Background and Motivation

What is learning?
What is machine learning?
How can we specify a learning problem?
Examples of learning algorithms
Representative applications in bioinformatics and computational biology

Computer science is the science of information processing – theory and practice of representation, processing, and use of information. Computer Science offers a powerful paradigm for modeling complex phenomena such as cognition and life, and representing, processing, acquiring, and communicating knowledge that is new in the history of humanity.

Machine Learning

Background – Computer Science
History of computer science is really a history of human attempts to understand nous (the rational mind) — intelligence — processes of acquiring, processing, and using information and knowledge.

- Aristotle (384-322 BC) distinguishes matter from form thereby laying the foundations of representation.
- Panini (350 BC) develops a formal grammar for Sanskrit.
- Al-Khowarizmi (825) introduces algorithms in his text on mathematics.
- Descartes (1556-1650) — *Cogito ergo sum!*
- Hobbs (1650) suggests that thinking is a rule-based process analogous to arithmetic.
- Leibnitz (1646-1716) seeks a general method for reducing all truths to a kind of calculation.
- Boole (1815-1864) proposes logic and probability as the basis of laws of thought.
- Frege (1848-1925) further develops first order logic.
- Tarski (1902-1983) introduces a theory of reference for relating objects in a logic to objects in the world.

- Hilbert (1862-1943) presents the decision problem — is there an effective procedure for determining whether or not a given theorem logically follows from a given set of axioms?
- Godel (1906-1978) shows the existence of an effective procedure to prove any theorem in Frege’s logic and proves the incompleteness theorem.
- Turing (1912-1954) invents the Turing Machine to formalize the notion of an effective procedure.
The road to Computer Science

Church, Kleene, Post, Markov (1930-1950) develop other models of computation based on alternative formalizations of effective procedures. Turing and Church put forth the Church-Turing thesis that Turing machines are universal computers. Several special purpose analog and digital computers are built (including the Atanasoff-Berry Computer). Wiener introduces Cybernetics – the science of communication and control in humans and machines. Shannon (1948) and Wiener develop information theory – offering a means to quantify information.

Several digital computers are constructed and universal languages for programming are developed – Lisp, Snobol, Fortran…


McCarthy, Minsky, Selfridge, Simon, Newell, Uhr et al (1956) begin to investigate the possibility of artificial intelligence. Dantzig and Edmunds (1960-62) introduce reduction – a general transformation from one class of problems to another. Cobham and Edmunds (1964-65) introduce polynomial and exponential complexity. Cook and Karp (1971-72) develop the theory of NP-completeness which helps recognize problems that are intractable. The rest is recent history 😊.
Computer science and artificial intelligence

It is the human urge to understand what makes us human — our ability to perceive, think, reason, learn, and act — that led to the birth of Computer Science.

The language of computation is the best language we have so far for describing how information is encoded, stored, manipulated and used by natural as well as synthetic systems.

Algorithmic or information processing models provide for biological, cognitive, and social sciences what calculus provided for classical physics.

Algorithmic explanations of mind

Computation: Cognition :: Calculus: Physics (Artificial Intelligence, Cognitive Science)
What are the information requirements of learning?
What is the algorithmic basis of learning?
What is the algorithmic basis of rational decision making?
Can we automate scientific discovery?
Can we automate creativity?

Algorithmic explanations of mind

Computation: Cognitive Science :: Calculus: Physics
Computer science offers fundamentally new ways to understand cognitive processes —
Perception
Memory and learning
Reasoning and planning
Rational decisions and problem solving
Communication and language
Behavior
Algorithmic explanations of life

Computation: Life :: Calculus : Physics
(Computational Biology, Computational Ecology ..)
How is information acquired, stored, processed, and used in living systems – in gene expression, protein folding, protein-protein interaction, reproduction?
How do brains process information?
How do genes and environment determine behavior?
What does the genetic program for fetal development look like?

Algorithmic explanations of social phenomena

Computation: Society :: Calculus : Physics
(Computational Economics, Computational Organization Theory)
What is the informational and algorithmic basis of inter-agent interaction, communication, and coordination?
Under what conditions can self-interested rational agents cooperate to achieve a common good?
How do groups and coalitions form?
How do different social organizations (democracies, economies, etc.) differ in terms of how they process information?
Algorithmic explanations of social phenomena

Computation: Social Sciences :: Calculus : Physics

Computer science offers fundamentally new ways to understand economic, and organizational, and social phenomena

Cooperation and competition

Bounded rationality and economic behavior

Community, coalition, and organization formation

Social contract

Organization, interaction, and communication

Rise and fall of cultures e.g., the Anasazi Indians

Conceptual impact of Computer Science

Computer science offers fundamentally new ways to model and understand

cognitive, biological, and social processes

through computational or information processing or algorithmic models

Algorithms as theories

We will have a theory of learning when we have precise information processing models of learning (computer programs that learn from experience)

protein folding when we have an algorithm that accepts a linear sequence of amino acids as input and produces a description of the 3-dimensional structure of a protein as output

bounded rationality when we have an algorithm for rational decision making under limited information, memory or computation
Conceptual impact of Computer Science

Pre-Turing
Focus on physical basis of the universe with the objective of explaining all natural phenomena in terms of physical processes

Post-Turing
Focus on informational and algorithmic basis of the universe with the objective of explaining natural phenomena in terms of processes that acquire, store, process, manipulate, and use information
We understand a phenomenon when we can write a computer program that models it at the desired level of detail
When theories and explanations take the form of algorithms, all science becomes computer science!

