EE5907/EE5027 Week 2: Probabilistic Estimation + Conjugate Priors

BT Thomas Yeo

ECE, CIRC, Sinapse, Duke-NUS, HMS

Last Week Recap

- Discrete & continuous random variables
- Probability distribution function (pdf) / probability mass function (pmf)
- Joint distributions of N random variables
- Marginalization / Law of Total Probability
- Conditional probability
- Bayes' Rule
- Independence, Conditional Independence
- Expectation, conditional expectation

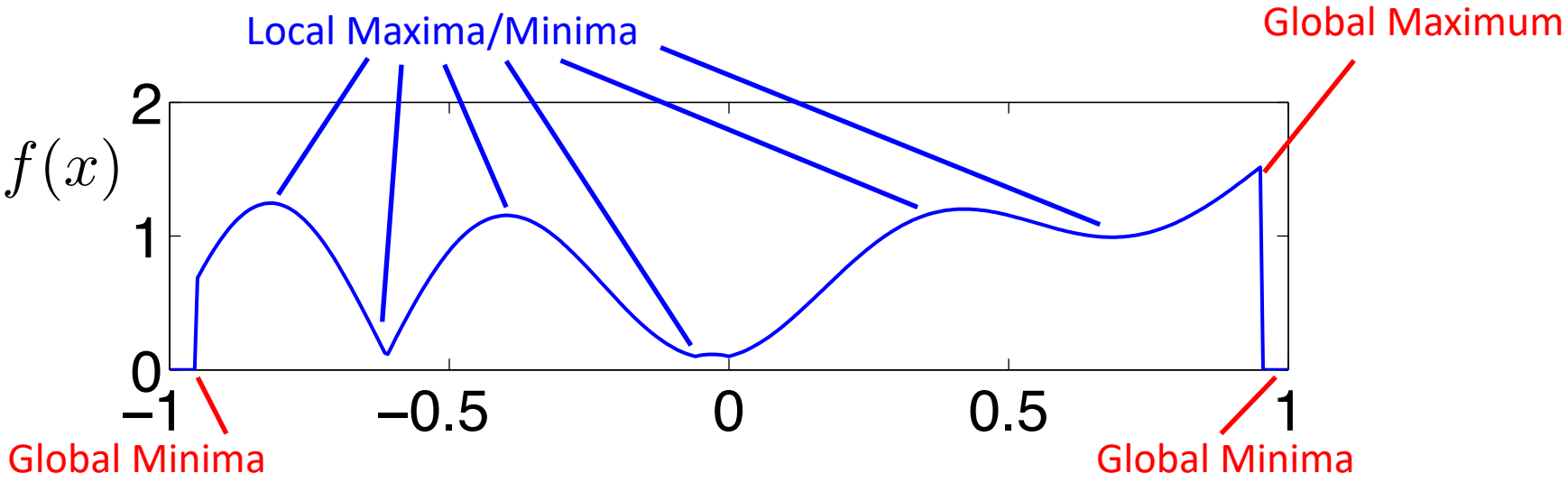
Key Goal in Machine Learning: Signal Detection or Estimation

- Given observation x, estimate y by computing y* = d(x)
  - ”d” stands for “detection”
- In real world, have to take action based on what we think y is
  - x is cancer test results & y is whether patient has cancer
  - x is radar signal & y is whether there is an incoming missile
  - x is facebook photo & y is name of person in photo
- Obviously “detection” does not need to be probabilistic, but the world is uncertain (e.g., measuring instruments have noise, unexpected things might happen)
  - Therefore, we will focus on probabilistic signal detection or estimation

Maximum-A-Posterior (MAP) and Maximum Likelihood (ML) Estimation

Digression: argmax and argmin

- $\text{argmax}_x f(x)$ is value of x where $f(x)$ is biggest
- $\text{argmin}_x f(x)$ is value of x where $f(x)$ is smallest
- $f(x) = \begin{cases} |\sin(4x)^2 + x| \exp(-x) + x^2 + 0.1, & -0.95 \leq x \leq 0.95 \\ 0 & \text{otherwise} \end{cases}$
  - Generally easier to evaluate $f(x)$ than find maximum or minimum
  - Real problems: may have to live with local maximum or minimum



