Probability and Statistics

Michele Caprio

Department of Computer Science, University of Manchester
Manchester Centre for AI Fundamentals

COMP 64101 – Reasoning and Learning under Uncertainty
Lecture 1

Topics: Probability

"I know it's Tuesday. It's a good day for math!"

Max Mintz

- Discrete vs Continuous Distributions (Murphy, 2023, § 2.1.2 - 2.1.3)
- Bayes' Rule (Murphy, 2023, § 2.1.5 - 2.1.6)
- Some Common Probability Distributions (Murphy, 2023, § 2.2 - 2.3)
  - Mixture of Gaussians (Murphy, 2023, § 28.2.1)
- Markov Chains (Murphy, 2023, § 2.6)

Topics: Statistics

"I know it's Tuesday. It's a good day for math!"

Max Mintz

- (Some Concepts of) Bayesian Statistics (Murphy, 2023, § 3.2)
- (Some Concepts of) Frequentist Statistics (Murphy, 2023, § 3.3)
- Maximum Likelihood Estimator and the EM Algorithm (Murphy, 2023, § 6.5.3)

Probability Space

- To talk about probability, we need to introduce the probability space

$$(\Omega, \mathcal{F}, \mathbb{P})$$

Probability Space

- To talk about probability, we need to introduce the probability space

$$(\Omega, \mathcal{F}, \mathbb{P})$$

- Ω is the sample space (possible outcomes from an experiment)
- $\mathcal{F}$ is the event space (σ-algebra), a collection of subsets of Ω
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ is the probability measure

Discrete Sample Space Example

- We flip a coin twice

Discrete Sample Space Example

- We flip a coin twice
- $\Omega = \{\omega_1 = (H, H), \omega_2 = (H, T), \omega_3 = (T, H), \omega_4 = (T, T)\}$

Discrete Sample Space Example

- We flip a coin twice
- $\Omega = \{\omega_1 = (H, H), \omega_2 = (H, T), \omega_3 = (T, H), \omega_4 = (T, T)\}$
- $\mathcal{F} = 2^{\Omega}$, so $|\mathcal{F}| = 2^4 = 16$

Discrete Sample Space Example

- We flip a coin twice
- $\Omega = \{\omega_1 = (H, H), \omega_2 = (H, T), \omega_3 = (T, H), \omega_4 = (T, T)\}$
- $\mathcal{F} = 2^\Omega$, so $|\mathcal{F}| = 2^4 = 16$
- $\mathbb{P}(\{\omega_i\}) = 1/4$, $i \in \{1, \dots, 4\}$, and the probability of the other sets in $\mathcal{F}$ follows by additivity (next slide)

(Kolmogorovian) Probability Axioms

- $\mathbb{P}(E) \geq 0$, for all $E \in \mathcal{F}$

(Kolmogorovian) Probability Axioms

- $\mathbb{P}(E) \geq 0$, for all $E \in \mathcal{F}$
- $\mathbb{P}(\Omega) = 1$

(Kolmogorovian) Probability Axioms

- $\mathbb{P}(E) \geq 0$, for all $E \in \mathcal{F}$
- $\mathbb{P}(\Omega) = 1$
- $\mathbb{P}(\sqcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$

(Kolmogorovian) Probability Axioms

- $\mathbb{P}(E) \geq 0$, for all $E \in \mathcal{F}$
- $\mathbb{P}(\Omega) = 1$
- $\mathbb{P}(\sqcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$
  - Only Finite Additivity: Subjectivist Approach to Probability de Finetti (1974, 1975)
  - Super/Subadditivity: Imprecise Approach to Probability Walley (1991); Augustin et al. (2014)

