CPSC 440: Machine Learning

Conjugate Priors

Winter 2022

Admin

- Assignment 1 grades available on Gradescope.
  - Can ask TAs to look at issues on Gradescope, ask me on Piazza if cannot resolve.
- Assignment 2 due Friday.
  - Hopefully you have started already.
- Assignment 3 will be due 4 weeks after that.
  - 3 weeks of class plus reading week.
  - "Project proposal" will be included as part of Assignment 3.

- Please take reasonable precautions:
  - Spread out through room.
  - Wear mask properly.
  - Do not come to class if you feel sick.
- I am going to try to broadcast/record lectures via Zoom, but no promises.
  - I probably will not follow the chat. Please keep it civil.

Last Time: Monte Carlo Methods

- Monte Carlo approximates expectation of random functions:

$$\underbrace{\mathbb{E}[g(x)] = \sum_{x \in \mathcal{X}} g(x)p(x)}_{\text{discrete } x} \quad \text{or} \quad \underbrace{\mathbb{E}[g(x)] = \int_{x \in \mathcal{X}} g(x)p(x)dx}_{\text{continuous } x}$$

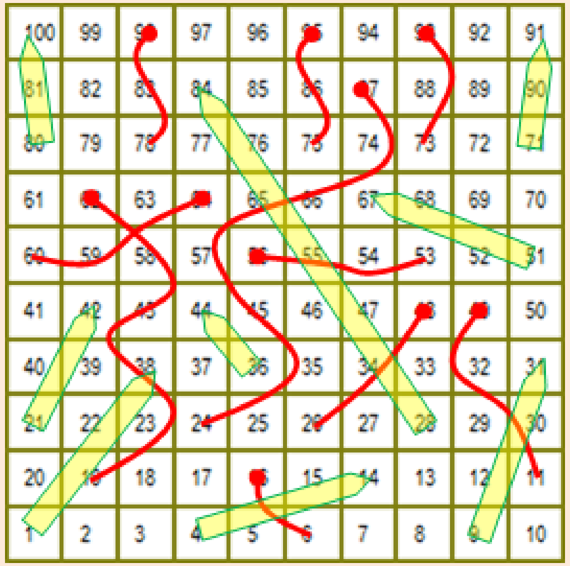
  – Approximation is average of function 'g' applied to samples from 'p':

$$\mathbb{E}[g(x)] \approx \frac{1}{n}\sum_{i=1}^{n} g(x^i)$$

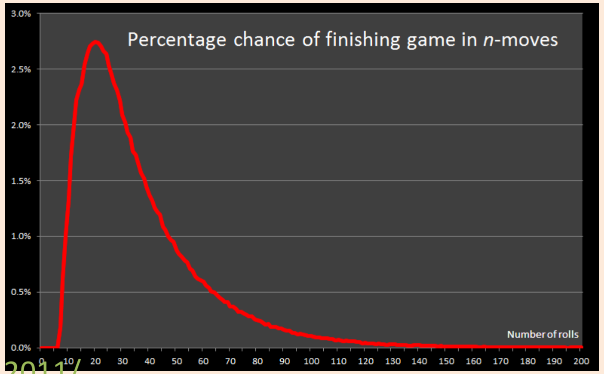
- Can approximate a wide variety of quantities by changing 'g':
  – Mean: g(x) = x.
  – Probability of event 'A': g(x) = I["A happened"].
  – CDF: g(x) = I[x $\leq$ c].
- This is useful when:
  – You know how to sample from p(x).
  – You do not know how to efficiently compute $\mathbb{E}$[g(x)].
  – Are patient because it converges really slowly.

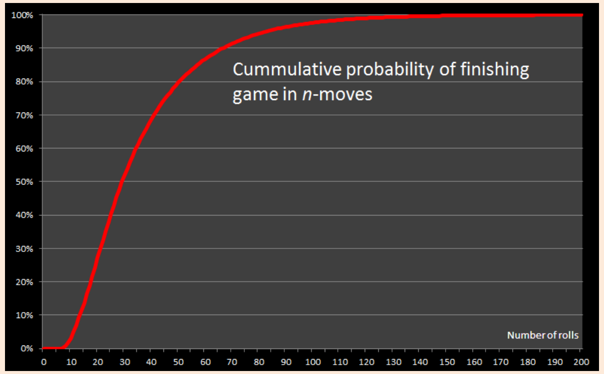
Monte Carlo for Snakes and Ladders

- Consider the children's game "Snakes and Ladders":
  - Start on '1', roll di, move marker, go up/down on ladder/snake, end at 100.
  - No decisions, so you can simulate the game.