MAP and ML Estimation

- Maximum-A-Posteriori (MAP) estimation:

$$y_{MAP} \triangleq \operatorname*{argmax}_y p(y|x)$$

Bayes' rule

$$= \operatorname*{argmax}_y \frac{p(y)p(x|y)}{p(x)}$$

p(x) not function of y

$$= \operatorname*{argmax}_y p(y)p(x|y)$$

- Example: $p(y = \text{chair}|x = \text{photo}) = 0.6$, $p(y = \text{human}|x = \text{photo}) = 0.3$, $p(y = \text{cat}|x = \text{photo}) = 0.1 \implies y_{MAP} = \text{chair}$
- Example: $p(y = \text{chair}|x = \text{photo}) \propto 1.2$, $p(y = \text{human}|x = \text{photo}) \propto 0.6$, $p(y = \text{cat}|x = \text{photo}) \propto 0.2$
  - Notice $1.2 + 0.6 + 0.2 = 2$, so not a valid probability distribution, which is why we use "$\propto$" rather than "$=$" but y_{MAP} still the same (chair)
  - If we want proper distribution, $c = 1.2+0.6+0.2 = 2$, so we can normalize to become proper distribution: $p(y = \text{chair}|x = \text{photo}) = 1.2/c = 0.6$, $p(y = \text{human}|x = \text{photo}) = 0.6/c = 0.3$, $p(y = \text{cat}|x = \text{photo}) = 0.2/c = 0.1$

MAP and ML Estimation

- Maximum-A-Posteriori (MAP) estimation:

$$\begin{aligned} y_{MAP} &\triangleq \underset{y}{\operatorname{argmax}}\, p(y|x) \\ &= \underset{y}{\operatorname{argmax}}\, \frac{p(y)p(x|y)}{p(x)} \\ &= \underset{y}{\operatorname{argmax}}\, p(y)p(x|y) \end{aligned}$$

Bayes' rule

p(x) not function of y

- Prior $p(y)$ is constant (uniform) $\implies$ maximum likelihood (ML) estimate

$$y_{MAP} = \underset{y}{\operatorname{argmax}}\, p(x|y) \triangleq y_{ML}$$

- ML often easier to compute; often use when prior is unknown (or do not want to assume priors)
- If # samples goes to infinity (infinite amount of data), then $\lim_{N\to\infty} y_{MAP} = y_{ML}$

What is hard here?

- ML is a special case of MAP, so let's focus on MAP (for now)
- In previous example of chair, human & cat, I gave you p(y | x = photo), but how to get p(y | x = photo) in the first place?
- Much of machine learning is about how to choose a model for p(y | x) and how to evaluate/optimize model parameters

Two strategies for modeling posterior probability p(y | x): Generative vs Discriminative Models

Modeling p(y | x)

- $p(y \mid x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y)$
  - Generative supervised learning: model likelihood p(x|y) & prior p(y) directly, then apply above formula to get p(y | x)
  - ”Generative” because we can sample (generate) new data (x, y) from p(x, y) = p(y)p(x|y)
  - Examples: Naive Bayes (future lesson), hidden Markov models, etc.
- Discriminative supervised learning: model p(y | x) directly
  - Because we have p(y | x), but do not model p(x), we do not have p(x, y), so cannot sample new data (x, y)
  - Examples: Logistic regression (future lesson), conditional random fields, etc

Pros and Cons

- Suppose our only goal is to predict y from x,
  - Discriminative approach models p(y | x) directly => focusing all modeling resources on goal directly => generally better accuracy than generative approach
- Generative model generally more interpretable than discriminative model
  - Example: Given two features $x_1 = y + \varepsilon$ and $x_2 = -\varepsilon$
  - Perfect (supervised) classification accuracy if we predict $y = x_1 + x_2$
  - Hard to interpret classifier "$1x_1 + 1x_2$" because both features are weighted equally (x_1 and x_2 have coefficients of 1), but x_2 serves to cancel noise and has basically no relationship with y.
- Generative models allow us to sample data from p(x, y). If the data looks realistic, then prediction accuracy can be high

Generative/Discriminative Models Need Not Be Probabilistic

- More general definition: Discriminative models are models focused on prediction (e.g., support vector machines)
- More general definitions: Generative models are models that can (in principle) be used to "generate" observed data
- Newton's 3 laws of motion is a generative model because the laws allow one to generate trajectories of balls given external forces + initial states

