Scientific Discovery through Explainability

Can Deep Learning solve all our problems?

Professor Ritesh Ajoodha

School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg

ExplainableAI Lab
— MODELLING. DECISION MAKING. CAUSALITY —

Research in Machine Learning

Drug Discovery

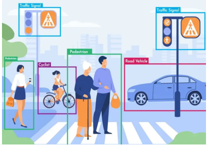
simulate and predict how molecules will behave for effective treatment

Natural Language Processing

sentiment analysis, machine translation, interaction with human language.

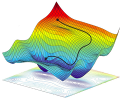
Computer Vision

object detection, facial recognition, and image synthesis



Optimization Algorithms

route planning, scheduling, and resource allocation



Climate Change

optimise energy consumption, cleaner energy sources.

Algorithmic Trading

predict market trends and make trading decisions

Machine Learning in MaSS

Area and Topology

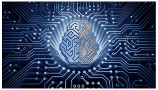
shape recognition and classification, automated theorem proving, geospatial analysis, persistent homology

Graph Theory

graph neural networks, graph embedding, social network analysis, recommendation systems.

Machine Intelligence

transfer learning, self-supervised learning, automated machine learning, online learning



Mod/Analysis Life Science

drug discovery, genomics, medical imaging, protein folding, precision medicine

Mathematical Physics

solving differential equations, quantum computing, simulating physical systems.

Mathematics Education

personalised learning, adaptive assessments, automated grading intelligent tutoring systems

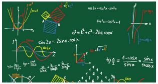
Machine Learning in 12 MaSS

Mathematics in Industry

predictive maintenance, supply chain optimisation, financial modelling, quality control

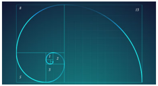
Number Theory

prime number detection, pattern recognition, integer factorisation, theorem proving



Numerical/Applied Mathematics

optimisation, data fitting and regression, parameter estimation, integration, monte-carlo simulation

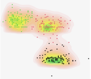


Operator Algebras/Functional Analysis

Solving operator equations, spectral analysis, optimizing functional spaces

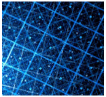
Applied Stats or Theory

data analysis and interpretation, clustering and classification, hypothesis testing, dimensionality reduction.



Symmetry, Mechanics + App

symmetry detection, molecular symmetry analysis, crystallography, optimisation of symmetrical designs

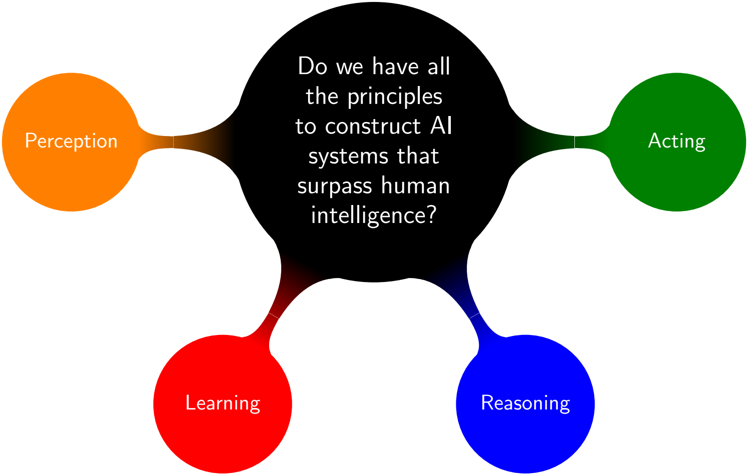


Building an AI System

Do we have all the principles* to construct AI systems that surpass human intelligence?

*fundamental concepts, theories, algorithms, methodologies, heuristics, and techniques

Building an AI System



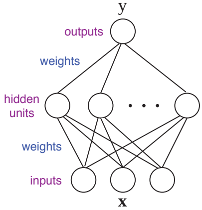
Can Deep Learning Solve All Our Problems?

Neural networks are tunable nonlinear functions with many parameters.

Parameter θ are weights of neural net.