- How many turns does it take for this game to end?
  - Simulate game many times, count number of turns.
  - Compute average number of turns.
- Probability and cumulative probability:





https://www.datagenetics.com/blog/november12011/

Conditional Probabilities with Monte Carlo

- We often want to compute conditional probabilites.
  - "What is the probability that the game will go more than 100 turns, if it already went 50 turns?"

- A Monte Carlo approach for estimating p(A | B):
  - Generate a large number of samples.
  - Use Monte Carlo estimate of p(A, B) and p(B) to approximate conditional:

$$p(A \mid B) = \frac{p(A,B)}{p(B)} \approx \frac{\sum_{i=1}^{n} I[\text{"A and B happened"}]}{\sum_{i=1}^{n} I[\text{"B happened"}]}$$

    - Frequency of first event in samples consistent with the second event.
      - This is the MLE for a binary variable that is '1' when 'A' happens, conditioned on 'B' happening.

- This is a special case of rejection sampling (general case later).
  - Unfortunately, if 'B' is rare then most samples are "rejected" (ignored).
  - The conditional probability demo here has a good visualization of this.

Next Topic: MLE and MAP for Categorical

MLE for Categorical Distribution

- Now we will consider how to train a categorical model ("learning").
  - Goal is to go from samples to an estimate of parameters $\theta_1, \theta_2, \dots, \theta_k$:

X =

Party?
LIB
CPC
NDP
LIB
GRN

p(x = LIB) = 0.34, p(x=NDP) = 0.34,
p(x = CPC) = 0.27, p(x=GRN) = 0.03,
p(x = PPC) = 0.02.

- As before we will first consider maximum likelihood estimation:

$$\hat{\Theta} \in \underset{\Theta}{\operatorname{argmax}}\{p(x^1, x^2, \dots, x^n \mid \Theta)\}$$

$\{\theta_1, \theta_2, \dots, \theta_k\}$

  - In this case the MLE is given by $\theta_c = \frac{n_c}{n}$ (n_c is number 'c' examples).
    - If "34% of your samples are LIB, your guess for θ_{LIB}=0.34".
    - As with Bernoulli, the derivation of the MLE is not as a simple as the result.

Derivation of MLE (that does not work)

- Last time we showed that the likelihood has the form:

$$p(X \mid \Theta) = \theta_1^{n_1} \theta_2^{n_2} \cdots \theta_k^{n_k}$$

$x^1, x^2, \ldots, x^n$

$\theta_1, \theta_2, \ldots, \theta_k$

- Let's take the log:

$$\log p(X \mid \Theta) = n_1 \log \theta_1 + n_2 \log \theta_2 + \cdots n_k \log \theta_k$$

- Take the derivative for a particular θ_c:

$$\nabla_{\theta_c} \log p(X \mid \Theta) = \frac{n_c}{\theta_c}$$

- Set derivative equal to zero:

$$0 = \frac{n_c}{\theta_c}$$

- Now what?

Derivation of MLE: Handling "Sum to 1"

- "Set derivative of log-likelihood equal to 0" does not work.
  - Because of constraint that the θ_c must sum to 1, derivative is not zero at MLE.

- Approaches used in textbooks to enforce constraints:
  - Use "Lagrange multipliers" and find stationary point of "Lagrangian".
  - Define $\theta_k = 1 - \sum_{c=1}^{k-1} \theta_c$ to make it unconstrained.
  - See StackExchange thread here.

- We will take a different approach to make it unconstrained:
  1. Use a unnormalized parameterization $\tilde{\theta}_c$ that do not have constraints.
  2. Compute the MLE for the $\tilde{\theta}_c$ by setting log-likelihood derivative zero.
  3. Convert from the $\tilde{\theta}_c$ parameters to our usual θ_c parameters by normalizing.