Concept of Generative Models Not Limited To Supervised Learning

- Although I introduce generative models in context of supervised learning, generative models can be used in unsupervised learning, where we only have features x
  - Examples: Gaussian mixture models, latent dirichlet allocation, restricted boltzmann machine, etc
- By definition, discriminative models are focused on prediction => the term “discriminative model” only makes sense in the context of supervised learning
- Neural networks can be generative or discriminative

Probabilistic Estimation of Model Parameters

Probabilistic Estimation of Model Parameters

- Previous example:
  - p(y = chair | x = photo) = 0.6
  - p(y = human | x = photo) = 0.3
  - p(y = cat | x = photo) = 0.1
  - How can 0.6, 0.3 and 0.1 appear on right side, but not left side of "=" sign?
- We should technically include $\boldsymbol{\theta} = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix}$ as variable on left hand side
  - p(y = chair | x = photo, $\boldsymbol{\theta}$) = $\boldsymbol{\theta}_1$ = 0.6
  - p(y = human | x = photo, $\boldsymbol{\theta}$) = $\boldsymbol{\theta}_2$ = 0.3
  - p(y = cat | x = photo, $\boldsymbol{\theta}$) = $\boldsymbol{\theta}_3$ = 0.1
  - In this example, posterior distribution is categorical distribution with parameter $\boldsymbol{\theta}$
- In general, $\boldsymbol{\theta}$ needs to be learned from the training set for both generative models p(x, y | $\boldsymbol{\theta}$) & discriminative models p(y | x, $\boldsymbol{\theta}$)
- In first bullet point on this slide: y & x are concrete things (photo, chair, human, cat), but ML/MAP can also be used to estimate "abstract" quantities like $\boldsymbol{\theta}$. In other words, we can also treat parameters of probability distribution as random variables and estimate them using ML/MAP

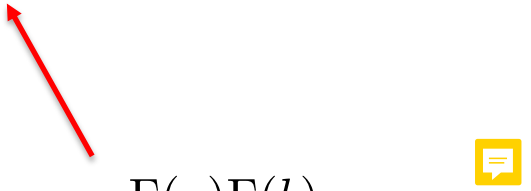
Parameters of Probability Distribution can themselves be treated as random variables

- Given training set D = $\{x_i, y_i\}_{i=1:N}$, where x = feature, y = target label
- Goal: learn parameters $\boldsymbol{\theta}$ of generative model p(x, y | $\boldsymbol{\theta}$) from D, so that given new test data x, can predict y using MAP estimate of posterior by plugging in estimate of $\boldsymbol{\theta}$: $p(y \mid x, \boldsymbol{\theta}) \propto p(x, y \mid \boldsymbol{\theta}) = p(x \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta})$
- Strategy 1 (Maximum likelihood)
  - Step 1: Estimate $\boldsymbol{\theta}_{ML}$ = $\text{argmax}_{\boldsymbol{\theta}}$ p(D | $\boldsymbol{\theta}$)
  - Step 2: Plug in $\boldsymbol{\theta}_{ML}$ into p(x, y | $\boldsymbol{\theta}_{ML}$) and find MAP estimate of y
- Strategy 2 (Maximum-A-Posteriori)
  - Step 1: Estimate $\boldsymbol{\theta}_{MAP}$ = $\text{argmax}_{\boldsymbol{\theta}}$ p($\boldsymbol{\theta}$ | D)
  - Step 2: Plug in $\boldsymbol{\theta}_{MAP}$ into p(x, y | $\boldsymbol{\theta}_{MAP}$) and find MAP estimate of y
- Strategy 3 (Posterior Predictive / Bayesian Model Averaging)
  - More in the future
- Since parameters $\boldsymbol{\theta}$ of probability distributions are treated as random variables that can be estimated, let's see how this is done for various distributions

Beta-Binomial Generative Model (multiple coin tossing)

Let's Model θ as a Random Variable

- Consider N coin tosses: data $D = (N_0, N_1)$, where $N_0 = \#$ tails, $N_1 = \#$ heads in N coin tosses, $\theta =$ probability of head, then
  - $p(D|\theta) = \text{Bin}(N_1|N, \theta) = \binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}$ (binomial likelihood)
- Assume beta prior on θ: $p(\theta|a, b) = \text{Beta}(\theta|a, b) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$
  - $a > 0, b > 0$ called hyperparameters