Background: Artificial Intelligence

Computation : Cognition :: Calculus: Physics

Algorithms or computation or information processing provide for study of cognition what calculus provided for physics
We have a theory of intelligent behavior when we have precise information processing models (computer programs) that produce such behavior

Why study machine learning?

Practical
Intelligent behavior requires knowledge
Explicitly specifying the knowledge needed for specific tasks is hard, and often infeasible
If we can program computers to learn from experience, we can
• Dramatically enhance the usability of software e.g., personalized information assistants
• Dramatically reduce the cost of software development e.g., for medical diagnosis
• Automate data driven discovery
Why study machine learning?

Scientific
- Information processing models can provide useful insights into
  - How humans and animals learn
  - Information requirements of learning tasks
  - The precise conditions under which certain learning goals are achievable
  - Inherent difficulty of learning tasks
  - How to improve learning – e.g. value of active versus passive learning
  - Computational architectures for learning

Machine Learning in Context

Machine Learning: Applications
- Bioinformatics and Computational Biology
- Cognitive Science
- e-Commerce
- e-Enterprises
- e-Government
- e-Science
- Environmental Informatics
- Human Computer Interaction
- Intelligent Information Infrastructure
- Medical Informatics
- Security Informatics
- Smart Artifacts
- Robotics
- Engineering
### Machine Learning: Contributing Disciplines

**Computer Science** — Artificial Intelligence, Algorithms and Complexity, Databases, Data Mining  
**Statistics** — Statistical Inference, Experiment Design, Exploratory Data Analysis  
**Mathematics** — Abstract Algebra, Logic, Information Theory, Probability Theory  
**Psychology and Neuroscience** — Behavior, Perception, Learning, Memory, Problem solving  
**Philosophy** — Ontology, Epistemology, Philosophy of Mind, Philosophy of Science

### What is learning?

**Learning** = **Inference** + **Memorization**

- **Inference**
  - Deduction: $\forall x \ (\text{Smoke}(x, 1) \implies \text{Fire}(x, 1))$
  - Induction: $\forall x \ (\neg \text{Smoke}(x, 1) \land \text{Fire}(x, 1) \implies \forall y \ (\text{Smoke}(y, 1) \implies \text{Fire}(y, 1)))$
  - Abduction: $\forall x \ (\text{Fire}(x, 1) \implies \text{Smoke}(x, 1))$

### What is Machine Learning?

A program $M$ is said to **learn** from experience $E$ with respect to some class of tasks $T$ and performance measure $P$ if its performance as measured by $P$ on tasks in $T$ in an environment $Z$ improves with experience $E$.

**Example 1**
- $T$ — cancer diagnosis  
- $E$ — a set of diagnosed cases  
- $P$ — accuracy of diagnosis on new cases  
- $Z$ — noisy measurements, occasionally misdiagnosed training cases  
- $M$ — a program that runs on a general purpose computer
What is Machine Learning?

Example 2
- **T** – solving calculus problems
- **E** – practice problems + rules of calculus
- **P** – score on a test

Example 3
- **T** – driving on the interstate
- **E** – a sequence of sensor measurements and driving actions recorded while observing an expert driver
- **P** – mean distance traveled before an error as judged by a human expert

A general framework for learning

Learning = Inference + Memorization

Knowledge Base

Learning Element

Critic

Performance Element

Environment

Data
Types of learning

**Rote Learning** - useful when it is less expensive to store and retrieve some information than to compute it

**Learning from Instruction** - transform instructions into operationally useful knowledge

**Learning from Examples (and counter-examples)** - extract predictive or descriptive regularities from data

**Learning from Deduction (and explanation)** - generalize instances of deductive problem-solving

**Learning from Exploration** - learn to choose actions that maximize reward

Why should Machines Learn?