Unconstrained Parameterization

- Consider categorical distribution with unnormalized parameters:

$$p(x=c \mid \tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_k) \propto \tilde{\theta}_c$$

  – To give non-negative probabilities, we require that $\bar{\theta}_c \geq 0$ for all 'c'.

- The normalized probability can then be written:

$$p(x=c \mid \tilde{\Theta}) = \frac{\tilde{\theta}_c}{\sum_{c'=1}^{k} \tilde{\theta}_{c'}} = \frac{\tilde{\theta}_c}{Z}$$

$Z = \sum_{c'=1}^{k} \tilde{\theta}_{c'}$ is called the "normalizing constant"

It is constant in terms of 'x' but a function of $\tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_k$

  – The "normalizing constant" makes the probability sum to 1 across 'c' values.
    - So we do not need to an explicit "sum to 1" constraint.
  – We convert from unnormalized to normalized by dividing by 'Z': $\theta_c = \frac{\tilde{\theta}_c}{Z}$.

Derivation of MLE (that does work)

- Using the unnormalized parameters in the likelihood gives::

$$p(X \mid \Theta) = \left(\frac{\tilde{\theta}_1}{Z}\right)^{n_1}\left(\frac{\hat{\theta}_2}{Z}\right)^{n_2}\cdots\left(\frac{\theta_k}{Z}\right)^{n_k} = \frac{\tilde{\theta}_1^{n_1}\hat{\theta}_2^{n_2}\cdots\theta_k^{n_k}}{Z^n}$$

- Let's take the log: $\log p(X \mid \Theta) = n_1 \log(\hat{\theta}_1) + n_2 \log(\hat{\theta}_2) + \cdots + n_k \log(\hat{\theta}_k) - n \log Z$

- Take the derivative for a particular θ_c: $\nabla_{\hat{\theta}_c} p(X \mid \Theta) = \frac{n_c}{\hat{\theta}_c} - \frac{n}{Z}$

- Set derivative equal to zero: $0 = \frac{n_c}{\hat{\theta}_c} - \frac{n}{Z}$

- Solve for $\tilde{\theta}_c$: $\frac{\tilde{\theta}_c}{Z} = \frac{n_c}{n}$ $\Big\} \longrightarrow$ Convert to normalized: $\theta_c = \frac{n_c}{n}$

(and possible to show this maximizes likelihood)

$\log Z = \log\left(\sum_{c=1}^{k} \tilde{\theta}_c\right)$

$\nabla_{\tilde{\theta}_c} \log Z = \frac{1}{\sum_{c=1}^{k} \tilde{\theta}_c} = \frac{1}{Z}$

MAP Estimation and Dirichlet Prior

- As before, we may prefer to use a MAP estimate over the MLE.
  - Often becomes more important as ‘k’ grows.
    - More parameters to [over]fit.

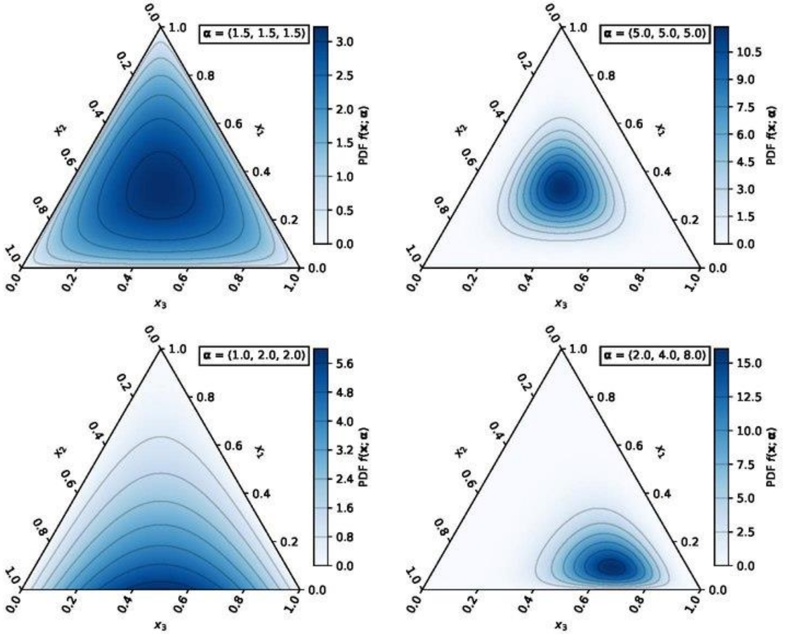
- Most common prior for categorical is the Dirichlet distribution:

