Lecture 1: Introduction & Preliminaries

Thibault Randrianarisoa

University of Toronto, Winter 2026

January 6, 2026

Overview of the lecture

- First part: Organization of the course
  - What is this class about?
  - Administration details (Assessment, Calendar)
- Second part: Overview of probabilistic models
  - History and context
  - MLE
  - Exponential families & Sufficient statistics

Part 1: Course presentation

Learning Outcomes: Today

- Know what topics are and aren't in the course.
- An idea of if you have the background + how hard the material will be.
- What you should be able to do with this knowledge.
- Basics of probabilistics models

What is this class about?

Motivating Questions:

- How to fill in missing parts, conditioned on a description?

"zebras roaming in the field"

"a girl hugging a corgi on a pedestal"

GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Motivating Questions:

- How to generate images from text?

"a high-quality oil painting of a psychedelic hamster dragon"

"robots meditating in a vipassana retreat"

GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Motivating Questions:

- Which measurement would give the most useful information about a patient?
- How can I handle missing data?
- How can I figure out how good a player is from their wins and losses?
- How to combine all information automatically without designing a new algorithm for each type?

Answer:

- **Fit joint distributions** over pixels / words / measurements
  - Answer queries by marginalizing and conditioning

- Only need to:
  - Specify **meaningful models** (graphical models)
  - **Integrate over states** (approximate inference)
  - **Fit lots of parameters** (gradients and neural nets)

Tools of the trade: Probability

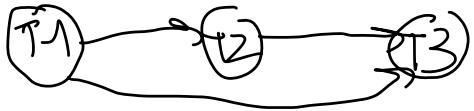
- Probabilities represent uncertainty about a fixed but unknown quantity, conditioned on some information.

- Inference and prediction is easy!:
  "Just write down the joint probability of everything, and integrate out everything you don't know." - MacKay

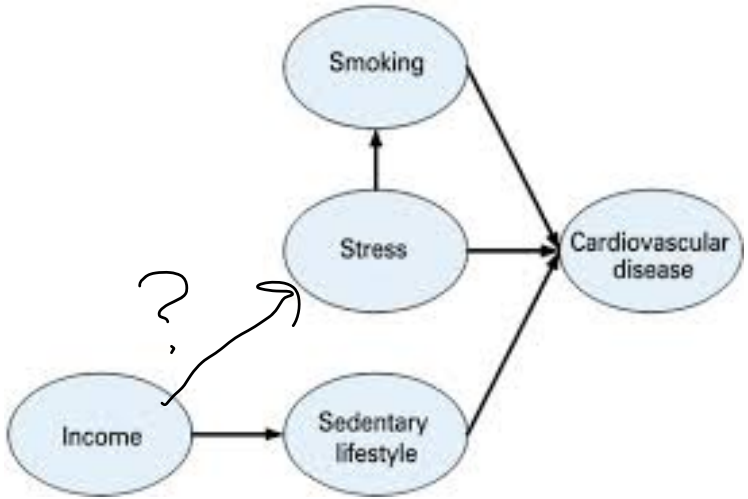
- No need to pretend to identify parameters, except for computational efficiency.

+ Decision Theory

Tools of the trade: Graphical Models



- Large joint distributions are expensive to reason about.
- Conditional independence avoids combinatorial explosions when summing.
- Can reason about conditional independence using graphs.
- Old-fashioned, since often simpler to assume everything is connected.



$$p(i, sm, st, se, c) = p(i)p(se \mid i)p(s)p(sm \mid s)p(c \mid sm, se, st)$$

Tools of the trade: Neural Networks

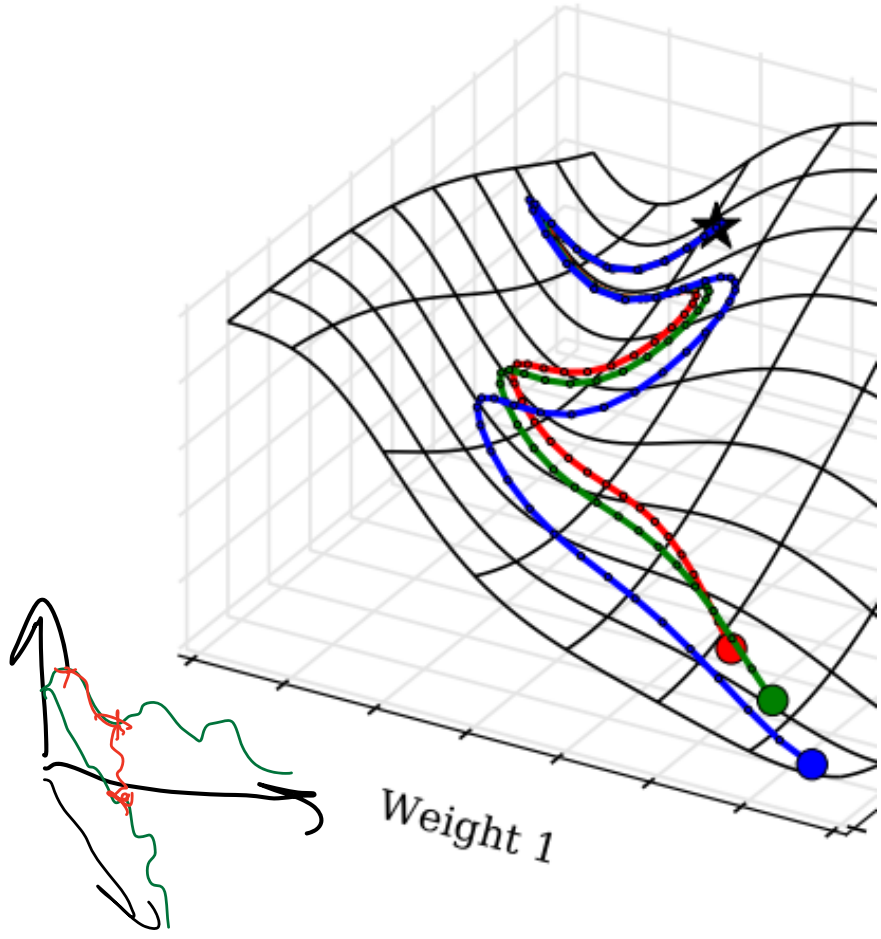
- Not profound or especially mysterious: Just a large nonlinear parametric function.
- Can fit anything if we overparameterize enough and use gradients.
- Main issues: Overfitting, non-differentiable objectives, hard to debug.