Some tasks are best specified by example (e.g., credit risk assessment, face recognition)

Some tasks are best shown by demonstration (e.g., landing an airplane)

Buried in large volume of data are useful predictive relationships (data mining)

The operating environment of certain types of software (user characteristics, distribution of problem instances) may not be completely known at design time

Environment changes over time – ability of software to adapt to changes would enhance usability

Examples of Applications of Machine Learning

**Data Mining** –
- Using historical data to improve decisions
  - credit risk assessment, diagnosis, electric power usage prediction
- Using scientific data to acquire knowledge
  - in computational molecular biology
- Software applications that are hard to program
  - autonomous driving, face recognition, speech recognition
- Self-customizing programs
  - newsreader that learns user interests
Course objectives

Understand, implement, and use machine learning algorithms to solve practical problems
Make intelligent choices among learning algorithms for specific applications
Formulate and solve new machine learning problems combining or adapting elements of existing algorithms
Analyze learning algorithms (e.g., performance guarantees) and distinguish between easy and hard learning problems
Gain adequate background to understand current literature
Gain an understanding of the current state of the art in machine learning
Learn to conduct original research in machine learning

Course materials

No required text. Several recommended texts available.
Assigned readings (~50% of the material) from journals, conference proceedings, lecture notes, etc.
Lecture outlines and weekly study guide posted on the course web page: http://www.cs.iastate.edu/~cs573x/
Programming language – Java – Must learn on your own if you do not know it
Software environment – WEKA (Open Source tool for developing machine learning algorithms)

Course mechanics

Assignments
  - Problem sets, reading and writing assignments
  - Laboratory (implementation) exercises
  - Examinations (2)
  - Term project
  - Transcribing lecture notes
Course staff
  - Instructor – Vasant Honavar, Professor of Computer Science
  - TA – Oksana Yakhnenko, Ph.D. student in Computer Science
Office hours etc. See web page for information
Computational Model of Learning

Model of the Learner: Computational capabilities, sensors, effectors, knowledge representation, inference mechanisms, prior knowledge, etc.
Model of the Environment: Tasks to be learned, information sources (teacher, queries, experiments), performance measures
Key questions: Can a learner with a certain structure learn a specified task in a particular environment? Can the learner do so efficiently? If so, how? If not, why not?

To make explicit relevant aspects of the learner and the environment
To identify easy and hard learning problems (and the precise conditions under which they are easy or hard)
To guide the design of learning systems
To shed light on natural learning systems
To help analyze the performance of learning systems

Models of Learning: What are they good for?

To make explicit relevant aspects of the learner and the environment
To identify easy and hard learning problems (and the precise conditions under which they are easy or hard)
To guide the design of learning systems
To shed light on natural learning systems
To help analyze the performance of learning systems

Designing a learning program for a task

Experience – What experiences are available?
- Data – in medical diagnosis, expert diagnosed cases, feedback
  How representative is the experience?

Critic – Can the learner ask questions?
- What type of questions?
- How am I doing? – performance query
- How would you diagnose X? – example based query
- Why was I wrong? – explanation
Designing a learning program

Performance element –
How is the learned knowledge encoded?
– rules, probabilities, programs
How is the learned knowledge used?
– e.g. matching rules
What is the performance measure?
How is performance measured?
– online? batch?

Designing a learning program

Learning element
What is the learning algorithm?
– search for a set of classification rules that are likely to perform well on novel cases (how?)
– estimate a class conditional probability distribution (how?)

Environment
Deterministic or stochastic?
Noisy or noise free?
...

Machine Learning

Learning involves synthesis or adaptation of computational structures
Functions
Logic programs
Rules
Grammars
Probability distributions
Action policies
Behaviors

Machine Learning = (Statistical) Inference + Data Structures + Algorithms
Learning input – output functions

Target function $f$ – unknown to the learner – $f \in F$
Learner’s hypothesis about what $f$ might be – $h \in H$

$H$ – hypothesis space
Instance space – $X$ – domain of $f, h$
Output space – $Y$ – range of $f, h$
Example – an ordered pair $(x, y)$ where $x \in X$ and $f(x) = y \in Y$

$F$ and $H$ may or may not be the same!

Training set $E$ – a multi set of examples
Learning algorithm $L$ – a procedure which given some $E$, outputs an $h \in H$

Learning input – output functions

Must choose
- Hypothesis language
- Instance language
- Semantics associated with both

Machines can learn only functions that have finite descriptions or representations if we require learning programs to be halting programs
Examples: “Tom likes science fiction horror films”
“$F = ma$”

Inductive Learning

Premise – A hypothesis (e.g., a classifier) that is consistent with a sufficiently large number of representative training examples is likely to accurately classify novel instances drawn from the same universe

We can prove that this is an optimal approach (under appropriate assumptions)
Learning input – output functions

Examples of $H$
- the set of all Boolean functions of $n$ inputs
  - DNF – each $h$ is a disjunction of conjunctions (terms)
  - CNF – each $h$ is a conjunction of disjunctions (clauses)
- the set of all conjunctions defined over $n$ literals and their negations
- the set of all disjunctions defined over $n$ literals and their negations
- monotone conjunctions (in which no literals are negated)
- $k$-term DNF – each $h$ is a disjunction of $k$ terms
- $k$-CNF – each $h$ is a conjunction of clauses where each clause has at most $k$ literals (negated or un-negated)
- Boolean threshold functions

Online learning of conjunctive Boolean concepts

Concepts –
  - extensional definition (set of instances)
  - intensional definition (syntactic description)

$\langle X, c(X) \rangle \ldots \rightarrow \text{Learner} \rightarrow c \in C$

A Simple Learning Scenario

Example – Learning Conjunctive Concepts

Given an arbitrary, noise-free sequence of labeled examples $(X_1, c(X_1)), (X_2, c(X_2)), \ldots, (X_m, c(X_m))$ of an unknown binary conjunctive concept $C$ over $\{0, 1\}^n$, the learner's task is to predict $C(X)$ for a given $X$. 
Online learning of conjunctive concepts

Algorithm A.1
Initialize \( L = \{X_1, \neg X_1, \ldots, X_n, \neg X_n\} \)
Predict according to match between an instance and the conjunction of literals in \( L \).
Whenever a mistake is made on a positive example, drop the offending literals from \( L \).