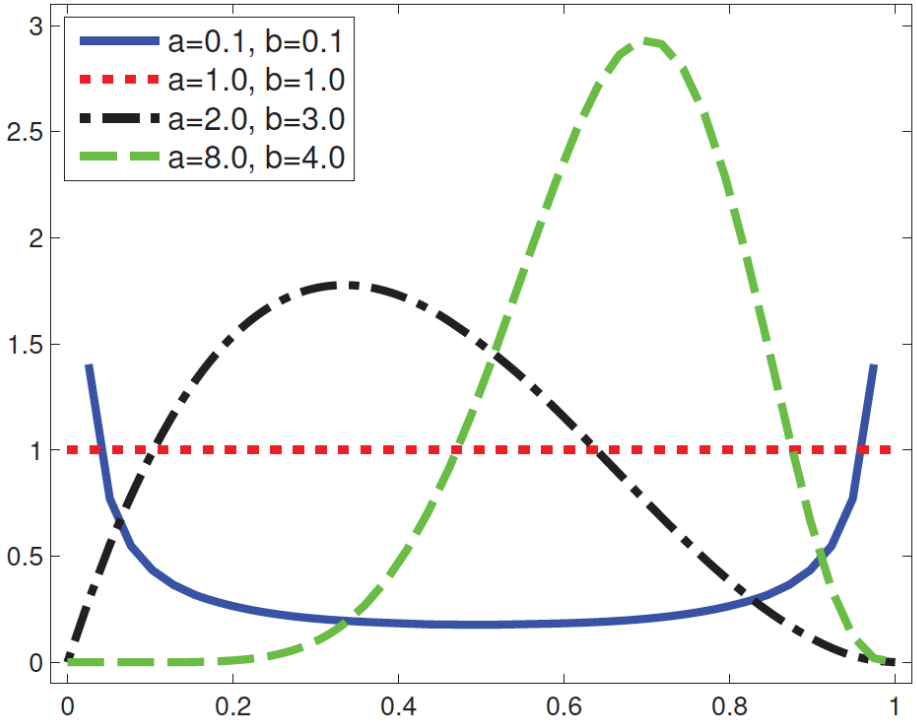
$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$$

$$\Gamma(n) = (n-1)! \text{ if } n \in \mathbb{Z}^+$$

Let's Model θ as a Random Variable

- Consider N coin tosses: data $D = (N_0, N_1)$, where $N_0 = \#$ tails, $N_1 = \#$ heads in N coin tosses, $\theta =$ probability of head, then
  - $p(D|\theta) = \text{Bin}(N_1|N, \theta) = \binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}$ (binomial likelihood)
- Assume beta prior on θ: $p(\theta|a, b) = \text{Beta}(\theta|a, b) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$
  - $a > 0, b > 0$ called hyperparameters



- $a = b = 1 \implies$ uniform
- Bigger $a, b \implies$ extreme values less likely
- $a > b \implies \theta$ likely to be > 0.5

Beta-Binomial Model

- Consider N coin tosses: data $D = (N_0, N_1)$, where $N_0 = \#$ tails, $N_1 = \#$ heads in N coin tosses, $\theta =$ probability of head, then
  - $p(D|\theta) = \text{Bin}(N_1|N, \theta) = \binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}$ (binomial likelihood)
- Assume beta prior on θ: $p(\theta|a, b) = \text{Beta}(\theta|a, b) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$
  - $a > 0, b > 0$ called hyperparameters
- Posterior

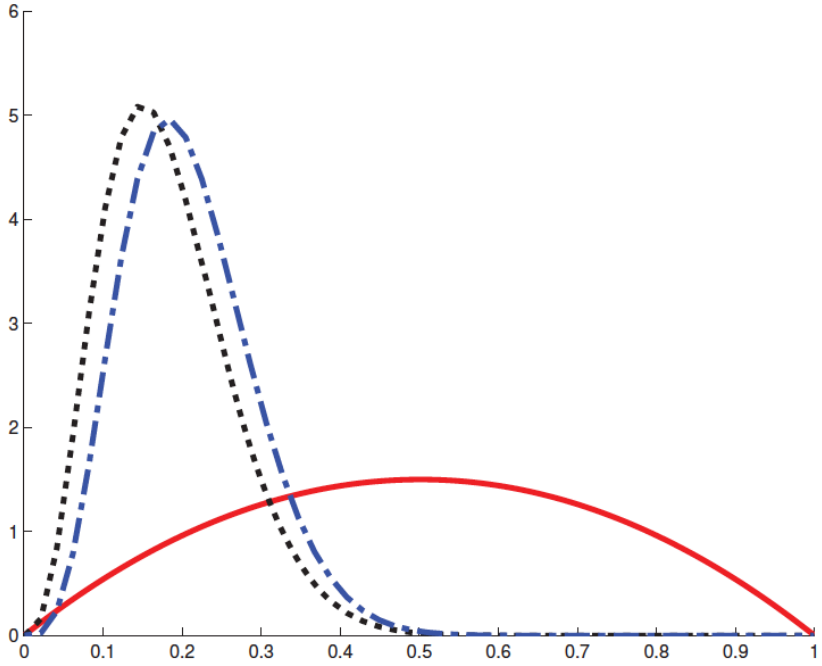
$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} = \frac{\binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}\frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}}{p(D)}$$