Source: xkcd

Tools of the trade: Gradient-based optimization

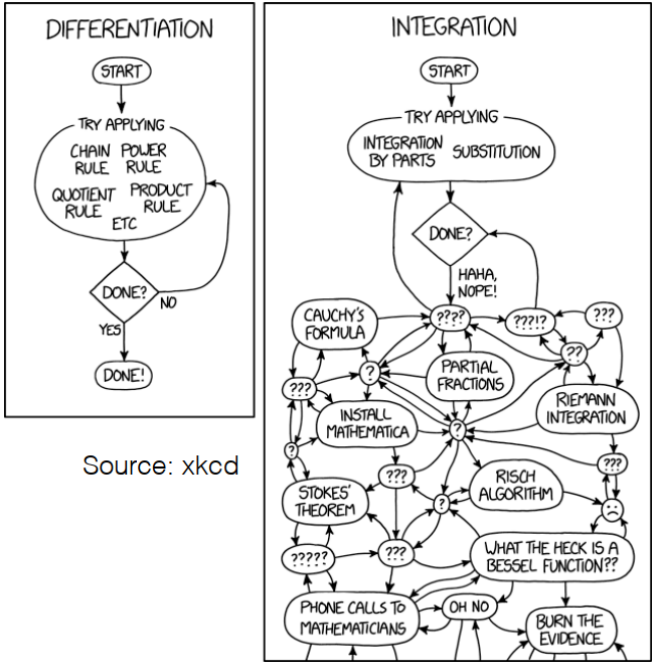
- Unconstrained, high-dimensional, stochastic, first-order gradient descent is surprisingly applicable.
- Hinton: SGD "works much better than anyone had any right to expect".
- More parameters → more progress before getting stuck.

• automatic differentiation



Tools of the trade: Automatic Differentiation

- Reverse-mode grads have same asymptotic time cost as original function.
- Biggest change in last 10 years of ML practice.
- Vector-Jacobian products are cheap.



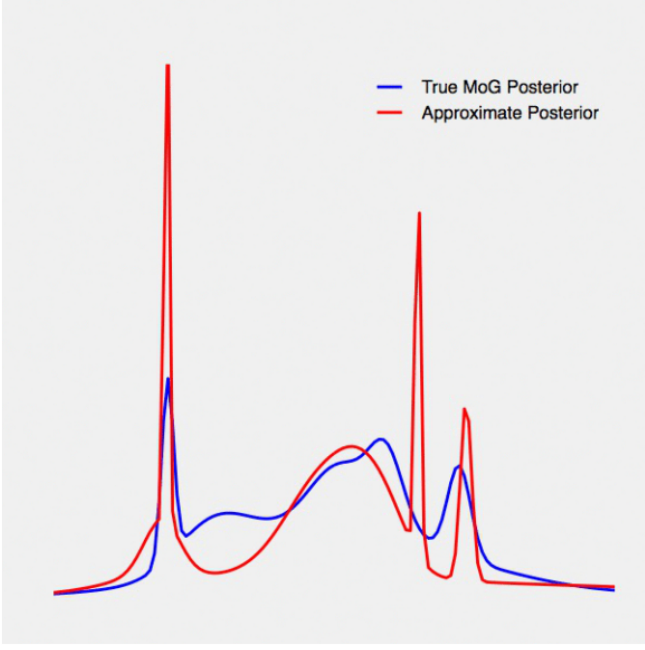
Source: xkcd

Tools of the trade: Approximate Inference

- Have $p(h, d) = p(h)p(d|h)$
- want samples from

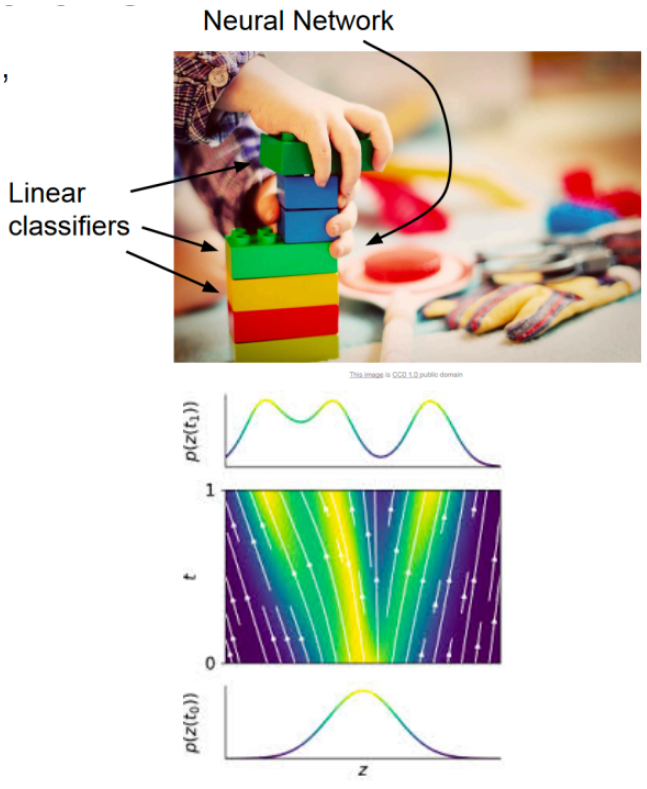
$$p(h|d) = \frac{p(h)p(d|h)}{\sum_{h'} p(h')p(d|h')}$$

- Gradient based methods:
  - Variational inference
  - MCMC



What can you build with these tools?