Example
(0111, 1) will result in \( L = \{\neg X_1, X_2, X_3, X_4\} \)
(1110, 1) will yield \( L = \{X_2, X_3\} \)

Mistake bound analysis of conjunctive concept learning

Theorem: Exact online learning of conjunctive concepts can be accomplished with at most \( (N + 1) \) prediction mistakes.

Proof Sketch
No literal in \( C \) is ever eliminated from \( L \).
Each mistake eliminates at least one literal from \( L \).
The first mistake eliminates \( N \) of the \( 2N \) literals.
Conjunctive concepts can be learned with at most \( (N + 1) \) mistakes.

Conclusion: Conjunctive concepts are easy to learn

Learning input – output functions

The size of \( H \) turns out to be important for learning.
\( |H| = 3^n \) when \( H \) is the set of conjunctions.

Exercise – Compute the size of \( H \) in each case (note there is no closed form expression when \( H \) is the set of threshold functions).
Learning and Bias

Example

There is an infinite number of functions that match any finite number of training examples!

Bias free function learning is impossible!

Learning and Bias

Suppose \( H = \text{set of all } n \text{- input Boolean functions} \)

Suppose the learner is unbiased

\[ |H| = 2^{2^n} \]

\( H_v = \text{version space - the subset of } H \text{ not yet ruled out by the learner} \)

\[ \log_2 |H_v| \]

Number of unique examples already seen

Learning and Bias

Weaker bias \( \rightarrow \) more open to experience, flexible

\( \rightarrow \) more expressive hypothesis representation

Occam’s razor

– simpler hypotheses preferred

– Linear fit preferred to quadratic fit assuming both yield relatively good fit over the training examples

With stronger bias, there is less reliance on the training data

Learning in practice requires a tradeoff between complexity of hypothesis and goodness of fit
Learning as Bayesian Inference

Probability is the logic of Science (Jaynes)
Bayesian (subjective) probability provides a basis for updating beliefs based on evidence
By updating beliefs about hypotheses based on data, we can learn about the world.
Bayesian framework provides a sound probabilistic basis for understanding many learning algorithms and designing new algorithms
Bayesian framework provides several practical reasoning and learning algorithms
Hence ... we take a brief detour to visit probability theory and random variables
The world according to Agent Bob

An atomic event or world state is a complete specification of the state of the agent's world.

Event set is a set of mutually exclusive and exhaustive possible world states (relative to an agent's representational commitments and sensing abilities).

From the point of view of an agent Bob who can sense only 3 colors and 2 shapes, the world can be in only one of 6 states.

Atomic events (world states) are mutually exclusive and exhaustive.

Probability as a subjective measure of belief

Suppose there are 3 agents - Adrian, Oksana, Jun, in a world where a dice has been tossed. Adrian observes that the outcome is a “6” and whispers to Oksana that the outcome is “even” but Jun knows nothing about the outcome.

Set of possible mutually exclusive and exhaustive world states = {1, 2, 3, 4, 5, 6}

Set of possible states of the world based on what Oksana knows = {2, 4, 6}

Probability is a measure over all of the world states that are possible, or simply, possible worlds, given what the agent knows.

\[ \text{Possibleworlds}_{Adrian} = \{6\}, \text{Possibleworlds}_{Oksana} = \{2, 4, 6\}, \text{Possibleworlds}_{Jun} = \{1, 2, 3, 4, 5, 6\} \]

\[ \Pr_{Adrian}(\text{worldstate} = 6) = 1 \]
\[ \Pr_{Oksana}(6) = \frac{1}{3} \]
\[ \Pr_{Jun}(6) = \frac{1}{6} \]
Defining Probability – Probability spaces

Definition: Finite Probability Space \((\mathcal{E}, P)\)

Let \(\mathcal{E}\) be a finite event set and let \(P: \mathcal{E} \rightarrow \mathbb{R}^+\) be a function from \(\mathcal{E}\) to non-negative real numbers such that \(\sum_{e \in \mathcal{E}} P(e) = 1\). Then we refer to \(\mathcal{E}\) as an event set and \(P(e)\) as the probability that event \(e\) occurs, or simply, the probability of \(e\).

The elements of \(\mathcal{E}\) are called simple events or elementary events and \(P\) the probability distribution.

