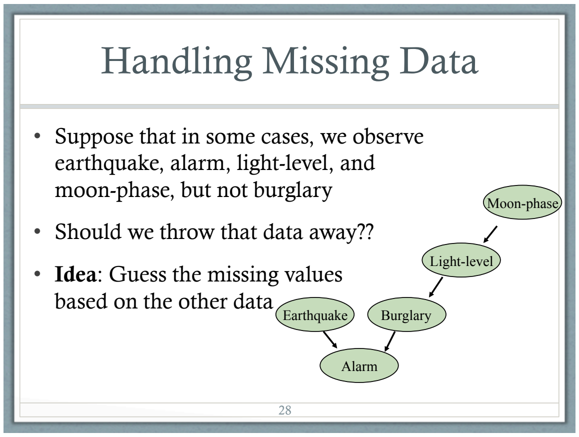


Heuristic Search



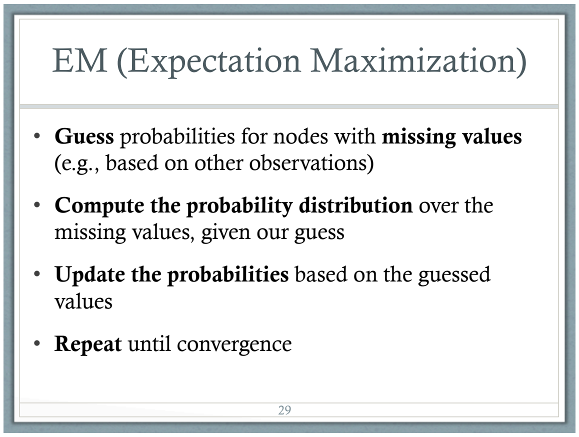
Variations on a Theme

- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!



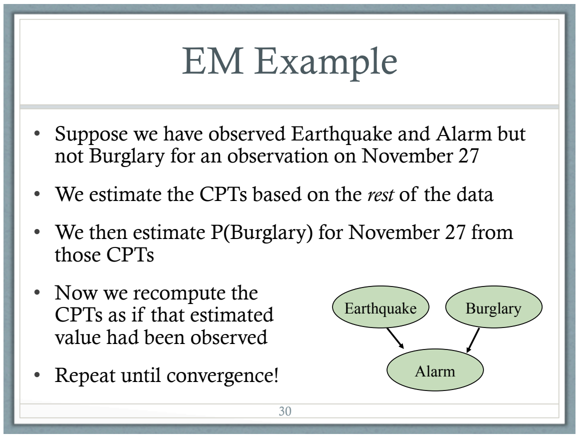
Handling Missing Data

- Suppose that in some cases, we observe earthquake, alarm, light-level, and moon-phase, but not burglary
- Should we throw that data away??
- **Idea**: Guess the missing values based on the other data



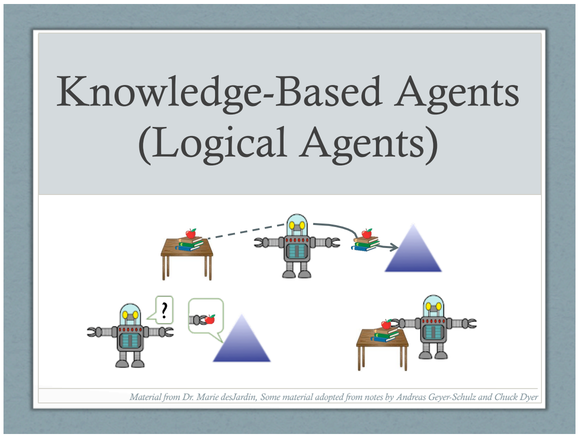
EM (Expectation Maximization)

- **Guess** probabilities for nodes with **missing values** (e.g., based on other observations)
- **Compute the probability distribution** over the missing values, given our guess
- **Update the probabilities** based on the guessed values
- **Repeat** until convergence



EM Example

- Suppose we have observed Earthquake and Alarm but not Burglary for an observation on November 27
- We estimate the CPTs based on the *rest* of the data
- We then estimate P(Burglary) for November 27 from those CPTs
- Now we recompute the CPTs as if that estimated value had been observed
- Repeat until convergence!