- Naive Bayes, Mixture of Gaussians, Logistic Regression, Bayesian Linear Regression, Hidden Markov Models, Factor Analysis.
- Neural network classifiers, LSTMs, RNNs, Transformers, Convnets, Neural ODEs, Deep Equilibrium models.
- Variational Autoencoders, Generative Adversarial Networks, Normalizing Flows, Diffusion models.



Scope of course

- Designing, fitting, and interpreting parametric probabilistic models.
  - Conditioning, marginalizing, Normalized versus unnormalized distributions, Graphical models
- Neural nets, gradient-based optimization, automatic differentiation.
- Approximate inference, sampling, variational inference.
- A bit of decision theory.
- Standard software tools: numerics, autodiff.

ML as a bag of tricks

Special cases:

- K-means
- Kernel Density Estimation
- Support Vector Machines
- Boosting
- Random Forests
- K-Nearest Neighbors

Extensible family:

- Mixture of Gaussians
- Latent variable models
- Gaussian processes
- Deep neural nets
- Bayesian neural nets
- Attention based models

Regularization as a bag of tricks

Fast special cases:

- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference

AI as a bag of tricks

Russel and Norvig's parts of AI:

- Machine learning
- Natural language processing
- Knowledge representation
- Automated reasoning
- Computer vision
- Robotics

Extensible family:

- Deep probabilistic latent-variable models + decision theory

This Course

- Introduction to Probabilistic Machine Learning (PML).
- Objectives:
  - Make predictions when there is uncertainty.
  - Combine probability and graph theory to simplify calculations.
  - Learn to sample from unknown distributions reliably.
  - Intersection of PML, GenAI and LLMs.
- Aimed at advanced undergrad and master level graduate students.
- Homeworks more hands-on but overall the lecture focuses on the theory.
- We will use a **fair amount** of real analysis, probability, and linear algebra.

Prerequisites

Do I have the appropriate background?

- **Linear algebra:** vector/matrix manipulations, basic geometric intuitions.
- **Calculus:** partial derivatives/gradient.
- **Probability:** common distributions, Bayes Rule.
- **Statistics:** expectation, variance, covariance, median, maximum likelihood.
- **Machine Learning**: Regression, Basic NN, SGD

E-commerce & Recommender Systems : Amazon, Netflix,...

Inspired by your shopping trends

Customers who bought this item also bought

Fahrenheit 451
› Ray Bradbury
3,502
#1 Best Seller in Censorship & Politics
Paperback
$8.99

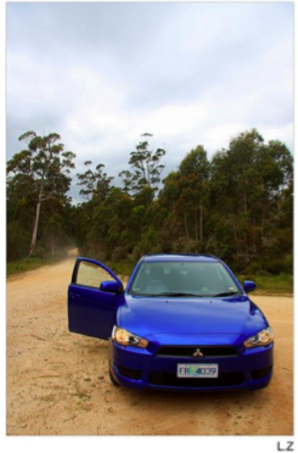
The Glass Castle: A Memoir
› Jeannette Walls
7,651
#1 Best Seller in Journalist Biographies
Paperback
$9.79

Image Infill

Pixel Recurrent Neural Networks (2015)

Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu

Image to Text



a car is parked in
the middle of nowhere .

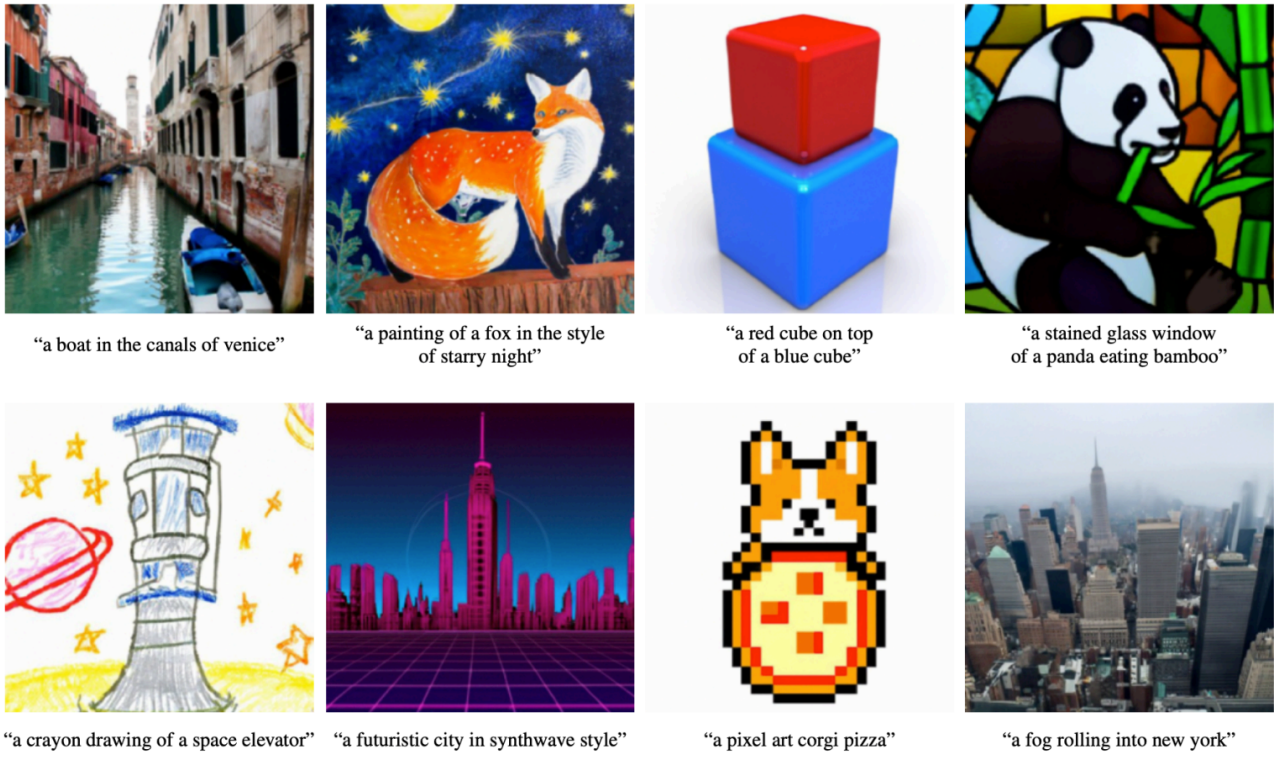
a wooden table and chairs
arranged in a room .

there is a cat sitting on a shelf .