Example: \(\mathcal{E} = \{H, T\}; P(H) = P(T) = \frac{1}{2}\)

Probability of compound events

Let \((\mathcal{E}, P)\) be a finite probability space. A compound event \(A\) corresponds to a subset of \(\mathcal{E}\) (a possible world).

\[
P(A) = \sum_{e \in A} P(e)
\]

We say that \(A\) occurs if some \(e \in A\) occurs. Note: Note the "overloading" of the function \(P\).

Fundamental theorem of probability

Let \((\mathcal{E}, P)\) be a finite probability space. Then

a. if \(A \subseteq B \subseteq \mathcal{E}\), \(0 \leq P(A) \leq P(B) \leq 1\)

b. if \(A, B \subseteq \mathcal{E}\), \(P(A \cap B) + P(A \cap \bar{B}) = P(A)\)

c. if \(A, B \subseteq \mathcal{E}\), \(P(A \cup B) = P(A) + P(B) - P(A \cap B)\)

d. if \(A_i \subseteq \mathcal{E}\), \(1 \leq i \leq n\) and \(\forall i \neq j \ A_i \cap A_j = \emptyset\),

\[
P\left(\bigcup_{i=1}^{n} A_i = \sum_{i=1}^{n} P(A_i)\right)
\]
Conditional Probability

Let A, B be events. The conditional probability – the probability that A occurs given B occurs is
\[ P(A|B) = \frac{P(A \cap B)}{P(B)} \]

Example: Suppose I have two coins – one a normal fair coin, and the other with 2 heads. I pick a coin at random and tell you that the side I am looking at is a head. What is the probability that I am looking at a normal coin?

Label the sides h, h' so that the side labeled h corresponds to a head on the normal coin, and the side labeled h' corresponds to a tail on the normal coin and a head on the 2-head coin.

\( n, t \) - normal versus 2-sided
\( \varphi = \{n, t\} \times \{h, h'\} \)

Compound events N, H

\( N = \{(n, h), (n, h')\} \) (selecting the normal coin)
\( H = \{(n, h), (t, h), (t, h')\} \) (selecting a head)

\[ P(N|H) = \frac{P(N \cap H)}{P(H)} = \frac{1/4}{3/4} = 1/3 \]

Conditional probability and Bayes Rule

We abuse notation and use events and the propositions that denote them interchangeably: A can stand for a set or a proposition indicating membership in the set A.

\[ P(A \land B) = P(A|B)P(B) = P(B|A)P(A) \]

\[ P(A_1, \ldots, A_n) = P(A_1)P(A_1|A_2)P(A_2|A_3 \ldots)P(A_n) \]

\[ P(A_1, \ldots, A_n) = P(A_1, A_2, A_3, \ldots, A_n) \]
Bayes Theorem

Does patient have cancer or not?
A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer.

\[
P(\text{cancer}) = \frac{P(+) \cdot P(\text{cancer})}{P(+)} = \frac{P(-) \cdot P(\neg \text{cancer})}{P(-)}
\]

\[
P(\neg \text{cancer}) = \frac{P(-) \cdot P(\text{cancer})}{P(-)} = \frac{P(+) \cdot P(\neg \text{cancer})}{P(+)}
\]

Bayes Theorem

Does patient have cancer or not?

\[
P(\text{cancer}) = 0.008 \quad P(\neg \text{cancer}) = 0.992
\]

\[
P(+ | \text{cancer}) = 0.98 \quad P(+ | \neg \text{cancer}) = 0.02
\]

\[
P(- | \text{cancer}) = 0.97 \quad P(- | \neg \text{cancer}) = 0.03
\]

\[
P(+) = 0.98 \cdot 0.008 + 0.02 \cdot 0.992 = 0.079
\]

\[
P(-) = 0.97 \cdot 0.008 + 0.03 \cdot 0.992 = 0.020
\]

The patient, more likely than not, does not have cancer

Random Variables

A random variable defines a set of compound events which form a partition of the event set.

\[
E = \{(\text{red}, \text{square}), (\text{yellow}, \text{square}), (\text{red}, \text{circle}), (\text{yellow}, \text{circle})\}
\]

The random variable Color with domain \(S = \{\text{red}, \text{yellow}\}\) partitions \(E\) into \(E_{\text{red}}\) and \(E_{\text{yellow}}\).

\[
E_{\text{red}} = \{(\text{red}, \text{square}), (\text{red}, \text{circle})\}, \quad E_{\text{yellow}} = \{(\text{yellow}, \text{square}), (\text{yellow}, \text{circle})\}
\]

\[E = E_{\text{red}} \cup E_{\text{yellow}} \text{ and } E_{\text{red}} \cap E_{\text{yellow}} = \emptyset\]
Random variables

The "domain" of a random variable is the set of values it can take. The values are mutually exclusive and exhaustive. Discrete random variables take values from a countable domain. Consider a random variable Color with domain \{Red, Yellow\}. If \( E = \{ (\text{Red}, \text{Square}), (\text{Yellow}, \text{Circle}), (\text{Red}, \text{Circle}), (\text{Yellow}, \text{Square}) \} \) then the proposition \( \text{Color} = \text{Red} \) is True in the world states \( \{(\text{Red}, \text{Square}), (\text{Red}, \text{Circle})\} \).