Neural nets model $p(y^{(n)} \mid \mathbf{x}^{(n)}, \theta)$ as a nonlinear function of θ and $\mathbf{x}$, e.g.:

$$p(y^{(n)} = 1 \mid x^{(n)}, \theta) = \sigma(\sum_i \theta_i x_i^{(n)})$$



Multilayer neural networks model the overall function as a composition of functions (layers). e.g.:

$$y^{(n)} = \sum_j \theta_j^{(2)} \sigma(\sum_i \theta_{ji}^{(1)} x_i^{(n)}) + \epsilon^{(n)}$$

Usually trained to maximise likelihood (or penalised likelihood) using variants of stochastic gradient descent (SGD) optimisation

NN = nonlinear function + basic statistics + basic optimisation

Deep Learning

Deep Learning systems are neural network models similar to those popular in the '80s and '90s, with:

1. some architectural and algorithmic innovations (e.g. algorithms to learn with many layers; different forms of regularisation; and new architectures),
2. vastly larger data sets (e.g. web-scale),
3. vastly larger compute resources (e.g. GPU, computing on the cloud, new hardware architectures),
4. much better software tools (e.g. TensorFlow, PyTorch, Keras),
5. vastly increased industry investment and media hype (i.e. more research $\implies$ more progress!)

Limitations of Deep Learning

Neural networks and deep learning systems give great performance on many benchmark tasks but they are generally:

1. very data hungry (i.e. needing millions of training examples),
2. very compute-intensive to train and deploy,
3. poor at representing uncertainty,
4. non-trivial to incorporate prior knowledge and symbolic representations,
5. easily fooled by adversarial examples (e.g. perturbing few pixels causes the neural network to be confidently wrong),
6. finicky to optimise (e.g. choice of architecture, learning procedure, and initialization is seen as a black art),
7. uninterpretable black-boxes (i.e. lack transparency and untrustworthy).

Why do we need probabilities?

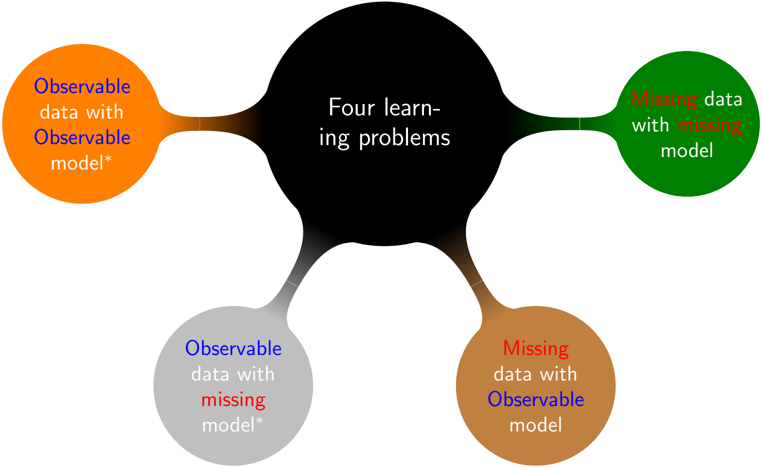
- **Calibrated model and prediction uncertainty**: getting systems that "*know when they don't know*"
- Automatic model **complexity control and structure learning** (Bayesian Occam's Razor, over-fitting?!?)
- Building systems that make **rational decisions**
- As a way of building **prior knowledge** into learning systems, and making sure that knowledge is updated in a coherent and robust way as you get more data
- Provides a **data-driven approach** to model construction
- Make sure that learning works on both **small and large datasets**

Bayesian Philosophy

- In a **Bayesian's world** there are only two things out there: the stuff that you measure and everything else!
- Before we observe the data we have the prior - $P(\texttt{hypo})$.
- For any hypothesis we need to measure how likely the data is under that hypothesis - $P(\texttt{data} \mid \texttt{hypo})$.
- Re-normalise by dividing by all probable hypotheses we are willing to consider - $\sum_{\texttt{hypo}} P(\texttt{hypo})P(\texttt{data} \mid \texttt{hypo})$
- Bayes rule tells us how to do inference about hypotheses (uncertain quantities) from data (measured quantities):

$$P(\texttt{hypo} \mid \texttt{data}) = \frac{P(\texttt{hypo})P(\texttt{data} \mid \texttt{hypo})}{\sum_{\texttt{hypo}} P(\texttt{hypo})P(\texttt{data} \mid \texttt{hypo})}$$

The Learning Problem



The alterations in TCR signaling pathway.