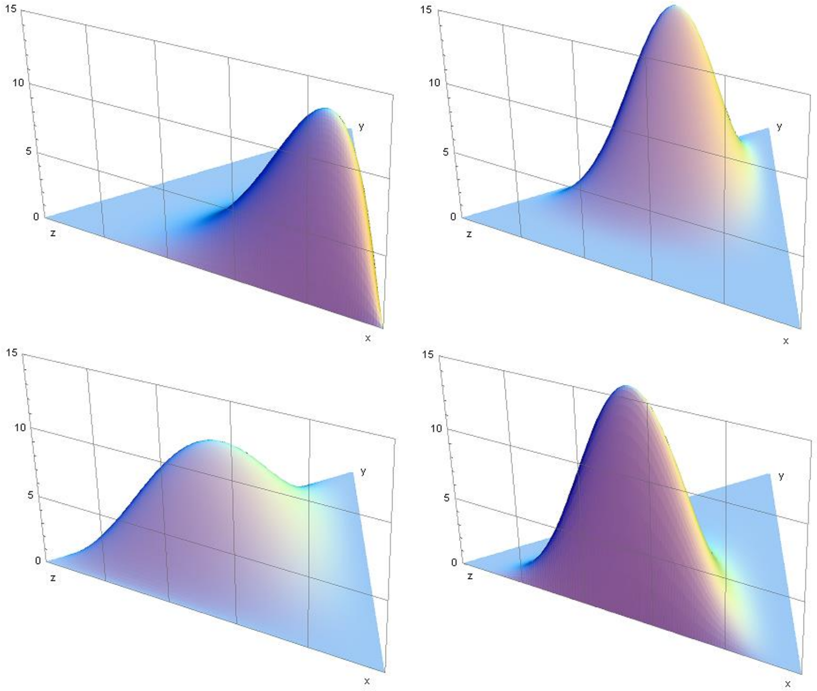
$$p(\theta_1, \theta_2, \dots, \theta_k \mid \alpha_1, \alpha_2, \dots, \alpha_k) \propto \theta_1^{\alpha_1 - 1} \theta_2^{\alpha_2 - 1} \cdots \theta_k^{\alpha_k - 1}$$

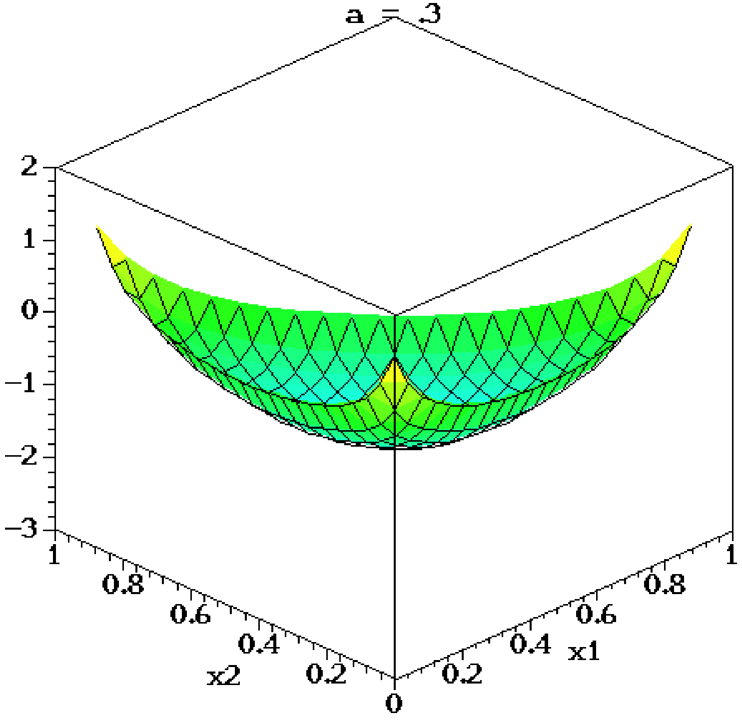
  - Generalization of the beta distribution to ‘k’ classes.
- This is a distribution over Θ values:
  - Since the Θ parameterize probabilities,
    Dirichlet is a probability distribution over possible probability distributions.