$$= \binom{N}{N_1}\frac{1}{B(a,b)}\frac{B(N_1+a, N_0+b)}{B(N_1+a, N_0+b)}\frac{\theta^{N_1+a-1}(1-\theta)^{N_0+b-1}}{p(D)}$$

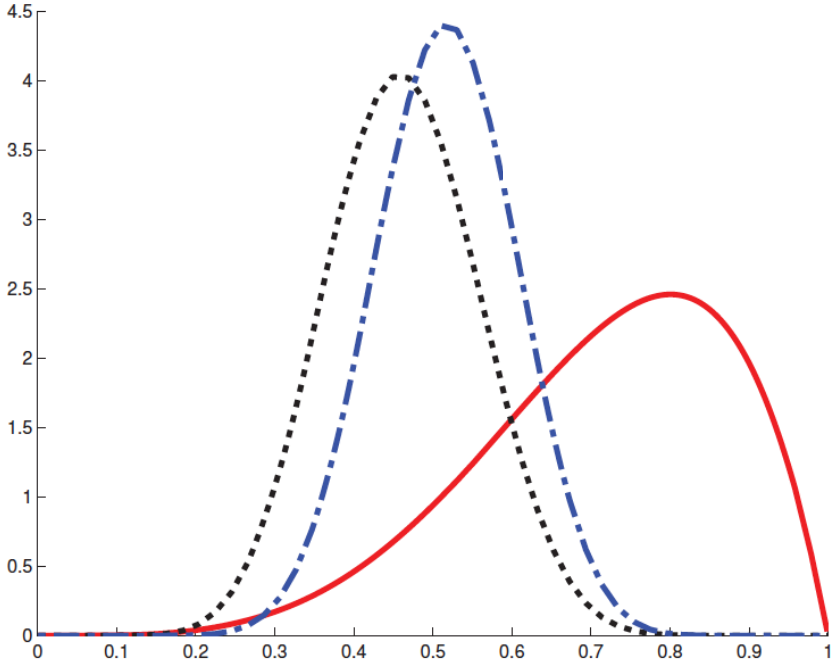
$$= \frac{\kappa(N_0, N_1, a, b)\text{Beta}(\theta|N_1+a, N_0+b)}{p(D)} = \text{Beta}(\theta|N_1+a, N_0+b)$$

  - Posterior obtained by adding prior hyperparameters to observed counts. Therefore a, b called pseudo-counts
  - Posterior same form as prior $\implies$ beta is conjugate prior of binomial

Beta-Binomial Model



Prior: $\text{Beta}(2, 2)$
Likelihood: Binomial: $N_1 = 3, N_0 = 17$
Posterior: $\text{Beta}(5, 19)$



Prior: $\text{Beta}(5, 2)$
Likelihood: Binomial: $N_1 = 11, N_0 = 13$
Posterior: $\text{Beta}(16, 15)$

Observe posterior tends to be “between” likelihood and prior

MAP, ML Estimates

- For $\text{Beta}(x|c,d)$, mode $= \frac{c-1}{c+d-2}$
- Posterior $p(\theta|D) = \text{Beta}(\theta|N_1 + a, N_0 + b)$
- Maximum-A-Posteriori (MAP) estimate

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmax}}\, p(\theta|D) = \frac{N_1 + a - 1}{N + a + b - 2}$$

- Maximum-likelihood (ML) estimate

$$\hat{\theta}_{ML} = \frac{N_1}{N} \text{ by setting a = b = 1 (uniform prior)}$$