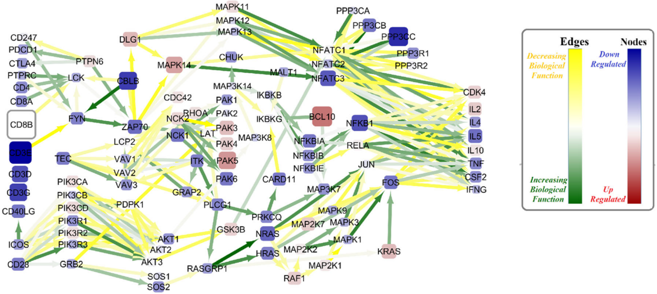
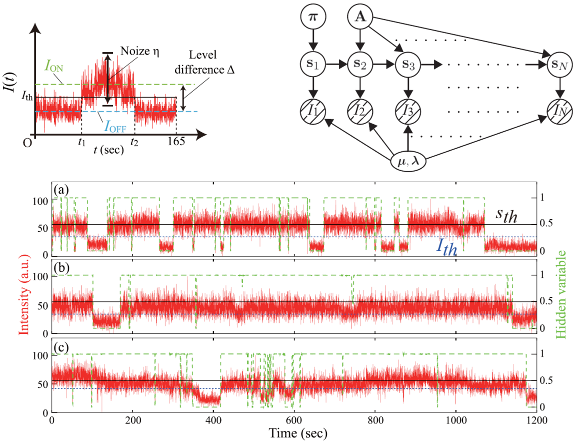


Table 1 Description of the datasets used in the study

No.	GSE no.	GPL/platform	No. of sample		Cell type	Update (year)	Race
			SLE patients	Controls			
1	17755	1291/Hitachisoft	22	55	PBMC	2010	Japanese
2	12374	1291/Hitachisoft	11	6	PBMC	2012	Japanese
3	50772	570/Affymetrix	61	20	PBMC	2015	Unknown[a]
4	81622	10558/Illumina	30	25	PBMC	2016	Unknown[b]
5	121239	13158/Affymetrix	65[c]	20	PBMC	2018	Caucasian/African American[d]
6	126307	13369/Illumina	31	9	PBMC	2019	Several races[e]

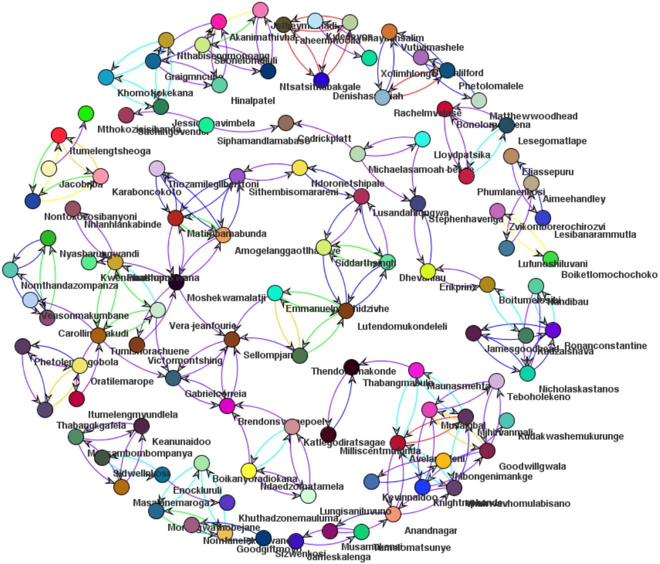
Credit: *Maleknia et al.* "**An integrative Bayesian network approach to highlight key drivers in systemic lupus erythematosus**". *Arthritis Research & Therapy 22 (2020): 1-12*

Time series of fluorescent intensity measured in a wet-nitrogen atmosphere

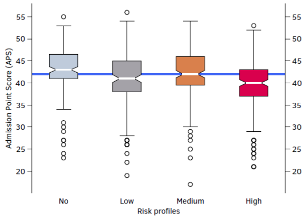


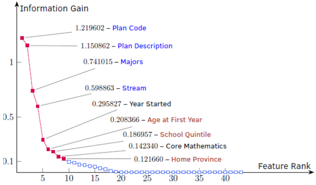
Credit: *Furuta, et al.* "**Hidden Markov model analysis for fluorescent time series of quantum dots**". *Physical Review B 106.10 (2022): 104305*

Using Markov Networks to Detect Plagiarism



Identifying academically vulnerable learners in first-year science programmes





Credit: *Ajoodha.* "**Identifying academically vulnerable learners in first-year science programmes at a South African higher-education institution**". *South African Computer Journal 34.2 (2022): 120-148*

Connectome of repeatedly picked up edges in 100 trials.