Knowledge-Based Agents (Logical Agents)

Material from Dr. Marie desJardin, Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

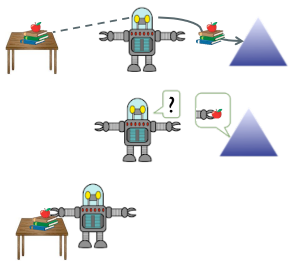
A Knowledge-Based Agent

- A knowledge-based agent needs (at least):
  - A **knowledge base**
  - An **inference system**
- A knowledge base (KB) is a set of representations of facts about the world.
  - Each individual representation is a **sentence** or **assertion**
  - Expressed in a **knowledge representation language**
  - Usually starts with some background knowledge
    - Can be general (world knowledge) or specific (domain language)
- Many existing ideas apply – is it closed-world, etc.

A Knowledge-Based Agent

- Operates as follows:

1. TELLs the knowledge base what it perceives.
2. ASKs the knowledge base what action to perform.
3. Performs the chosen action.



Architecture of a Knowledge-Based Agent

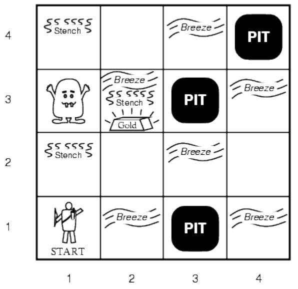
- **Knowledge Level**
  - The most abstract level
  - Describe agent by saying what it knows
    - Example: A taxi agent might know that the Golden Gate Bridge connects San Francisco with the Marin County.
- **Logical Level**
  - Level at which **knowledge** is encoded into **sentences**.
    - Example: Links(GoldenGateBridge, SanFrancisco, MarinCounty)
- **Implementation Level**
  - The physical representation of the sentences in the logical level.
    - Example: '(links goldengatebridge sanfrancisco marincounty)'

The Wumpus World Environment

- The Wumpus computer game
  - Agent explores a cave consisting of rooms connected by passageways.
  - Lurking somewhere in the cave is the Wumpus, a beast that eats any agent that enters its room.
  - Some rooms contain bottomless pits that trap any agent that wanders into the room.
  - Occasionally, there is a heap of gold in a room.
  - The goal is to collect the gold and exit the world without being eaten (or trapped).

A Typical Wumpus World

- The agent always starts in the field [1,1].
- The task of the agent is to find the gold, return to the field [1,1] and climb out of the cave.



Agent in a Wumpus World: Percepts

- Agent perceives
  - **Stench** in the square containing the wumpus and in adjacent squares (not diagonally)
  - **Breeze** in the squares adjacent to a pit
  - **Glitter** in the square where the gold is
  - **Bump**, if it walks into a wall
  - **Woeful** scream everywhere in the cave, if the wumpus is killed
- The percepts are given as a five-symbol list.
- If there is a stench and a breeze, but no glitter, no bump, and no scream, the percept is:

  [Stench, Breeze, None, None, None]
- The agent cannot perceive its own location

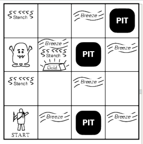
Wumpus Agent Actions

- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab**: Pick up an object that is in the same square as the agent
- **shoot**: Fire an arrow in a straight line in the direction the agent is facing.
  - The arrow continues until it either hits and kills the wumpus or hits the outer wall.
  - The agent has only one arrow, so only the first Shoot action has any effect
- **climb**: leave the cave. This action is only effective in the start square
- **die**: This action automatically happens if the agent enters a square with a pit or a live wumpus