Example

Let \( S = \{0, 1\} \) \( E_k \) set of all possible outcomes of \( k \) coin tosses \( \{H, T\}^k \). Let \( X: E \rightarrow S \) be a random variable which has value 0 if the outcome of the \( k \)th coin toss is a \( H \) and 1 otherwise.
Probability of random variables

Let $X$ be a random variable on the probability space $(E, P)$. $X = s$ defines a compound event. The probability that $X = s$, that is, $P(X = s)$ is simply the probability of the corresponding compound event.

Now the function symbol $P$ stands for a class of functions whose domains depend on the “domains” of the random variables involved. By $P(X)$ we mean the function whose domain is the range of $X$ and whose “value at $X = s$” is $P(X = s)$ where

$$P(X = s) = \sum_{s' \in X^{-1}(s)} P(s')$$

Probability Distribution of Random Variables

If $X$ is a random variable with a finite domain, we use $P(X)$ to denote the unconditional probabilities associated with each possible value of $X$.

Example: Domain(Height) = {tall, medium, short}
Domain(Play) = {yes, no}

Joint Distribution $P(Height, Play)$ is a 3×2 table of entries that sum to 1.

Inference using the joint distribution

<table>
<thead>
<tr>
<th></th>
<th>ache</th>
<th>¬ache</th>
</tr>
</thead>
<tbody>
<tr>
<td>cavity</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>¬cavity</td>
<td>0.1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

$$P(cavity) = P(cavity, ache) + P(cavity, ¬ache)$$
**Conditional Probability of Random Variables**

\[ P(X = a \mid Y = b) = P(X = 'a' \mid Y = 'b') \]

\[ P(X \mid Y) \] is a function whose domain is \( \text{Domain}(X) \times \text{Domain}(Y) \)

**Example**

| X | Y | P(X|Y) |
|---|---|--------|
| 0 | 0 | 0.6    |
| 1 | 0 | 0.4    |
| 0 | 1 | 0.3    |
| 1 | 1 | 0.7    |

Note: The sum of entries in a row = 1 (why?)

**Conditional probability and Product rule**

\[ P(A \land B) = P(A \mid B)P(B) = P(B \mid A)P(A) \]

\[ P(A_1, \ldots, A_n) = P(A_1)P(A_1 \mid A_2)P(A_2 \mid A_3, A_1)P(A_3 \mid A_3, A_4, A_1) \ldots P(A_n \mid A_n, \ldots, A_n) \]

In the case of random variables \( X, Y \)

\[ P(X, Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X) \]

denotes a set of equations corresponding to possible assignments of values to the random variables

**Possible worlds, Propositions, and Truth**

A possible world is an assignment of Truth values to every simple proposition about the world. Let \( \Omega \) be a set of possible worlds. Let \( \omega \in \Omega \) and let \( p, q \) be propositions (atomic sentences or syntactically well formed logical formulae). Then \( p \) is True in \( \omega \) (written \( \omega \models p \)) where

\[ \omega \models p \text{ if } \omega \text{ assigns value True to } p \]

\[ \omega \models p \land q \text{ if } \omega \models p \text{ and } \omega \models q \]

\[ \omega \models p \lor q \text{ if } \omega \models p \text{ or } \omega \models q \text{ (or both) } \]

\[ \omega \models \neg p \text{ if } \neg \omega \models p \]
Possible Worlds and Random Variables

A possible world is an assignment of exactly one value to every random variable. Let \( \Omega \) be a set of possible worlds. Let \( \omega \in \Omega \) and let \( f \) be a (logical) formula. Then \( f \) is True in \( \omega \) (written \( \omega \models f \)) where

\[
\begin{align*}
\omega \models X &= v \text{ if } \omega \text{ assigns value } v \text{ to } X \\
\omega \models f \land g &= \text{ if } \omega \models f \text{ and } \omega \models g \\
\omega \models f \lor g &= \text{ if } \omega \models f \text{ or } \omega \models g \text{ (or both)} \\
\omega \models \neg f &= \text{ if } \omega \models \neg f.
\end{align*}
\]

Probability as a Measure over Possible worlds

Associated with each possible world is a measure. When there are only a finite number of possible worlds, the measure of the world \( \omega \), denoted by \( \mu(\omega) \) has the following properties:

\[
\sum_{\omega \in \Omega} \mu(\omega) = 1 \quad \text{for all } \omega \in \Omega.
\]

The probability of a formula \( f \), written as \( P(f) \), is the sum of the measures of the possible words in which \( f \) is True. That is,

\[
P(f) = \sum_{\omega \models f} \mu(\omega).
\]

Conditional probability as a Measure over Possible worlds not ruled out by evidence