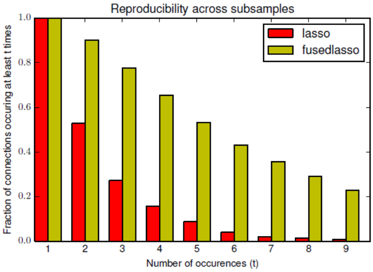


Figure 3: Reproducibility of results from sub-sampling using uncorrected error rate. The fused lasso is much more likely to detect edges and produce stable results. Using corrected p-values no detections are made by lasso (figure in supplementary material).

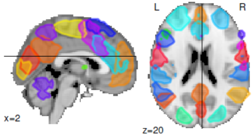


Figure 4: Outlines of the regions of the MSDL atlas.

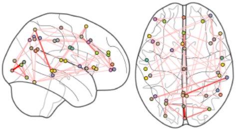
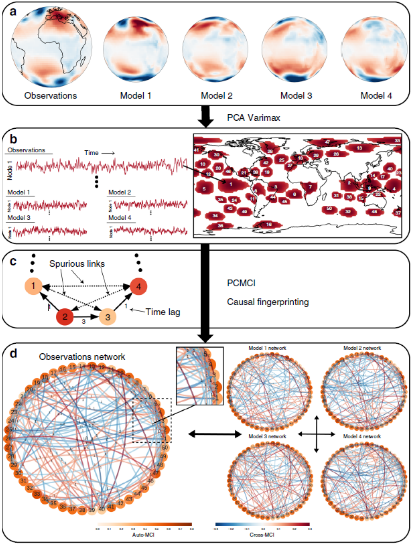


Figure 5: Connectome of repeatedly picked up edges in 100 trials. We only show edges selected more than once. Darker red indicates more frequent selection.

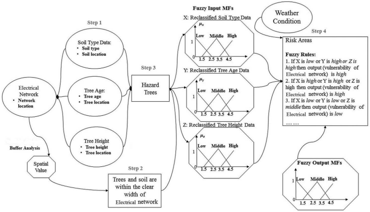
Credit: *Belilovsky et. al.* "**Testing for differences in Gaussian graphical models: Applications to brain connectivity**". *Advances in neural information processing systems 29 (2016)*

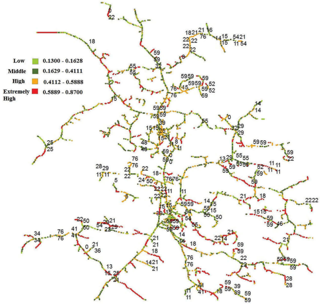
Sketch of the causal model evaluation framework



Nowack, Peer, et al. **Causal networks for climate model evaluation and constrained projections**. *Nature communications* 11.1 (2020): 1415

An example of an SFID modelling tree-related electric outages





Zhang, Zhe, et al. **A spatial fuzzy influence diagram for modelling spatial objects dependencies: A case study on tree-related electric outages.** International Journal of Geographical Information Science 32.2 (2018): 349-366.

Software to implement PGMs

1. **ksvanhorn** (Bayesian statistical Inference)
2. **Bassist** (C++) by U. Helsinki
3. **BNJ** (Java) by Hsu (Kansas)
4. **BNT** (MATLAB) by Murphy
5. **CRFtoolbox** (C and MATLAB) by Schmidt and Murphy
6. **Graphical Model Toolkit** by Bilmes (UW), Zweig (IBM)
7. **B.R.I.O.** (Java and C++) by Ajoodha
8. **PGM_PyLib** (Python) by Serrano-Perez and Sucar
9. **PGMPY** (Python) by Ankan and Panda
10. **CausalNex** (Python)
11. **PyMC3** (Python) by Salvatier, Wiecki, and Fonnesbeck.