MAP Trades Off Prior and Observations

- $\hat{\theta}_{MAP} = \frac{N_1+a-1}{N+a+b-2}$ and $\hat{\theta}_{ML} = \frac{N_1}{N}$
- Let $m_0 = \frac{a-1}{a+b-2}$ (prior mode), $\alpha_0 = a + b - 2$ (pseudocounts),

$$\begin{aligned}\hat{\theta}_{MAP} &= \frac{N_1 + a - 1}{N + a + b - 2} = \frac{N_1 + m_0\alpha_0}{N + \alpha_0} \\ &= \left(\frac{N}{N + \alpha_0}\right)\frac{N_1}{N} + \left(\frac{\alpha_0}{N + \alpha_0}\right)m_0 \\ &= (1 - \lambda)\hat{\theta}_{ML} + \lambda m_0\end{aligned}$$

- $\hat{\theta}_{MAP}$ is convex combination of $\hat{\theta}_{ML}$ and prior mode
  - Weaker prior (α_0 small) $\implies$ $\hat{\theta}_{MAP}$ closer to $\hat{\theta}_{ML}$
  - As $N \to \infty$ $\implies$ $\hat{\theta}_{MAP} \to \hat{\theta}_{ML}$

Predicting Future Coin Tosses

- Suppose we are given training data $D = \{N_0 \text{ tails}, N_1 \text{ heads}\}$
- Let $\tilde{x}$ be outcome of next coin toss. What is the probability that $\tilde{x} = 1$?
- Strategy 1 (ML estimation): $p(\tilde{x} = 1) = \theta_{ML}$
- Strategy 2 (MAP estimation): $p(\tilde{x} = 1) = \theta_{MAP}$
- Strategy 3 (Posterior Predictive Distribution)
  - Problem with ML/MAP strategies is that they are not quite optimal
  - Above question says "Suppose we are given D, what is probability of $\tilde{x} = 1$", so the probability we want to evaluate is $p(\tilde{x} = 1|D)$, which is not equal to θ_{ML} or θ_{MAP}

Posterior Predictive Distribution

- For $\text{Beta}(x|c,d)$, mean $= \frac{c}{c+d}$
- Posterior $p(\theta|D) = \text{Beta}(\theta|N_1 + a, N_0 + b)$. $\tilde{x}$ = outcome of single future coin toss

$$p(\tilde{x} = 1|D) = \int_0^1 p(\tilde{x} = 1, \theta|D)d\theta = \int_0^1 p(\tilde{x} = 1|\theta, D)p(\theta|D)d\theta$$

$$= \int_0^1 p(\tilde{x} = 1|\theta)p(\theta|D)d\theta = \int_0^1 \theta p(\theta|D)d\theta$$

$$= E(\theta|D) = \frac{N_1 + a}{N + a + b}$$

- As $N \to \infty$, $p(\theta|D)$ becomes delta function at $\hat{\theta}_{MAP}$

$$p(\tilde{x} = 1|D) \approx \int_0^1 \theta\delta(\theta - \hat{\theta}_{MAP})d\theta = \hat{\theta}_{MAP}$$

  - Called plug-in approximation to posterior predictive density
  - Extremely common: equivalent to getting point estimate from training data and applying to test data, but can overfit

Black Swan Paradox (overfitting)

- Instead of posterior predictive, suppose $p(\tilde{x} = 1|D) \approx \hat{\theta}_{ML} = \frac{N_1}{N}$
  - For small sample size, $\hat{\theta}_{ML}$ can perform poorly
  - e.g., $N = 3$ and all are tails, then $\hat{\theta}_{ML} = 0 \implies$ heads impossible for future coin toss
- Even in big data era, once we partition data based on certain criteria, (e.g., number of times specific person engaged specific activity), sample size can become very small
- Black swan paradox: In ancient times, only white swans observed. Using ML estimate, this would imply black swan can never be observed. But actually black swans discovered in Australia in 17th century!
  - Can resolve paradox with posterior predictive distribution, e.g., $a = b = 1$, then probability of observing black swan is $p(\tilde{x} = 1|D) = \frac{N_1+1}{N_1+N_0+2}$, which is non-zero (Laplace Smoothing)
  - For $a = 1, b = 1$, $\hat{\theta}_{MAP} = \hat{\theta}_{ML}$, so MAP will not resolve problem either

Interlude: Baseball Example

- How is this useful?
- How to set the hyperparameters of beta distribution?
  - Empirical Bayes

http://varianceexplained.org/r/empirical_bayes_baseball/

Baseball Recruiter Problem

- Suppose I am a baseball recruiter, trying to choose two potential players. Player one has achieved 4 hits in 10 chances. Player 2 has achieved 300 hits in 1000 chances. Who should I recruit?
- The typical baseball player hits roughly 25.9% of the time.
- Player one has a higher proportion of hits (4/10), but it could be luck since he has only batted 10 times.
- Player two has a lower proportion of hits (3/10), but there is a lot of evidence that he is better than average (25.9%)

http://varianceexplained.org/r/empirical_bayes_baseball/

Best & Worse Batters in History?