A given piece of evidence \( e \) rules out all possible worlds that are incompatible with \( e \) or selects the possible worlds in which \( e \) is True. Evidence \( e \) induces a new measure \( \mu_e \).

\[
\begin{align*}
\mu_e(\omega) &= \begin{cases} 
\frac{1}{P(e)} \mu(\omega) & \text{if } \omega \models e \\
0 & \text{if } \omega \models \neg e
\end{cases} \\
P'(h|e) &= \sum_{\omega : \omega \models e} \mu_e(\omega) = \frac{P(h \land e)}{P(e)}
\end{align*}
\]
Effect of Evidence on Possible worlds

Evidence $z$, e.g., (color = red) rules out some assignments of values to some of the random variables

Evidence redistributes probability mass over possible worlds

A given piece of evidence $z$ rules out all possible worlds that are incompatible with $z$ or selects the possible worlds in which $z$ is True. Evidence $z$ induces a distribution $P_z$.

$$P(e) = \begin{cases} \frac{1}{P(z)} P(e) & \text{if } e \models z \\ 0 & \text{if } e \not\models z \end{cases}$$

Evidence can be generalized to handle vector valued random variables.

Defining probability as a Measure over Possible worlds – infinite sets of variables, continuous random variables

When a random variable take on real values the measure corresponds to a probability density function $p$. The probability that a random variable $X$ takes values between $a$ and $b$ is given by

$$P(a \leq X \leq b) = \int_a^b p(x) \, dx$$

Example:

This definition can be generalized to handle vector valued random variables.
**Axioms of probability**

Let \( f, g \) be logical formulae.
1. \( P(f) = P(g) \) if \( f \) and \( g \) are logically equivalent.
2. \( 0 \leq P(f) \) for any formula \( f \).
3. \( P(\bot) = 1 \) if \( \bot \) is a tautology.
4. \( P(f \lor g) = P(f) + P(g) \) if \( P(f \land g) = 0 \)

Axioms 1-4 are sound and complete with respect to the possible world semantics.

**Why Axioms of probability are reasonable**

Axioms of probability restrict the set of probabilistic beliefs that an agent can hold. Why are these restrictions reasonable?

A logically rational agent cannot simultaneously believe propositions that lead to a logical contradiction—e.g., \( A, B, \) and \( \neg(A \land B) \) because at least one of them must be false in the world.

Why can’t an agent believe the following probabilistic statements about the world?

\[
P(A) = 0.4; P(B) = 0.3; P(A \land B) = 0.0; P(A \lor B) = 0.8
\]

**Why Axioms of probability are reasonable**

<table>
<thead>
<tr>
<th>Agent</th>
<th>Oksana</th>
<th>Outcome for Adrian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposition</td>
<td>Belief</td>
<td>Bet</td>
</tr>
<tr>
<td>( A )</td>
<td>0.4</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( B )</td>
<td>0.3</td>
<td>( \bot )</td>
</tr>
<tr>
<td>( A \lor B )</td>
<td>0.8</td>
<td>( \neg (A \land B) )</td>
</tr>
</tbody>
</table>

Oksana bets that \( A \) will occur $4 against Adrian’s $6 (that \( A \) will not occur). Similarly Oksana bets $3 for \( B \), and $2 on \( \neg (A \lor B) \).

If your beliefs violate the axioms of probability, then \( \exists \) a combination of bets that an adversary (e.g., Oksana) can use to ensure that you (e.g., Adrian) will lose money regardless of the outcome (state of the world).
**Notation**

Let $Y, Z$ denote sets of random variables

- $Y = \{ Y_1, Y_2, Y_3 \}$; $P(Y) = P(Y_1, Y_2, Y_3)$
- $Z = \{ Z_1, Z_2 \}$; $P(Z) = P(Z_1, Z_2)$

$P(Y \cup Z) = P(Y_1, Y_2, Y_3, Z_1, Z_2) = P(Y, Z)$

$P(Y \mid Z) = \frac{P(Y \cup Z)}{P(Z)}$

Note the overloading of $P$ and an unfortunate consequence of the set notation.

---

**Inference using the joint distribution**

<table>
<thead>
<tr>
<th></th>
<th>ache</th>
<th>¬ache</th>
</tr>
</thead>
<tbody>
<tr>
<td>cavity</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>¬cavity</td>
<td>0.1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

$P(\text{cavity}) = P(\text{cavity, ache}) + P(\text{cavity, ¬ache})$

---

**Marginalization**

Let $Y, Z$ denote sets of random variables

- $P(Y) = \sum_z P(Y, z)$ where the summation is over all assignment of values to random variables in $Z$.
- Similarly, $P(Y) = \sum_x P(Y \mid x) P(x)$

Example: $Y = \{ Y_1, Y_2 \}$, $Z = \{ Z_1, Z_2 \}$

Suppose all random variables are binary.

The joint distribution over the variables in $Z \cup Y$ has $2^r$ entries.