Further Reading

Graphical Modelling:

1. Sucar, Luis Enrique. *Probabilistic graphical models.* Advances in Computer Vision and Pattern Recognition. London: Springer London. DOI: 10.978 (2015): 1.
2. Koller, Daphne, and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
3. Murphy, K. *Probabilistic machine learning: Advanced topics.*(2022).

Causality:

4. Peters, Jonas, Dominik Janzing, and Bernhard Schlkopf. *Elements of causal inference: foundations and learning algorithms.* The MIT Press, 2017.
5. Pearl, Judea. *Models, reasoning and inference.* Cambridge, UK: Cambridge University Press 19.2 (2000).

Prior Knowledge Assumed

1. Graduate-level Mathematics (Notation, Complexity, Algebra)
2. Computer science Background (Algorithms, Data Structures)
3. Using a Scientific Calculator (Log, Power, Solving Solving Linear Equations)
4. Programming in Python, Matlab, R, or similar.

Learning Outcomes

1. Design & analyze statistical relationships using PGMs for AI.
2. Apply & evaluate local probability models in context.
3. Develop dynamic Bayesian networks for uncertain AI domains.
4. Design template-based PGMs for object-relational AI.
5. Optimize exact & approximate inference on PGMs.
6. Apply & optimize particle-based inference on PGMs.
7. Use parameter estimation for PGM variable learning.
8. Apply & evaluate structure learning for knowledge discovery.
9. Represent & evaluate causality with graphical models.
10. Use & evaluate influence diagrams for AI decisions.

Mark Breakdown

Weight	**Assessment**	**Notes**
20%	Tutorials	Online weekly on Moodle
20%	Test	Written Assessment
20%	Assignment	Written Report
40%	Exam	Written, Compulsory

Special School Rules

1. Satisfactory Performance (SP) Requirement: attendance and assessment attempts.
2. Exam sub-minimum of 35%
3. The use of external devices such as phones, tablets, or personal laptops is prohibited in assessments

Prescribed Text

1. Koller, Daphne, and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Learning Management System

1. Tutorials and Slides will be on Moodle.
2. The test and exam will be written.

Places and Spaces

Activity	Duration	Day and Time	Venue
Lectures	2×45 minutes	Mondays, 14:15-16:15	LB146
Tutorials	45 minutes	Uploads Mondays 16:15	MSL108
Test	2 hours	24 March 2025, 14:15-16:15	DJ EXAM HALL
Exam	3 hours	Issued by EGO	Issued by EGO
Consultations	1 hour	Mondays, 16:15-17:00	UG 17, MSB

Course Structure

Week	Week Topic	General Topic	Block
1	Scientific Discovery through Explainability	Introduction	Block 1
2	Bayesian Networks	Representation	Block 1
3	Local Probability Models and Template Models	Representation	Block 1
4	Markov Networks and Building PGMs	Representation	Block 1
5	Exact and Approximate Inference	Inference	Block 1
6	TEST WEEK	TEST WEEK	Block 1
	BREAK		
7	Particle-Based Approximate Inference	Inference	Block 2
8	Parameter Estimation	Learning	Block 2
9	Structure Learning	Learning	Block 2
10	Causal Graphical Models	Causality	Block 2
11	Structured Decision Problems	Decision Making	Block 2
12	ASSIGNMENT DUE	ASSIGNMENT DUE	Block 2

Welcome to the Course!

Questions?

WITS UNIVERSITY

ExplainableAI Lab

— MODELLING. DECISION MAKING. CAUSALITY —

Foundations

Professor Ajoodha

AI

Problem Statement

Deep Learning

Probabilistic Modelling

PGMs in Action

Coding

This Course

Foundations

Foundations

Probability and Graph Theory

Professor Ajoodha

School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg

Probability Distributions

Definition

"Probability" refers to the degree of confidence that an event of uncertain nature will occur.