Dirichlet Distribution

- Wikipedia's visualizations of Dirichlet distribution for k=3:







all α_i equal

- Can bias towards various types of probabilities.

https://en.wikipedia.org/wiki/Dirichlet_distribution
https://commons.wikimedia.org/wiki/Category:Dirichlet_distribution

MAP Estimation and Dirichlet Prior

- The MAP for categorical with Dirichlet prior is given by:

$$\hat{\theta}_c = \frac{n_c + \alpha_c - 1}{\sum_{c'=1}^{k} [n_{c'} + \alpha_{c'} - 1]}$$

  - Derivation is similar to the MLE derivation.
- Dirichlet has 'k' hyper-parameters α_c.
  - We often set $\alpha_c = \alpha$ for some constant α (reduces to 1 hyper-parameter).
  - This simplifies the MLE to:

$$\hat{\theta}_c = \frac{n_c + \alpha - 1}{\sum_{c'=1}^{k} n_{c'} + k(\alpha - 1)}$$

  - And with $\alpha = 2$ we get Laplace smoothing ("add 1 to count of each class").

Posterior for Categorical Likelihood + Dirichlet Prior

- People use the Dirichlet because posterior has a simple form:

$$p(\Theta \mid X, \alpha) \propto p(X \mid \Theta)\, p(\Theta \mid \alpha) \propto \theta_1^{n_1} \theta_2^{n_2} \cdots \theta_k^{n_k}\, \theta_1^{\alpha_1 - 1} \theta_2^{\alpha_2 - 1} \cdots \theta_k^{\alpha_k - 1}$$

$\{\theta_1, \theta_2, \dots, \theta_k\}$ $(\alpha_1, \alpha_2, \dots, \alpha_k)$

Assuming data is independent of parameters given hyper-parameters

$$= \theta_1^{(n_1 + \alpha_1) - 1} \theta_2^{(n_2 + \alpha_2) - 1} \cdots \theta_k^{(n_k + \alpha_k) - 1}$$

$$= \theta_1^{\tilde{\alpha}_1 - 1} \theta_2^{\tilde{\alpha}_2 - 1} \cdots \theta_k^{\tilde{\alpha}_k - 1}$$

  - This is another Dirichlet distribution with "updated" parameters $\tilde{\alpha}_c$.
    - Where $\tilde{\alpha}_c = n_c + \alpha\_c$.
    - Again, make sure you understand why we can recognize this as a Dirichlet.
      - The normalizing constant must be the normalizing constant for the Dirichlet.

$$Z = \int_0^1 \int_0^1 \cdots \int_0^1 \theta_1^{\tilde{\alpha}_1 - 1} \theta_2^{\tilde{\alpha}_2 - 1} \cdots \theta_k^{\tilde{\alpha}_k - 1}\, d\theta_1\, d\theta_2 \cdots d\theta_k$$

Conjugate Priors

- We have now some examples of a convenient phenomenon:
  - If we put a beta prior on a Bernoulli likelihood, posterior is beta.
    - Same happens if you put beta prior on binomial/geometric, posterior is beta.
  - If we put a Dirichlet prior on a categorical likelihood, posterior is Dirichlet.

- In these situations, we say the prior is conjugate to the likelihood.
  - With conjugate priors, the prior and posterior come from the same "family".

$$x \sim D(\theta), \quad \theta \sim P(\lambda) \quad \Rightarrow \quad \theta \mid x \sim P(\lambda')$$

this means "has the probability distribution of"

  - The posterior will look like the prior with "updated" parameters.

- Many computations become easier when we use conjugate priors.
  - Because we have an explicit formula for the posterior distribution.
  - But not all distributions have conjugate priors.

Next Topic: Bayesian Learning

Problems with MAP

- With good hyper-parameters, MAP usually outperforms MLE.