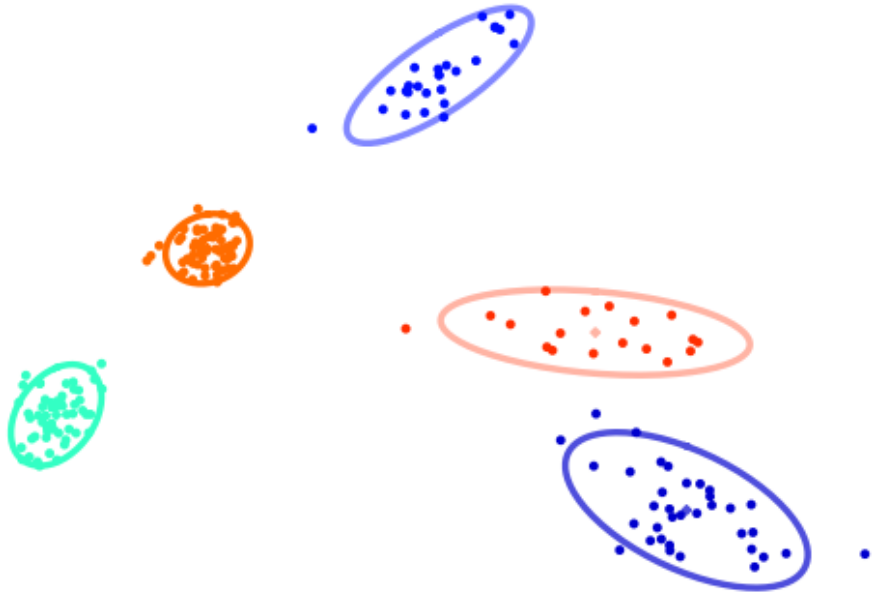
a ferry boat on a marina
with a group of people .

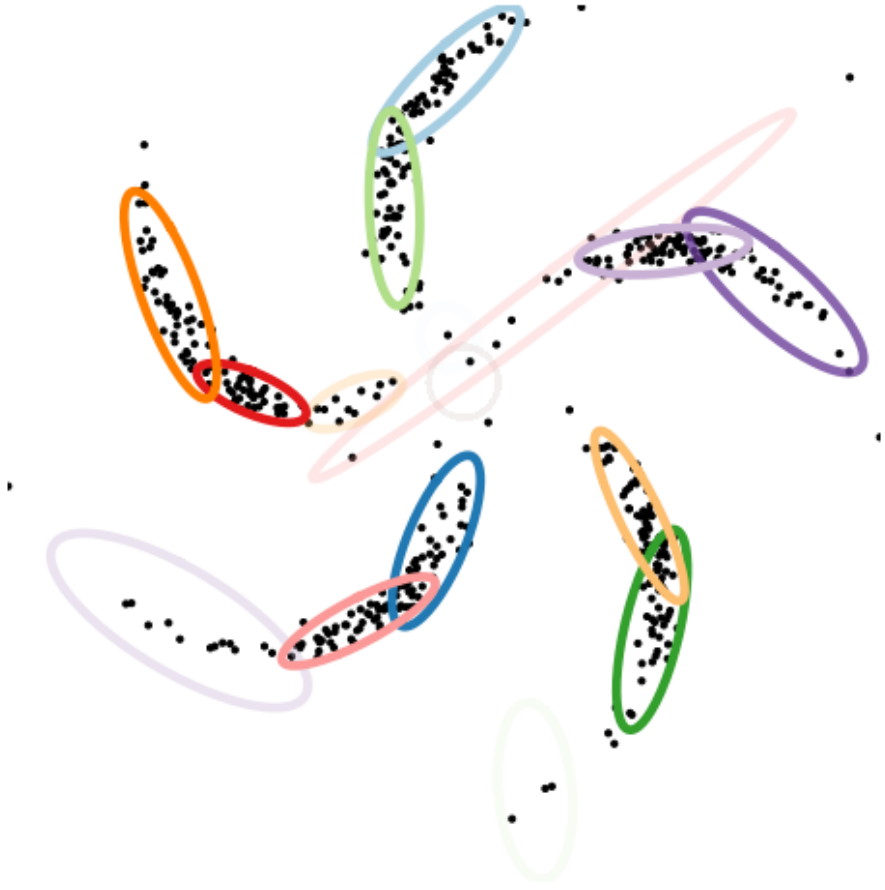
a little boy with a bunch
of friends on the street .