1. The **frequentist interpretation** views probabilities as frequencies of events.
2. The **Bayesian paradigm**:

$$\boxed{P(\alpha \mid \beta)} = \frac{\boxed{P(\beta \mid \alpha)}\, P(\alpha)}{P(\beta)}$$

3. Bayes' Rule is important since it computes the conditional probability from its inverse.
4. If $P(\texttt{cough} \mid \texttt{tuberculosis}) = 0.85$, $P(\texttt{tuberculosis}) = 0.05$ and $P(\texttt{cough}) = 0.6$. Then what is $P(\texttt{tuberculosis} \mid \texttt{cough})$? (ANS = 7%)

Independence and Conditional Independence

Independence:

1. α is independent ($\perp$) to β (in P), if $P(\alpha \mid \beta) = P(\alpha)$.
2. Alternatively $P(\alpha \cap \beta) = P(\alpha)P(\beta)$.
    - Is your class attendance dependent on the time of the class?
    - The outcome of rolling a dice and flipping a coin together?
    - Studying for the PGM test and watching Netflix?

Conditional Independence

3. α is conditionally independence of β given γ (in P), if $P(\alpha \mid \beta \cap \gamma) = P(\alpha \mid \gamma)$.
4. Do we need your metric maths mark to see if you will do well in the PGM course if we know what your MATH III mark is?
$P(\texttt{Pass PGMs} \perp \texttt{Pass Gr12 MATH} \mid \texttt{Pass MATH III})$

Querying a Distribution

Foundations

Professor Ajoodha

AI

Problem Statement

Deep Learning

Probabilistic Modelling

PGMs in Action

Coding

This Course

Foundations

1. In this course we focus on **two types** of queries:
   Probability Query:
2. $P(Y \mid E = e)$. That is, the probability of an event if we have some evidence e.
3. What is the probability of passing the PGM course if we know the student got 70% for the first test?
   $P(\texttt{Pass PGMs} \mid \texttt{Test 1 = 70\%})$?
   MAP Estimation:
4. Finding a joint assignment which occurs with a high probability. Let $\mathbf{W} = \mathcal{X} - \mathbf{E}$ and $\mathbf{Y}$ is a subset of variables.

$$\texttt{MAP}(\mathbf{W} \mid e) = \operatorname*{argmax}_{\mathbf{w}} P(\mathbf{w}, \mathbf{e})$$

$$\texttt{Marginal MAP}(\mathbf{Y} \mid e) = \operatorname*{argmax}_{\mathbf{Y}} \sum_{\mathbf{Z}} P(\mathbf{Y}, \mathbf{Z} \mid \mathbf{e})$$

Probability Density Functions

1. $p : \mathbb{R} \to \mathbb{R}$ is a **probability density function** (PDF) for $\mathcal{X}$ if it is a non-negative integrable function such that:

$$\int_{Val(X)} p(x)\, dx = 1.$$

2. X has a **uniform distribution**: $X \sim Unif[a, b]$ if it has the PDF:

$$p(x) = \begin{cases} \frac{1}{b-a} & b \geq x \geq a \\ 0 & \text{otherwise.} \end{cases}$$

3. X has a **Gaussian distribution**: $X \sim \mathcal{N}(\mu; \sigma^2)$ if it has the PDF:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Joint Density Function

Let P be a joint distribution over $X_1, \ldots, X_n$. A function $p(x_1, \ldots, x_n)$ is a joint density function of $X_1, \ldots, X_n$ if:

1. $p(x_1, \ldots, x_n) \geq 0 \; \forall x_1, \ldots, x_n \in X_1, \ldots, X_n$.
2. p is integratable.
3. For any choice of $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$:

$$P(a_1 \leq X_1 \leq b_1, \ldots, a_n \leq X_n \leq b_n)$$

$$= \int_{a_1}^{b_1} \ldots \int_{a_n}^{b_n} p(x_1, \ldots, x_n)\, dx_1 \ldots dx_n$$

How would you go about conditioning over the event: $x - \epsilon \leq X \leq x + \epsilon$?