Wumpus Goal

- Agent's goal is to:
  - Find the gold
  - Bring it back to the start square as quickly as possible
  - Don't get killed!
- Scoring
  - 1000 points reward for climbing out with the gold
  - 1 point deducted for every action taken
  - 10000 points penalty for getting killed

Wumpus Agent's First Step



1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

(a)

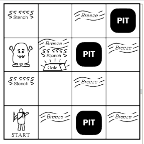
A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P? ¬W	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P? ¬W	4,1

(b)

Percepts: [None, None, None, None, None] Percepts: [None, Breeze, None, None, None]

Later



1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 ¬W ¬P OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P! ¬W	4,1

(a)

A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 ¬W ¬P V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P! ¬W	4,1

(b)

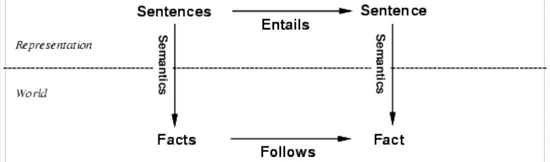
Wumpuses Online

- http://www.cs.berkeley.edu/~russell/code/doc/overview-AGENTS.html
  - Lisp version from Russell & Norvig
- http://www.dreamcodex.com/wumpus.php – Java-based version you can play online
- http://codenautics.com/wumpus/ – Downloadable Mac version

Representation, Reasoning, and Logic

- Point of knowledge representation is to express knowledge in a **computer usable** form
- Needed for agents to act on it (to do well, anyway)
- A knowledge representation language is defined by:
  - **Syntax**: all possible sequences of symbols that form sentences
    - Example: noun referents can be a single word or an adjective-then-noun
  - **Semantics:** facts in the world to which the sentences refer
    - What does it *mean*?
- Each sentence makes a claim about the world
- An agent is said to "believe" a sentence about the world

The Connection Between Sentences and Facts



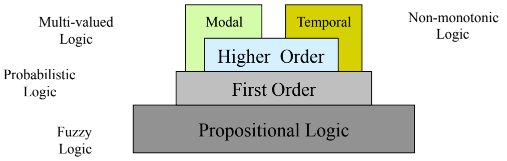
Semantics maps sentences in logic to facts in the world.
The property of one fact following from another is mirrored by the property of one sentence **being entailed** by another.

"Dr M is sick with the flu" ⊨ "Dr M is sick"

Entailment and Derivation

- **Entailment: KB ⊨ Q** x ⊨ y: x semantically entails y
  - Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
  - Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world in which all the premises in KB are true.
- **Derivation: KB ⊢ Q** x ⊢ y: y is provable from x
  - We can derive Q from KB if there is a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

Logic as a KR Language



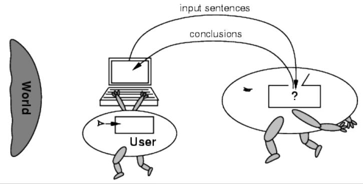
Ontology and Epistemology

- **Ontology** is the study of what there is—an inventory of what exists. An ontological commitment is a commitment to an existence claim.
- **Epistemology** is a major branch of philosophy that concerns the forms, nature, and preconditions of knowledge.

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

No Independent World Access

- The reasoning agent often gets its knowledge about the facts of the world as a *sequence of logical sentences.*
- Must draw conclusions from them without (other) access to the world.
- Thus it is very important that the agent's reasoning is sound!



KB Agents - Summary

- Intelligent agents need **knowledge about the world** for making good decisions.
- The knowledge of an agent is stored in a knowledge base in the form of **sentences** in a **knowledge representation language**.
- A knowledge-based agent needs a **knowledge base** and an **inference mechanism**. It operates by storing sentences in its knowledge base, inferring new sentences with the inference mechanism, and using them to deduce which actions to take.
- A **representation language** is defined by its syntax and semantics, which specify structure of sentences and how they relate to world facts.
- The **interpretation** of a sentence is the fact to which it refers. If this fact is part of the actual world, then the sentence is true.