Text to Image



- GLIDE, 2021





Deep generative models

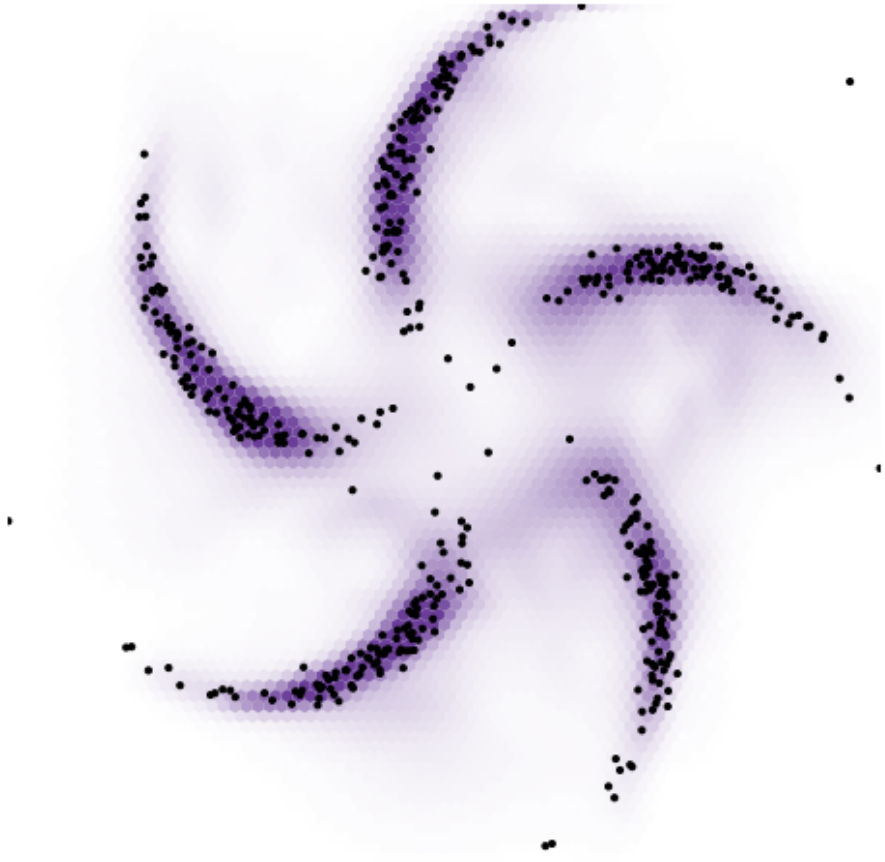
- Model distributions implicitly by a variable pushed through a deep net:

$$y = f_\theta(x)$$

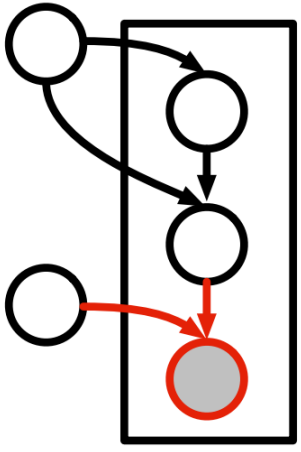
- Approximate intractable distribution by a tractable distribution parameterized by a deep net:

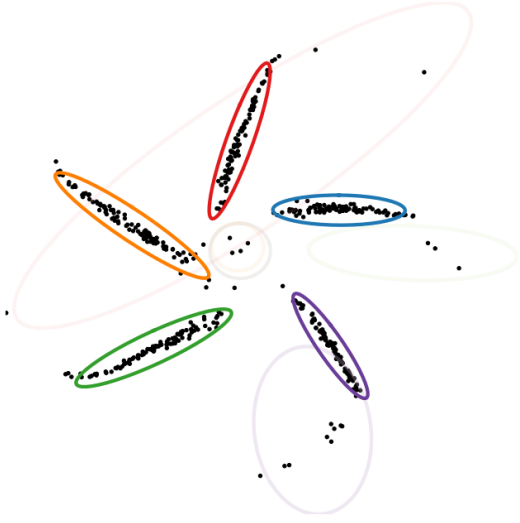
$$p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x))$$

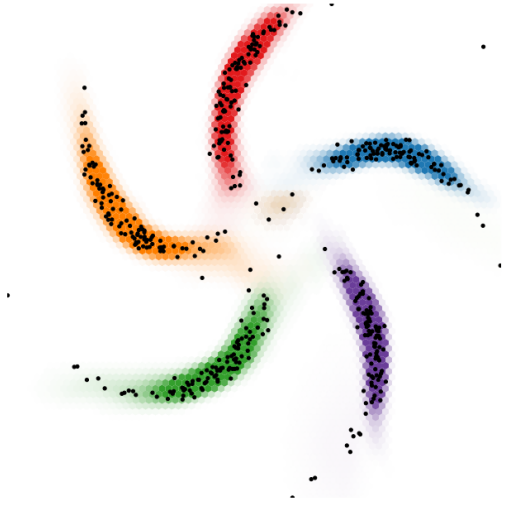
- Optimize all parameters using stochastic gradient descent



Modeling idea: graphical models on latent variables, neural network models for observations







Composing graphical models with neural networks for structured representations and fast inference. Johnson, Duvenaud, Wiltschko, Datta, Adams, NIPS 2016

Implementing machine learning systems

There are many neural networks frameworks, e.g. PyTorch, TensorFlow. Why it is useful to study the **theory** of probabilistic machine learning?

The theory gives you:

- a better understanding of how different algorithms and models work, and how to choose the appropriate ones for a given task.
- a deeper understanding of the mathematical and statistical concepts used in ML, and so a stronger foundation in the field.
- a way to more effectively use systems such as PyTorch or TensorFlow, and customize and fine-tune their models in more sophisticated ways.

Reasons to take this course

- Data science jobs
- Getting into research in ML (but it's a gold rush)
- Doing research in another area, but being able to build / tweak / question models (recommended?)
- Not being impressed by "it was done with deep learning / reinforcement learning / AI"

Administrative details

Course Logistics

Course Website: thibaultrandrianarisoa.netlify.app/courses/sta414/
Main source of information is the course webpage; check regularly!

We will also use Quercus for **announcements & grades etc.**

We will use Piazza for **discussions**.

- Sign up via quercus or:
  https://piazza.com/utoronto.ca/winter2025/sta4142104
- Your grade **does not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers.
- ⚠We will only allow questions that are related to the course materials/assignments/exams.

Subject: STA414

Course Logistics

- 2h lecture + 1h tutorial
- Instructor office hours are Tuesday 9:30-11:30AM, UY 9179
- TA office hours will be announced with each assignment.
- Questions during lectures/tutorials are always welcome!