Conditional Density Function

Suppose you would like to condition over the event: $x - \epsilon \leq X \leq x + \epsilon$

$$P(Y \mid x) = \lim_{\epsilon \to 0} P(Y \mid x - \epsilon \leq X \leq x + \epsilon).$$

If there is a continuous joint density function p(x,y) then:

$= P(a \leq Y \leq b \mid x - \epsilon \leq X \leq x + \epsilon)$

$= \frac{P(a \leq Y \leq b, x - \epsilon \leq X \leq x + \epsilon)}{P(x - \epsilon \leq X \leq x + \epsilon)}$

$= \frac{\int_a^b \int_{x-\epsilon}^{x+\epsilon} p(x', y)\, dy dx'}{\int_{x-\epsilon}^{x+\epsilon} p(x')\, dx'}$

Expectation

Let X be a discrete random variable that takes numerical values; then the **expectation (mean value)** of X under P is:

$$\mathbb{E}_P[X] = \sum_x x.P(x).$$

If X is continuous then the expectation of X is:

$$\mathbb{E}_P[X] = \int x.p(x)\, dx.$$

Linearity of expectation:

$$\mathbb{E}_P[X+Y] = \mathbb{E}_P[X] + \mathbb{E}_P[Y].$$

Conditional expectation:

$$\mathbb{E}_P[X \mid \mathbf{y}] = \sum_x x.P(x \mid \mathbf{y}).$$

Variance

The deviation from the mean, so called variance of X **(spread of X values around the expected value)**, is calculated as follows:

$$\mathbb{V}ar_P[X] = \mathbb{E}_P[(X - \mathbb{E}_P[X])^2].$$

Variance is the squared difference between X and its expected value.

The standard deviation can be calculated as:

$$\sigma_X = \sqrt{\mathbb{V}ar_P[X]}.$$

σ_X provides a normalised measure of distance from the expected value of X.

If X is a Gaussian distribution $X \sim \mathcal{N}(\mu; \sigma^2)$, then $\mathbb{E}[X] = \mu$ and $\mathbb{V}ar[X] = \sigma^2$.

Graph Theory

A **graph is a data structure** $\mathcal{K} = (\mathcal{X}, \mathcal{E})$ consisting of a set of nodes, denoted $\mathcal{X} = X_1, \ldots, X_n$, and edges, denoted $\mathcal{E}$.

$\mathcal{E}$ contains a set of pairs, where each pair can be $\forall X_i, X_j \in \mathcal{X}, i < j$:

$X_i \rightarrow X_j$ $\quad$ $X_i \leftarrow X_j$ $\quad$ $X_i - X_j$

Directed Graphs (denoted $\mathcal{G}$): All edges are of the form: $X_i \rightarrow X_j$

Undirected Graphs (denoted $\mathcal{H}$): All edges are of the form: $X_i - X_j$

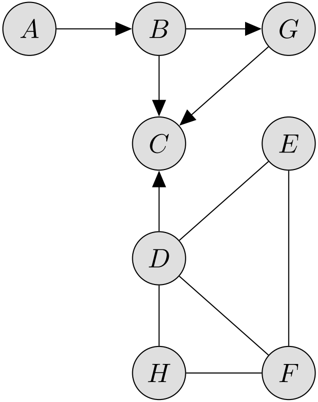
If $X_i \rightarrow X_j$, then X_j is the **child** of **parent** X_i.

Pa_X, Ch_X, and Nb_X (undirected) denotes the parent, children, and neighbours of X respectively.

$Boundary_X = Pa_X \cup Nb_X$

Example

- Parent$_C$ $= \{B, G, D\}$
- Child$_C$ $= \emptyset$, In-degree$_B$ $= 1$
- Neighbour$_F$ $= \{E, D, H\}$
- Boundary$_X$ $= Pa_X \cup Nb_X$
- Degree$_\mathcal{K}$ $= 3$
- Path $= \{E, F, D, C\}$
- Trail $= \{A, B, G, C, D, H\}$
- $\mathcal{K}$ is **connected** since there is a trail between every variable.
- Decedents$_A$ $= \{B, G, C\}$
- Ancestors$_D$ $= \{H, F, E\}$
- NonDecedents$_G$ $= \{A, B\}$



Polytrees and Forests

1. It is difficult to define a probability distribution over a cyclic graph, which is why this course will mostly consider graphs which are not cyclic.
2. In this course we will mostly consider Directed Acyclic Graphs (DAGs).
3. A loop in $\mathcal{K}$ is a trail $X_1, \ldots, X_k$, where $X_1 = X_k$.
4. **Singly connected graphs** contain no loops.
5. Singly connected **directed** graph is called a **polytree**.
6. Singly connected **undirected** graph is called a **forest**.
7. A directed graph is a **directed forest** if each node has at most one parent.
8. A fully connected (directed/undirected) forest is called a tree.