Marginalization over $Y$ results in a joint distribution over the variables in $Z$ yielding a table of $2^s$ entries.
Independence and Conditional Independence

Let \((E, P)\) be a probability space. Let \(A_1, A_2 \subseteq E\). We say that the events \(A_1\) and \(A_2\) are independent if

\[
P(A_1 \cap A_2) = P(A_1)P(A_2)
\]

If \(P(A_1) \neq 0\), \(P(A_1 \mid A_2) = P(A_1)\)

\[\Rightarrow A_1\text{ and } A_2\text{ are independent}\]

If \(P(A_1) \neq 0, P(A_2 \mid A_1) = P(A_2)\)

\[\Rightarrow A_1\text{ and } A_2\text{ are independent}\]

If \(P(A_1) = 0\) or \(P(A_2) = 0\) (or both), \(P(A_1 \cap A_2) = 0\)

Independence and Conditional Independence

If for every subset \(B = \{B_1, \ldots, B_n\}\) obtained by selecting \(k\) elements of \(A = \{A_1, \ldots, A_n\}\) \((1 \leq k \leq n)\) if we have

\[
P(B_1, \ldots, B_n \mid C) = \prod_{i=1}^{n} P(B_i \mid C)
\]

we say that \(A_1, \ldots, A_n\) are mutually independent given \(C\).

Independence and Conditional Independence

Let \(Z_1, \ldots, Z_n\) and \(W\) be pairwise disjoint sets of random variables on a given event space. \(Z_1, \ldots, Z_n\) are mutually independent given \(W\) if

\[
P(Z_1, \ldots, Z_n \mid W) = \prod_{i=1}^{n} P(Z_i \mid W)
\]

\[
P(Z_i \mid Z_1 \cup W) = P(Z_i \mid W)\text{ if } Z_i\text{ and } Z_j\text{ are independent.}
\]

Note that these represent sets of equations, for all possible value assignments to random variables.
Independence Properties of Random Variables

Let \( W, X, Y, Z \) be pairwise disjoint sets of random variables on a given event space. Let \( I(W, X, Y, Z) \) denote that \( X \) and \( Z \) are independent given \( Y \). That is, \( P(W \cup Z \mid Y) = P(W \mid Y)P(Z \mid Y) \) or \( P(W \mid Y, Z) = P(W \mid Y) \). Then:

- a. \( I(W, Z, Y) \Rightarrow I(Y, Z, X) \)
- b. \( I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y) \)
- c. \( I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Y) \)
- d. \( I(X, Z, Y) \land I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \)

Proof: Follows from definition of independence.

Expectation and Variance

Let \( X : \mathcal{E} \rightarrow \mathbb{R} \) be a random variable on a finite probability space \((\mathcal{E}, P)\) and \( B \subseteq \mathcal{E} \).

The conditional expectation or expected value of \( X \) given \( B \) is

\[
E(X \mid B) = \sum_{c \in B} P(c \mid B)X(c) = \frac{1}{P(B)} \sum_{c \in \mathcal{E}} P(c)X(c)
\]

The variance of \( X \) given \( B \) is given by

\[
\text{Var}(X \mid B) = E[(X - E(X \mid B))^2 \mid B]
\]

The unconditional expectation and variance correspond to the case \( B = \mathcal{E} \) in which case we simply drop "\mid B".

Conditional expectation of random variables

Expectation of a random variable \( X \) conditioned on a random variable \( Y \) is \( E(X \mid Y) = \sum_{c \in \mathcal{E}} P(c \mid Y)X(c) \).

Note that this denotes a set of equations for possible values of \( Y \).

The definitions can be extended to the case where \( X \) and \( Y \) are replaced by sets of random variables.

Example

<table>
<thead>
<tr>
<th>( P(X=0) ) ( Y=0 )</th>
<th>( P(X=1) ) ( Y=0 )</th>
<th>( P(X=0) ) ( Y=1 )</th>
<th>( P(X=1) ) ( Y=1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.7</td>
</tr>
</tbody>
</table>

\[
E(X \mid Y = 0) = P(X = 0 \mid Y = 0)X(0) + P(X = 1 \mid Y = 0)X(1) = 0.6 \\
E(X \mid Y = 1) = 0.7
\]
Properties of Expectation and Variance

Let $X_1, X_2, \ldots, X_n$ be random variables and $a, b, c_1, \ldots, c_n$ be real numbers.

If $X$ has mean $\mu$ and variance $\sigma^2$,

then $(\alpha X + b)$ has mean $(\alpha \mu + b)$ and variance $\alpha^2 \sigma^2$.

For any $c_i$ and $X_i$,

$$E\left(\sum c_i X_i \mid B\right) = \sum c_i E(X_i \mid B)$$

If $i \neq j$, $X_i$ and $X_j$ are independent given $B$, then

$$\text{Var}\left(\sum c_i X_i \mid B\right) = \sum c_i^2 \text{Var}(X_i \mid B)$$

Proof of these results is left as an exercise.