- Players with highest average

Name	# Hits	#Total Tries	Average
Jeff Banister	1	1	1
Doc Bass	1	1	1
Steve Biras	2	2	1
C. B. Burns	1	1	1
Jackie Gallagher	1	1	1

  - These aren't the best batters, they're just the batters who went up once or twice and got lucky.

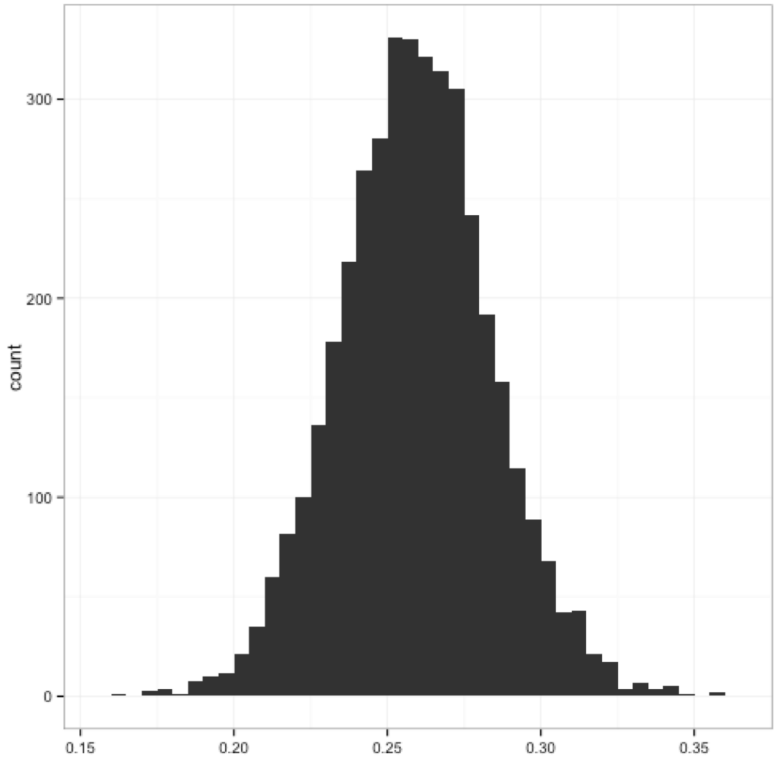
- Players with lowest average

Name	# Hits	#Total Tries	Average
Frank Abercrombie	0	4	0
Horace Allen	0	7	0
Pete Allen	0	4	0
Walter Alston	0	1	0
Bill Andrus	0	9	0

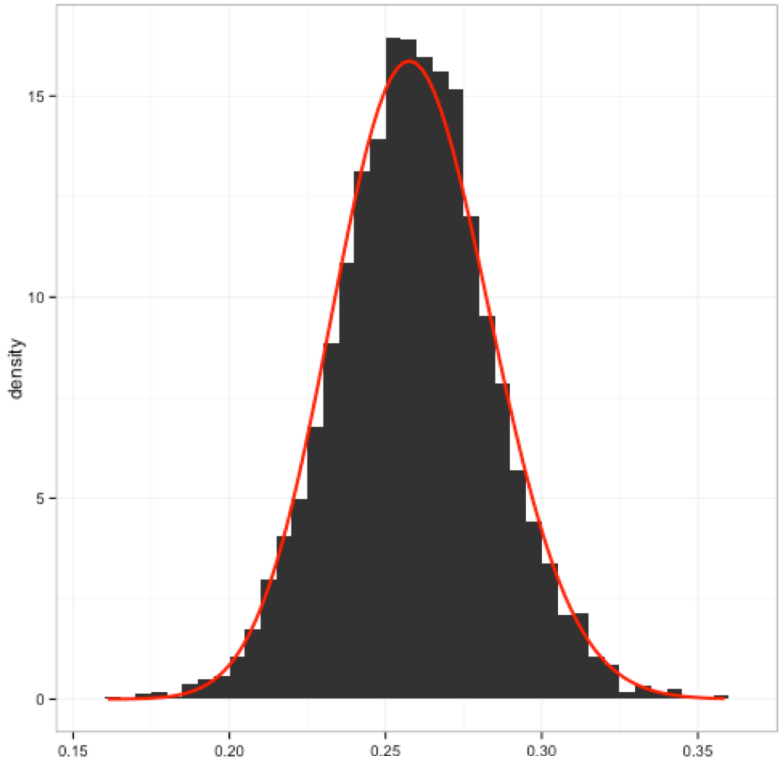
http://varianceexplained.org/r/empirical_bayes_baseball/