Discrete Random Variables

- Outcomes of the experiment constitute a countable set

Discrete Random Variables

- Outcomes of the experiment constitute a countable set
- Can we assign a number to each element of Ω (i.e. to each outcome of our experiment of interest)?
- Yes, via a Random Variable (rv)

$$X : \Omega \to \mathbb{R}$$

  - In this case X is a discrete rv because Ω is countable

Discrete Random Variables

- Outcomes of the experiment constitute a countable set
- Can we assign a number to each element of Ω (i.e. to each outcome of our experiment of interest)?
- Yes, via a Random Variable (rv)

$$X : \Omega \to \mathbb{R}$$

  - In this case X is a discrete rv because Ω is countable
- **Example cont'd:** Number of Heads via rv

$$X(\omega_1) = 2, \quad X(\omega_2) = X(\omega_3) = 1, \quad X(\omega_4) = 0$$

Discrete Random Variables

- Random Variable need not assign only numbers to the outcomes of the experiment
- In general, we call state space $\mathcal{X}$ the range[1] of rv X, i.e. $\mathcal{X} = X(\Omega)$

[1] The image of the domain Ω of X under X.

Discrete Random Variables

- Random Variable need not assign only numbers to the outcomes of the experiment
- In general, we call state space $\mathcal{X}$ the range[1] of rv X, i.e. $\mathcal{X} = X(\Omega)$
- We immediately obtain the probability of any given state in $a \in \mathcal{X}$ as

$$p_X(a) = \mathbb{P}[X^{-1}(a)], \quad X^{-1}(a) := \{\omega \in \Omega : X(\omega) = a\}$$

  - p_X is called the probability mass function (pmf) for rv X
  - Can be represented by a histogram or some parametric function

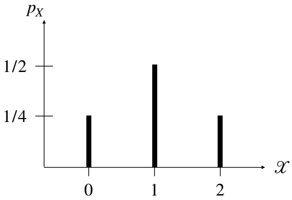
[1] The image of the domain Ω of X under X.

- Random Variable need not assign only numbers to the outcomes of the experiment
- In general, we call state space $\mathcal{X}$ the range[1] of rv X, i.e. $\mathcal{X} = X(\Omega)$
- We immediately obtain the probability of any given state in $a \in \mathcal{X}$ as

$$p_X(a) = \mathbb{P}[X^{-1}(a)], \quad X^{-1}(a) := \{\omega \in \Omega : X(\omega) = a\}$$

  - p_X is called the probability mass function (pmf) for rv X
  - Can be represented by a histogram or some parametric function

- **Example cont'd:** pmf is
  - $p_X(0) = \mathbb{P}(\{(T,T)\}) = 1/4$
  - $p_X(1) = \mathbb{P}(\{(H,T),(T,H)\}) = 1/2$
  - $p_X(2) = \mathbb{P}(\{(H,H)\}) = 1/4$



[1] The image of the domain Ω of X under X.

Continuous Random Variables

- Experiments with continuous outcome
- $\Omega \subseteq \mathbb{R}$, and $X(\omega) = \omega$, so that $\mathcal{X} = \Omega$
- **Example:** The duration of some event (in seconds), so $\Omega = \{t \in \mathbb{R}_+ : t \leq T_{\text{max}}\}$

Conditional Probability

- Consider events E_1 and E_2, and suppose $\mathbb{P}(E_2) > 0$. Then, conditional probability of E_1 given E_2 is

$$\mathbb{P}(E_1 \mid E_2) := \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)}$$

- In turn, $\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1 \mid E_2)\mathbb{P}(E_2) = \mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_1)$

Conditional Probability

- Consider events E_1 and E_2, and suppose $\mathbb{P}(E_2) > 0$. Then, conditional probability of E_1 given E_2 is

$$\mathbb{P}(E_1 \mid E_2) := \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)}$$

- In turn, $\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1 \mid E_2)\mathbb{P}(E_2) = \mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_1)$
- Conditional probability measures how likely an event E_1 is, given that event E_2 has happened