Course Information

- While cell phones and other electronics are not prohibited in lecture, talking, recording or taking pictures in class is strictly prohibited without the consent of the instructor.
- Lectures are recorded, and videos will be posted on the course webpage.
- Lecture slides and notes will be posted on the course webpage! Please do let us know about typos you notice and/or any suggestions you might have.
- Please do not distribute any course materials that are not publicly available! Check course syllabus for policy.
- For accessibility services: If you require additional academic accommodations, please contact UofT Accessibility Services as soon as possible, studentlife.utoronto.ca/as. No last minute arrangements will be considered.

Literature

Recommended readings given for each lecture.

- Murphy: *"Machine Learning: A Probabilistic Perspective"* (2012)
- **Murphy: *"Probabilistic Machine Learning: An introduction"* (2022)**
- **Murphy: *"Probabilistic Machine Learning: Advanced topics"* (2023)**
- Bishop: *"Pattern Recognition and Machine Learning"* (2006)
- Hastie, Tibshirani, and Friedman: *"The Elements of Statistical Learning"* (2009)
- James, Witten, Hastie and Tibshirani: *"Introduction to Statistical Learning"* (2023)

There are lots of freely available, high-quality ML resources.

Undergrads

Assessment

Evaluation	Weight	Details
Homework Assignments	10%	• Four Homework Assignments (2.5% each) • Pen & paper derivations + Coding (Python/Numpy)
Quizzes	15%	• Every week (starting next week, except Midterm week), before the tutorial • $\sim$ 10 minutes • 10 quizzes in total, keep top 8
Midterm	30%	• $\sim$ 2 hours • Tentative date: Feb 24 • Covers first 5 weeks
Final Exam	45%	• $\sim$ 3 hours • Date TBA • Conceptual/theoretical, minimal coding.

Assessment

project report : 4 - 8 pages

- Graduate students: A project replaces the last two assignments + Less weight on quizzes
- **Everybody must take the final exam! No exceptions.**

Use of generative AI

Homework assignments in this course are designed to reinforce lecture content and provide valuable learning experiences outside of class. While students are encouraged to utilize artificial intelligence tools, including generative AI, as learning aids or for assistance with assignments, it is important to remember that the ultimate responsibility for the submitted work rests with the students.

Grasping complex concepts often involves a layered learning process and intellectual engagement, which cannot be fully substituted by even the most creative use of tools like chat-GPT.

More on Assignments

- Collaboration on the assignments is allowed. After attempting the problems on an individual basis, you may discuss and work together on the homework assignments with up to two classmates. However, you must write your own code and write up your own solutions individually and explicitly name any collaborators at the top of the homework.
- The schedule of assignments will be posted on the course webpage.
- Assignments should be handed in by deadline; a late penalty of 10% **per day** will be assessed thereafter (up to 3 days, then submission is blocked).
- Extensions will be granted only in special situations, and you will need to fill out absence declaration form and **inform the instructor** or have documentation from the accessibility services.
- You will be using Python and Numpy on assignments.

Related Courses

- STA314 and CSC311: Intro ML (we build on these courses)
- STA414/2104: This course
- CSC412/2506: Mostly same material
- CSC413: Neural networks and deep learning
- STA302: Linear regression and classical statistics
- CSC2515: Advanced CSC311
- CSC2532: Learning theory - Focus on mathematics of ML
- Various topics and seminar style courses offered at DoSS and DCS

Why this class?

- This class compliments STA 314 with more focus on unsupervised learning.
- We discuss fundamental probabilistic ideas in machine learning.
- Probabilistic latent variable models and decision theory can cover a wide range of machine learning models.
- This is **not** (or slightly?) a deep learning course! But the principles you will learn are essential to understand deep learning models.

No statistical learning theory, fancy neural network architectures, logic-based AI or reasoning...

Provisional Calendar (tentative)

- week 1, Jan 6:
  - Introduction
  - Probabilistic models (exponential families, MLE)
- week 2, Jan 13:
  - Directed graphical models (DAGs)
  - Statistical Decision Theory
- week 3, Jan 20:
  - Exact inference
  - Message Passing
  - Assignment 1 release on Jan 20
- week 4, Jan 27:
  - Hidden Markov Models
  - Monte-carlo Methods (including Hamiltonian MC)
  - Assignment 1 due on Feb 2

Provisional Calendar (cont'ed)

- week 5, Feb 3:
  - MCMC
  - Assignment 2 release on Feb 3
- week 6, Feb 10:
  - Variational inference
  - Assignment 2 due on Feb 16
- week 7: Reading week
- week 8, Feb 24:
  - Midterm exam (Tentative)
- week 9, Mar 3:
  - Neural Networks
  - Assignment 3 release on Mar 3

Provisional Calendar (cont'ed)

- week 10, Mar 10:
  - Gaussian Processes, Kernel Methods
  - Assignment 3 due on Mar 16
- week 11, Mar 17:
  - Embeddings/Attention/Transformers
  - Assignment 4 release on Mar 18
- week 12, Mar 24:
  - Variational Autoencoders
  - Assignment 4 due on Mar 31
- week 13, Mar 31:
  - Diffusion Models
  - Final Exam Review
- TBD: Final Exam

Generative AI

Introduction to statistical learning

What is Machine Learning?

- Difficult to program correct behavior by hand (e.g., speech understanding, spam detection).
- **Approach:** Program an algorithm to automatically learn from data/experience.

ML vs Statistics

- **Statistics:** Emphasis on interpretability, mathematical rigour, helping scientists/policymakers draw conclusions.
  - Want guarantees, few assumptions, explanations.
- **Machine Learning:** Emphasis on predictive performance, scalability, autonomy, building autonomous agents.
  - No way around making assumptions. Just make model big enough, hopefully it includes something close to the truth.
  - Model needs to have a million parameters somewhere, reality is messy.
  - Can't use guarantees or bounds in practice, so empirically choose model details.

Types of Machine Learning

Supervised Learning

$p(y|x)$

Given input-output pairs $(x^{(i)}, y^{(i)})$, the goal is to learn the mapping f from inputs x to outputs y.