- But MAP is still weird.
  - Recall that we said that decoding can do weird things.
    - The value with highest probability/PDF may not represent “typical” behavior.
  - MAP is a decoding of the posterior.

- MAP is fine if you want to find parameters with highest probability, but in ML usually the goal is to make predictions (or decisions).
  - Our ultimate goal is not just to find the best parameters.

- You can show that MAP is a sub-optimal way to make predictions.

Example: "Two Heads" with "Fair vs. Unfair" Prior

- Suppose you have a Bernoulli variable and the following prior:
  - p(θ = 0.5) = 0.5 and p(θ = 1) = 0.5.
    - You think coin has 50% chance of being fair, 50% chance of "always landing head".
- The first two coin flips are "head".
  - $x^1 = 1$, $x^2 = 1$.
- What is the probability that the third flip will be a "head"?
  - MAP approach:

1. Find $\hat{\theta} \in \text{argmax}\{p(\theta \mid X)\} \equiv \text{argmax}\{p(X \mid \theta)p(\theta)\}$

2. Compute $p(x^3 = 1 \mid \hat{\theta} = 1) = 1$

$\theta = 1/2$: $(1/2)(1/2)(1/2) = 1/8$

$\theta = 1$: $(1)(1)(1/2) = 1/2$

Since $1/2 > 1/8$, set $\hat{\theta} = 1$

  - MAP predicts 100% chance of head.
    - But the MAP "decoding" of the parameters is over-confident.
      - There was a 1/4 chance of seeing two heads from the fair coin.

Example: "Two Heads" with "Fair vs. Unfair" Prior

- Can compute correct probability using marginalization rule over θ:

$$\underbrace{p(x^3=1 \mid X)}_{\text{the probability we want}} = \sum_{\theta \in \{0.5, 1\}} p(x^3=1, \theta \mid X) = \sum_{\theta \in \{0.5,1\}} \underbrace{p(x^3=1 \mid \theta, X)}_{\text{prediction given } \theta} \underbrace{p(\theta \mid X)}_{\text{posterior}}$$

marg. rule

product rule

- The correct probability weights possible predictions by posterior.
  - Assume x^3 is independent of X once we know θ: $p(x^3=1 \mid \theta, X) = p(x^3=1 \mid \theta)$
  - Use Bayes rule to compute posterior and get final answer:

$$p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{\sum_{\theta'} p(X \mid \theta')p(\theta')}$$

$\theta = 1/2$: $\frac{1/8}{1/8 + 1/2} = \frac{1}{5}$

$\theta = 1$: $\frac{1/2}{1/8 + 1/2} = \frac{4}{5}$

plug in

probability from "fair" case

probability from "unfair" case

$$p(x^3=1 \mid X) = (1/2)\left(\frac{1}{5}\right) + (1)\left(\frac{4}{5}\right) = \frac{9}{10}$$

Bayesian Approach to Machine Learning

- MAP predicted 100% chance that third coin would be a head.
  - But the correct value was only 90% (obtained by marginalizing over θ).

- "Compute correct probability by marginalizing over parameters" is called the Bayesian approach to machine learning.
  - MAP approach optimizes posterior over parameter values.
    - Searches for the single "best" parameter value according to posterior.
  - Bayesian approach marginalizes posterior over parameter values.
    - Considers all possible parameter values, but upweighting ones with high posterior.

- MAP and Bayes are similar if posterior is "concentrated" at one θ.
  - But if there are many reasonable θ, Bayes can be much better.

Summary

- MLE for categorical distribution:
  - Write using unnormalized parameters and normalizing constant 'Z'.
- Dirichlet distribution:
  - "Probability distribution over discrete probability distributions".
  - When used as prior for categorical, posterior is also Dirichlet.
  - MAP estimate with Dirichlet prior gives generalization of Laplace smoothing.
- Conjugate prior:
  - Prior for a particular likelihood such that posterior is in same "family".
- Bayesian learning:
  - Use marginalization rule to consider all possible parameters.
    - Unlike MLE/MAP which optimize to find "best" parameters.
  - The correct way to combine likelihood with prior.

- Next time: better way to reduce overfitting than averaging or regularization?