A Note on Independent Events

- E_1 and E_2 are independent events if

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1)\mathbb{P}(E_2)$$

- If both $\mathbb{P}(E_1) > 0$ and $\mathbb{P}(E_2) > 0$, this is equivalent to

$$\mathbb{P}(E_1 \mid E_2) = \mathbb{P}(E_1), \quad \mathbb{P}(E_2 \mid E_1) = \mathbb{P}(E_2)$$

A Note on Independent Events

- E_1 and E_2 are independent events if

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1)\mathbb{P}(E_2)$$

- If both $\mathbb{P}(E_1) > 0$ and $\mathbb{P}(E_2) > 0$, this is equivalent to

$$\mathbb{P}(E_1 \mid E_2) = \mathbb{P}(E_1), \quad \mathbb{P}(E_2 \mid E_1) = \mathbb{P}(E_2)$$

- E_1 and E_2 are conditionally independent given E_3 if

$$\mathbb{P}(E_1 \cap E_2 \mid E_3) = \mathbb{P}(E_1 \mid E3)\mathbb{P}(E_2 \mid E3)$$

Bayes' Theorem

- Consider events E_1 and E_2, and suppose $\mathbb{P}(E_1), \mathbb{P}(E_2) > 0$. Then, Bayes' rule is

$$\mathbb{P}(E_1 \mid E_2) = \frac{\mathbb{P}(E_2 \mid E_1)\mathbb{P}(E_1)}{\mathbb{P}(E_2)}$$

- Discrete case with $|\mathcal{X}| = K$,

$$p(X = k \mid E) = \frac{p(E \mid X = k)p(X = k)}{\sum_{k'=1}^{K} p(E \mid X = k')p(X = k')}$$

- Continuous case, e.g. with $\mathcal{X} = \mathbb{R}$,

$$p(x \mid E) = \frac{p(E \mid x)p(x)}{\int_{\mathcal{X}} p(E \mid x)p(x)\mathrm{d}x}$$

Common Discrete Distributions

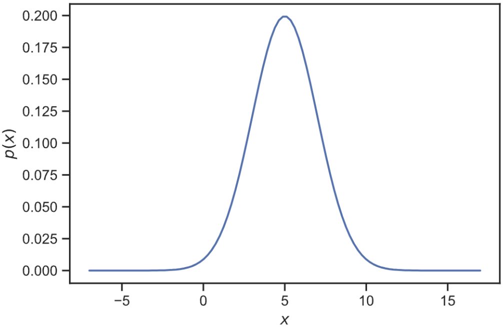
- Let $\mathcal{X} = \{1, \ldots, K\}$
- Binomial: $X \sim \text{Bin}(N, \mu)$, $p(x) = \binom{N}{x} \mu^x (1-\mu)^{N-x}$
  - $\binom{N}{x} := \frac{N!}{(N-x)!x!}$ and $\mu \in [0,1]$
  - Number x of successes in a sequence of N independent experiments, each asking a yes–no question, and having success probability μ
  - Implement it in Python: here

Common Discrete Distributions

- Let $\mathcal{X} = \{1, \ldots, K\}$
- Binomial: $X \sim \text{Bin}(N, \mu)$, $p(x) = \binom{N}{x}\mu^x(1-\mu)^{N-x}$
  - $\binom{N}{x} := \frac{N!}{(N-x)!x!}$ and $\mu \in [0,1]$
  - Number x of successes in a sequence of N independent experiments, each asking a yes–no question, and having success probability μ
  - Implement it in Python: here
- Categorical: $X \sim \text{Cat}(\theta)$, $p_X(k) = \theta_k$
  - θ is a probability vector, and hence belongs to unit simplex $\Delta^{K-1} \subset \mathbb{R}^K$
  - $p_X(k) = \theta_k \rightarrow$ probability that X is equal to k; such probability is the k^{th} entry of parameter θ
  - Fundamental for Classification Problems
  - Implement it in Python: here