Unsupervised Learning

$p(x)$

Given unlabeled data instances $x^{(i)}$, the goal is to find relations among inputs (e.g., clustering), which can be used for making predictions, decisions. The objective can vary..

Semi-supervised Learning

Given a limited amount of labeled data, i.e. $(x^{(i)}, y^{(i)})$ pairs, but lots of unlabeled $x^{(i)}$'s.

Reinforcement Learning

Learning system receives a reward signal, tries to learn to maximize the reward signal.

Note: These are all special cases of estimating distributions from data: $p(y|x)$, $p(x)$, $p(x, y)$!
This is the main focus of this course.

History of machine learning

- 1943 — Perceptron algorithm (implemented as a circuit in 1957)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him.
- 1969 — Minsky and Papert's book *Perceptrons*
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning (Generalization)
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book *Probabilistic Reasoning in Intelligent Systems* introduced Bayesian networks
    → DAGs

History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also a golden age for ML research
  - Markov chain Monte Carlo, variational inference
  - kernels, support vector machines
  - boosting
  - convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
  - 2010–2012: neural nets smashed previous records in speech-to-text and object recognition; increasing adoption by the tech industry
  - 2016: AlphaGo defeated the human Go champion
  - 2018: Transformer models revolutionized natural language understanding
  - 2020: Self-driving cars, GPT-3 and DALLE demonstrated impressive text and image generation capabilities
  - 2024: Prof. Geoff Hinton wins Nobel prize!

Questions?

?

Part 2: Probabilistic Machine Learning

Overview, MLE, and Exponential Families

Probabilistic Models: Overview

- Consider a random vector $X = (X_1, X_2, ..., X_d)$ (observed or partially observed).
- **Objective:** Model the relationship between these variables
- Probabilistic generative models: relate all variables by their joint probability distribution $p(x) = p(x_1, ..., x_d)$.

Our objective

Suppose there is a true joint p_* which can be approximated by our model $\mathcal{P}$ (i.e., we want to find $p \approx p_*$ where $p \in \mathcal{P}$)

Objective

This course will investigate

1. how we should specify a set of distributions $\mathcal{P}$,
2. what it means for p to be a good approximation of the true distribution p_*,
3. how we can find a reasonable $p \in \mathcal{P}$ efficiently.
4. useful modelling assumptions, e.g. conditional independence

These problem are studied in other statistics courses but here we focus on scalability and autonomy

A Probabilistic Perspective on ML Tasks

With this perspective, think about common machine learning tasks probabilistically:

- input data x (generally high dimensional),
- discrete outputs ("labels") c (e.g. $\{0, 1\}$),
- or continuous outputs y (e.g. daily temperature).

If we have the joint probability over these random variables, e.g. $p(x, y)$ or $p(x, c)$, we can use it for familiar ML tasks:

- **Regression:** $p(y|x) = p(x, y)/p(x) = p(x, y)/\int p(x, y)dy$
- **Classification / Clustering:** $p(c|x) = p(x, c)/\sum_c p(x, c)$

Example: Supervised Classification

We observe pairs of "input data" and "class labels",

$$\{x^{(i)}, c^{(i)}\}_{i=1}^{N} \overset{i.i.d.}{\sim} p(x, c).$$

The supervised classification problem will be to learn a distribution over class labels given new input data:

$$p(c|x) = p(x, c) / \sum_{c} p(x, c)$$

- **Discriminative models:** deal with $p(c|x)$.
- **Generative models:** deal with $p(c, x)$.

Observed vs Unobserved Random Variables

Supervised classification: datasets include input data and class labels

- **Supervised Dataset:** $\{x^{(i)}, c^{(i)}\}_{i=1}^{N} \sim p(x, c)$.
  In this case, the class labels are **observed**.

Unsupervised classification: the data still generated from $p(x, c)$ but instead of the pair $\{x^{(i)}, c^{(i)}\}$ we observe only $x^{(i)}$.

What is the probability of observing $x^{(i)}$?

- **Unsupervised Dataset:** $\{x^{(i)}\}_{i=1}^{N} \sim p(x) = \sum_c p(x, c)$.
  The common way to call an unobserved discrete class is "cluster".
  Possible complication if the number of clusters is unknown.

Desiderata of Probabilistic Models

In order to learn p_* from data $\{x^{(i)}\}$, we make modelling assumptions:

1. **IID data:** We almost always assume that samples $x^{(i)}$ are i.i.d.

2. **"Parametrized" distributions:** The distribution comes from a parametrized family $\mathcal{P} = \{p(x|\theta) : \theta \in \Theta\}$. This reduces the complexity of our search space to the complexity of Θ.
   - e.g. $\mathcal{P} = \{p(x|\theta) = \mathcal{N}(\theta, 1) : \theta \in \mathbb{R}\}$. Gaussian distributions with variance 1 and centered around $\theta \in \mathbb{R}$.
   - Θ may still be **very** high dimensional.

Maximum likelihood estimation

or $(x^{(i)}, y^{(i)})$

Likelihood Function

Let $x^{(i)} \sim p_* = p(x|\theta_*)$ for $i = 1, ..., N$ be i.i.d. random variables. The joint of $\mathcal{D} = \{x^{(1)}, ..., x^{(N)}\}$ is $p(\mathcal{D}|\theta_*) = \prod_i p(x^{(i)}|\theta_*)$.

Likelihood and Log-Likelihood

Assuming we observe $\mathcal{D}$ and θ_* is unknown:

$$\theta \mapsto \mathcal{L}(\theta; \mathcal{D}) = p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(x^{(i)}|\theta)$$

The log-likelihood function is:

$$\ell(\theta; \mathcal{D}) = \log \mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{N} \log p(x^{(i)}|\theta)$$

if $x^{(i)}$ discrete,
$p(x^{(i)}|\theta_*) = \mathbb{P}(X^{(i)} = x^{(i)})$

Natural interpretation in the case when x is discrete

$\mathcal{L}(\theta; \mathcal{D})$ is the probability of observing $\mathcal{D}$ if it was generated from $p(x|\theta)$.