Univariate Gaussian Distribution

- $\mathcal{X} = \mathbb{R}$
- Gaussian: $X \sim \mathcal{N}(\mu, \sigma^2)$, $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
  - $\mu \in \mathbb{R}$, $\sigma^2 \in \mathbb{R}_+$
  - Ubiquitous in science; parly because of the Central Limit Theorem
  - Standard Normal: $\mu = 0$, $\sigma = 1$
  - Sensitive to outliers
  - Implement it in Python: here, § 5

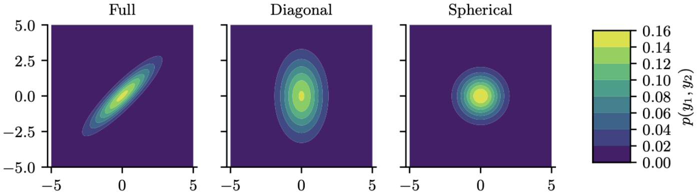


The Holy Grail

- Multivariate Normal: $X \sim \mathcal{N}(\mu, \Sigma)$,
  $p(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right]$
  - $\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{D \times D}$

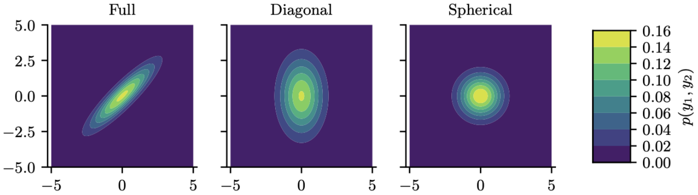
The Holy Grail

- Multivariate Normal: $X \sim \mathcal{N}(\mu, \Sigma)$,
  $p(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right]$
  - $\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{D \times D}$
  - Full Covariance Matrix: $D(D+1)/2$ parameters; we divide by 2 since Σ is symmetric
  - Diagonal covariance matrix: D parameters, and 0s in the off-diagonal terms
  - Spherical covariance matrix: $\Sigma = \sigma^2 I_D$, so only one free parameter σ



The Holy Grail

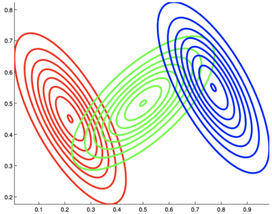
- Multivariate Normal: $X \sim \mathcal{N}(\mu, \Sigma)$,
  $p(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right]$
  - $\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{D \times D}$
  - Full Covariance Matrix: $D(D+1)/2$ parameters; we divide by 2 since Σ is symmetric
  - Diagonal covariance matrix: D parameters, and 0s in the off-diagonal terms
  - Spherical covariance matrix: $\Sigma = \sigma^2 I_D$, so only one free parameter σ



- Implement it in Python: here

Mixture of Gaussians

- Gaussian mixture model (GMM): $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$
  - The π_k's are non-negative and sum up to 1
  - If we let the number of mixture components grow sufficiently large, a GMM can approximate any smooth distribution over $\mathbb{R}^D$
  - GMMs are often used for unsupervised clustering of real-valued data samples $x_n \in \mathbb{R}^D$



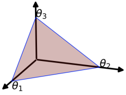
- Implement it in Python: here

Dirichlet Distribution (Unit Simplex Δ^{K-1})

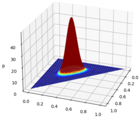
- $X \sim \text{Dir}(\alpha)$, $p(x) \propto \prod_{k=1}^{K} x_k^{\alpha_k - 1}$,

Dirichlet Distribution (Unit Simplex Δ^{K-1})

- $X \sim \mathrm{Dir}(\alpha)$, $p(x) \propto \prod_{k=1}^{K} x_k^{\alpha_k - 1}$, $\alpha \in \Delta^{K-1}$ and $\alpha_0 = \sum_{k=1}^{K} \alpha_k$
- α_0 controls how peaked the distribution is, and the α_k's control where the peak occurs
- Mean: $\mathbb{E}(x_k) = \alpha_k / \alpha_0$

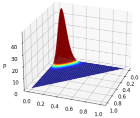


(a)



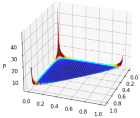
(b)