Maximum Likelihood Estimation

How to estimate the true parameter θ_*?

Very intuitive idea: To estimate θ_*, we pick values most likely to have generated the data:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \mathcal{L}(\theta; \mathcal{D}) = \arg\max_{\theta} \ell(\theta; \mathcal{D})$$

Maximizing the log-likelihood is typically easier.

MLE Example: Bernoulli Distribution

Let $x^{(i)} \in \{0, 1\}$ (coin flip) with $P(x = 1) = \theta$.

$$p(x|\theta) = \theta^x (1-\theta)^{1-x}$$

Log-likelihood:

$$\ell(\theta; \mathcal{D}) = \sum_{i=1}^{N} \left(x^{(i)} \log \theta + (1 - x^{(i)}) \log(1-\theta) \right)$$

Taking the derivative w.r.t θ and setting to 0:

$$\frac{\partial \ell}{\partial \theta} = \frac{\sum x^{(i)}}{\theta} - \frac{N - \sum x^{(i)}}{1-\theta} = 0$$

Result: $\hat{\theta}_{MLE} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}$.

Likelihood principle: All relevant information from $\mathcal{D}$ is in the likelihood

Sufficient Statistics

In the Bernoulli example, only $\sum x^{(i)}$ affects the likelihood.

Definition

A statistic is **sufficient** if it conveys exactly the same information about the parameter as the entire data.

Fisher-Neyman Factorization Theorem

$T(x)$ is a sufficient statistic for θ in the parametric model $p(x|\theta)$ if and only if:

$$p(x|\theta) = h(x)g_\theta(T(x))$$

for some functions h (independent of θ) and g_θ.

Exponential families

Exponential Families

Definition

Density of a member of the exponential family is of the form:

$$p(x|\eta) = h(x)\exp\{\eta^\top T(x) - A(\eta)\}$$

$$= h(x)\exp(\eta^\top T(x))\exp(-A(\eta))$$

- $T(x)$: Sufficient statistics
- η: Natural parameter
- $A(\eta)$: Log-partition function
- $h(x)$: carrying measure

Notice that in the exponent, natural parameter interacts with the data **only** through the sufficient statistics.

Examples: Gaussian, Gamma, Exponential, Beta, Dirichlet, Poisson, Geometric...

Some applications

Exponential families have many important applications:

- Many known distributions are EFs.
- Basis for **generalized linear models** (e.g. logistic regression).
- Widely used in multivariate statistics and spatial statistics.
- Many random graph models are exponential families.
- EFs arise as the solution of interesting optimization problems.

(Variational Inference)

The theory of EFs relies heavily on convex analysis.

1-sample example: Bernoulli distribution

We can write this distribution as an exponential family

$$
\begin{aligned}
p(x|\theta) &= \theta^x (1-\theta)^{1-x} \\
&= \exp\{x \log(\theta) + (1-x)\log(1-\theta)\} \\
&= \exp\left\{x \log\left(\frac{\theta}{1-\theta}\right) + \log(1-\theta)\right\}
\end{aligned}
$$

$T(x)$ η $= -A(\eta)$

Here,

$$T(x) = x$$

$$\eta = \log\left(\frac{\theta}{1-\theta}\right)$$

$$A(\eta) = \log(1+e^\eta)$$

$$h(x) = 1$$

$E[T(X)] = E[X] = \theta$

Notice that $A'(\eta) = \frac{e^\eta}{1+e^\eta} = \theta$ is the mean of $T(X) = X$ and $A''(\eta) = \frac{e^\eta}{(1+e^\eta)^2} = \theta(1-\theta)$ is the variance of X.

Mean of sufficient statistics

Moments of exponential families can be easily computed using the log-partition function. Let $X \sim p(x|\eta)$ and denote by $A'(\eta) = dA(\eta)/d\eta$

$$
\begin{aligned}
\mathbb{E}[T(X)] - A'(\eta) &= \int T(x)p(x|\eta)dx - A'(\eta)\int p(x|\eta)dx \\
&= \int \{T(x) - A'(\eta)\}h(x)\exp\{\eta^\top T(x) - A(\eta)\}dx \\
&= \int \frac{d}{d\eta}\left(h(x)\exp\{\eta^\top T(x) - A(\eta)\}\right)dx \\
&= \frac{d}{d\eta}\int p(x|\eta)dx \\
&= \frac{d}{d\eta}1 = 0.
\end{aligned}
$$

Thus, we conclude that $\mathbb{E}_\eta[T(X)] = A'(\eta)$.
The variance $\mathrm{var}_\eta(T(X))$ can be computed similarly: $\mathrm{Var}(T(X)) = A''(\eta)$

MLE for general Exponential Families

Recall: $p(x|\eta) = h(x)\exp\{\eta^\top T(x) - A(\eta)\}$.

After observing data $\mathcal{D}$ with N samples, we write the log-likelihood:

$$\frac{\partial}{\partial \eta}\ell(\eta;\mathcal{D}) = \frac{\partial}{\partial \eta}\log p(\mathcal{D};\eta) = \sum_{i=1}^{N}\log h(x^{(i)}) + \eta^\top \sum_{i=1}^{N} T(x^{(i)}) - NA(\eta)$$

For the MLE derivation we solve:

$$\ell'(\eta;\mathcal{D}) = \sum_{i=1}^{N} T(x^{(i)}) - NA'(\eta) = 0$$

The MLE satisfies: $A'(\hat{\eta}_{\text{MLE}}) = \frac{1}{N}\sum_{i=1}^{N} T(x^{(i)})$. This equation may not have an explicit solution but the solution always corresponds to the global maximum.

Summary

- Probabilistic models are our main tool in machine learning. (in this course)
- We make modelling assumptions (i.i.d., parametric models) for tractability.
- **MLE** is a fundamental method for parameter estimation in this setting.
- **Exponential families** provide a general parametric framework for many known distributions and generalized linear models.

Questions?

?