3.00,3.00,20.00



(c)

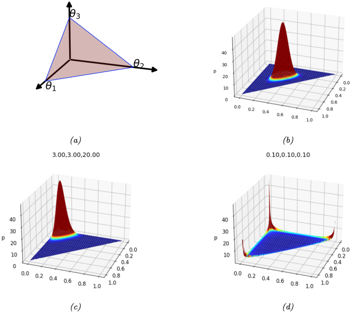
0.10,0.10,0.10



(d)

Dirichlet Distribution (Unit Simplex Δ^{K-1})

- $X \sim \text{Dir}(\alpha)$, $p(x) \propto \prod_{k=1}^{K} x_k^{\alpha_k - 1}$, $\alpha \in \Delta^{K-1}$ and $\alpha_0 = \sum_{k=1}^{K} \alpha_k$
- α_0 controls how peaked the distribution is, and the α_k's control where the peak occurs
- Mean: $\mathbb{E}(x_k) = \alpha_k / \alpha_0$



- Useful to quantify epistemic and aleatoric uncertainties in classification problems
- Implement it in Python: here

Markov Chains

- Let $(x_t)_{t \in \mathbb{N}}$ be a sequence of elements of $\mathbb{R}^D$
- Markov property: $p(x_{t+\tau} \mid x_1, \ldots, x_t) = p(x_{t+\tau} \mid x_t)$

Markov Chains

- Let $(x_t)_{t \in \mathbb{N}}$ be a sequence of elements of $\mathbb{R}^D$
- Markov property: $p(x_{t+\tau} \mid x_1, \dots, x_t) = p(x_{t+\tau} \mid x_t)$
- In turn, we have that $p(x_1, \dots, x_T) = p(x_1) \prod_{t=2}^{T} \underbrace{p(x_t \mid x_{t-1})}_{\text{transition function}}$
    - This is a Markov model (MM)
- If $p(x_t \mid x_{t-1})$ is indep. of time, the MM is called stationary
- Implement it in Python: here

Markov Chains

- Stationary distribution π: intuitively, it is the long term distribution over states
- Finding π: (Murphy, 2023, § 2.6.4)
- Stationary distributions need not always exist (Murphy, 2023, § 2.6.4.3 - 2.6.4.4)

Statistics

- Probability theory: modeling the distribution over observed data outcomes D given known parameters θ by computing $p(D \mid \theta)$
- Statistics: inverse problem. Infer the unknown parameters θ given observations, i.e. compute $p(\theta \mid D)$

Bayesian Statistics: Basic Concepts

- Parameter θ as unknown (rv), and data D as fixed and known
- Represent uncertainty about θ, after seeing data D, by computing the posterior distribution via Bayes' rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{\int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta} \propto p(\theta)p(D \mid \theta)$$

Bayesian Statistics: Basic Concepts

- Parameter θ as unknown (rv), and data D as fixed and known
- Represent uncertainty about θ, after seeing data D, by computing the posterior distribution via Bayes' rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{\int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta} \propto p(\theta)p(D \mid \theta)$$

  - Prior: $p(\theta)$, represents beliefs about parameter before seeing the data

Bayesian Statistics: Basic Concepts

- Parameter θ as unknown (rv), and data D as fixed and known
- Represent uncertainty about θ, after seeing data D, by computing the posterior distribution via Bayes' rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{\int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta} \propto p(\theta)p(D \mid \theta)$$

  - Prior: $p(\theta)$, represents beliefs about parameter before seeing the data
  - Likelihood: $p(D \mid \theta)$, represents beliefs about what data we expect to see, for each setting of the parameters

Bayesian Statistics: Basic Concepts

- Parameter θ as unknown (rv), and data D as fixed and known
- Represent uncertainty about θ, after seeing data D, by computing the posterior distribution via Bayes' rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{\int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta} \propto p(\theta)p(D \mid \theta)$$

  - Prior: $p(\theta)$, represents beliefs about parameter before seeing the data
  - Likelihood: $p(D \mid \theta)$, represents beliefs about what data we expect to see, for each setting of the parameters
  - Marginal likelihood: $p(D) = \int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta$, normalization constant, crucial in Bayesian Model Selection (BMS)

Bayesian Statistics: Basic Concepts

- Parameter θ as unknown (rv), and data D as fixed and known
- Represent uncertainty about θ, after seeing data D, by computing the posterior distribution via Bayes' rule

$$p(\theta \mid D) = \frac{p(\theta)p(D \mid \theta)}{\int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta} \propto p(\theta)p(D \mid \theta)$$

  - Prior: $p(\theta)$, represents beliefs about parameter before seeing the data
  - Likelihood: $p(D \mid \theta)$, represents beliefs about what data we expect to see, for each setting of the parameters
  - Marginal likelihood: $p(D) = \int_\Theta p(\theta)p(D \mid \theta)\mathrm{d}\theta$, normalization constant, crucial in Bayesian Model Selection (BMS)
- Example: see (Murphy, 2023, § 3.2.1)
  - If we assume iid data, then $p(D \mid \theta) = \prod_{y \in D} p(y \mid \theta)$
  - $p(y \mid \theta)$: distributions we introduced before, e.g. a Binomial

Bayesian Statistics: MAP

- Maximum A Posteriori estimate (MAP):

$$\hat{\theta}_{\text{map}} = \arg\max_{\theta \in \Theta} p(\theta \mid D) = \arg\max_{\theta \in \Theta} [\log p(\theta) + \log p(D \mid \theta)]$$

- It is the posterior mode (most probable value)
- Confront it with the MLE

$$\arg\max_{\theta \in \Theta} p(D \mid \theta) = \arg\max_{\theta \in \Theta} \log p(D \mid \theta)$$

  - An extra component coming from the prior $p(\theta)$
  - If we use uniform prior $p(\theta) \propto 1$, MAP $=$ MLE

Bayesian Statistics: MAP

- Maximum A Posteriori estimate (MAP):

$$\hat{\theta}_{\text{map}} = \arg\max_{\theta \in \Theta} p(\theta \mid D) = \arg\max_{\theta \in \Theta} [\log p(\theta) + \log p(D \mid \theta)]$$

- It is the posterior mode (most probable value)
- Confront it with the MLE

$$\arg\max_{\theta \in \Theta} p(D \mid \theta) = \arg\max_{\theta \in \Theta} \log p(D \mid \theta)$$

  - An extra component coming from the prior $p(\theta)$
  - If we use uniform prior $p(\theta) \propto 1$, MAP $=$ MLE
- Implement it in Python: here and (Murphy, 2023, § 6.5.3)

Bayesian Statistics: Posterior Predictive

- Posterior Predictive Distribution:

$$
\begin{aligned}
p(y \mid D) &= \int_{\Theta} p(y \mid \theta) p(\theta \mid D) \mathrm{d}\theta \\
&= \mathbb{E}_{\theta \sim p(\theta \mid D)} \left[p(y \mid \theta) \right]
\end{aligned} \tag{1}
$$

- Given the data D we observed, it tells us what is the probability that the next observation is some value y
- Implement it in Python: here

Bayesian Statistics: Posterior Predictive

- In ML: interested in predicting outcomes y given input features x
- Use conditional probability of the form $p(y \mid x, \theta)$ (e.g. coming from a neural network)
- (Conditional) likelihood is $p(D \mid \theta) = \prod_{(x,y) \in D} p(y \mid x, \theta)$

Bayesian Statistics: Posterior Predictive

- In ML: interested in predicting outcomes y given input features x
- Use conditional probability of the form $p(y \mid x, \theta)$ (e.g. coming from a neural network)
- (Conditional) likelihood is $p(D \mid \theta) = \prod_{(x,y) \in D} p(y \mid x, \theta)$
- Eq. (1) then becomes

$$\begin{aligned} p(y \mid x, D) &= \int_\Theta p(y \mid x, \theta) p(\theta \mid D) \mathrm{d}\theta \\ &= \mathbb{E}_{\theta \sim p(\theta \mid D)} \left[p(y \mid x, \theta) \right] \end{aligned}$$

- By integrating out the unknown parameters, we reduce the chance of overfitting
    - We are computing the weighted average of predictions from an infinite number of models

Frequentist Statistics: Basic Concepts

- Frequentist statistics: uncertainty by calculating how a quantity estimated from data (e.g. a parameter) would change if the data were changed
  - Captured by the sampling distribution of an estimator
- This notion of variation across repeated trials: uncertainty modeling by the frequentist approach

Frequentist Statistics: Sampling Distribution

- Estimator: decision procedure that specifies what action to take given some observed data D
  - Parameter estimation: the action space is to return a parameter vector via function δ, so $\hat{\theta} = \delta(D)$, e.g. the MLE
- Sampling distribution of an estimator: distribution of results we would see if we applied the estimator multiple times to different datasets sampled from some distribution
  - Parameter estimation: it is the distribution of $\hat{\theta}$, viewed as a random variable that depends on the random sample D
- Implement it in Python: here

Frequentist Statistics: Drawbacks

- Frequentist Statistics has some counterintuitive properties (Murphy, 2023, § 3.3.5 - 3.3.6)
- Popular because easy, taught at UG level, sometimes faster to implement than Bayesian
- "Inside every Non-Bayesian, there is a Bayesian struggling to get out", D. Lindley, cf. Jaynes (2002)
- CAREFUL: Bayesian approach is only as correct as its modeling assumptions
    - Check sensitivity of the conclusions to the choice of prior (and likelihood): BMS

Selecting the Prior: Conjugate Priors

- A prior $p(\theta) \in \mathcal{P}$ is a conjugate prior for a likelihood function $p(D \mid \theta)$ if the posterior is in the same parameterized family as the prior, i.e. $p(\theta \mid D) \in \mathcal{P}$
- That is, $\mathcal{P}$ is closed under Bayesian updating
- Conjugate priors simplify the computation of the posterior (Murphy, 2023, § 3.4)
- Implement it in Python: here

Selecting the Prior: Noninformative Priors

- When we have little or no domain specific knowledge, desirable to use a noninformative prior, to "let the data speak for itself"
- No unique way to define such priors, and they all encode some kind of knowledge
  - Better to use the term minimally informative prior (Murphy, 2023, § 3.5)
- Implement it in Python: here

Other Statistical Concepts

- Model selection: we have a set of different models $\mathcal{M}$, each of which may fit the data to different degrees, and each of which may make different assumptions
  - How to pick the best model from this set, or to average over all of them
  - Assumed that the "true" model is in $\mathcal{M}$ (Murphy, 2023, § 3.8)

- Model checking: Bayesian inference is "optimal", but only if the modeling assumptions are correct. How to assess if a model is reasonable?
  - We assume that we do not have a specific alternative model in mind
  - We see if the data we observe is "typical" of what we might expect if our model were correct (Murphy, 2023, § 3.9)

References

Thomas Augustin, Frank P. A. Coolen, Gert de Cooman, and Matthias C. M. Troffaes. *Introduction to imprecise probabilities*. John Wiley & Sons,, West Sussex, England, 2014.

Bruno de Finetti. *Theory of Probability*, volume 1. New York : Wiley, 1974.

Bruno de Finetti. *Theory of Probability*, volume 2. New York : Wiley, 1975.

Edwin T. Jaynes. *Probability Theory. The Logic of Science*. Cambridge University Press: Cambridge, 2002.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL `http://probml.github.io/book2`.

Peter Walley. *Statistical reasoning with imprecise probabilities*, volume 42 of *Monographs on Statistics and Applied Probability*. London : Chapman and Hall, 1991.