Estimate Prior Distribution

Histogram of Batting Averages Across Players



Beta distribution (a = 78.7, b = 224.9)



http://varianceexplained.org/r/empirical_bayes_baseball/

Baseball Recruiter Problem

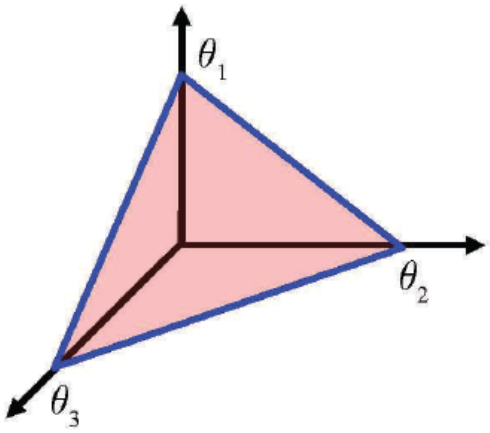
- Suppose I am a baseball recruiter, trying to choose two potential players. Player one has achieved 4 hits in 10 chances. Player 2 has achieved 300 hits in 1000 chances. Who should I recruit?
- Now we have Beta distribution (a = 78.7, b = 224.9) as a prior
  - Note that mean = a/(a+b) = 0.259
- Posterior predictive distribution for player one: (4 + a)/(10+a+b) = 0.263
- Posterior predictive distribution for player two: (300+a)/(1000+a+b) = 0.291
- Therefore, we should recruit player two

http://varianceexplained.org/r/empirical_bayes_baseball/

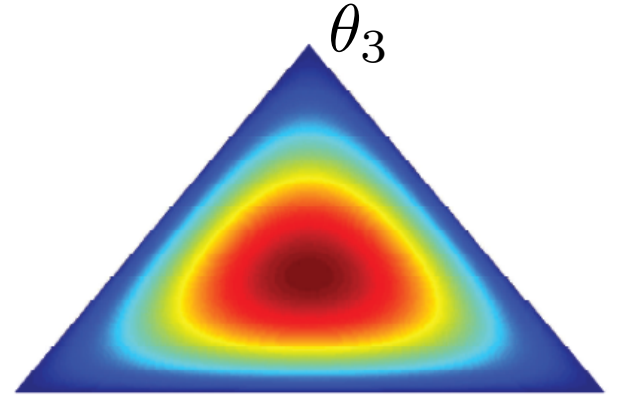
Dirichlet-Multinomial Generative Model (Multiple Dice Rolling)

Dirichlet-Multinomial Model

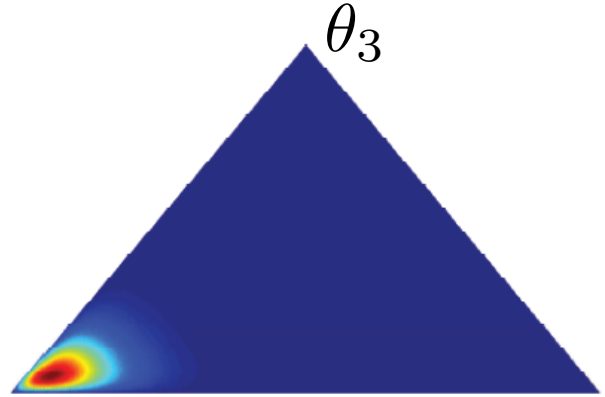
- Consider N rolls of K-sided dice. Let $D = \{N_k\}_{k=1:K}$, $N_k = \#$ times k shows up on dice. Let $\theta = \{\theta_k\}_{k=1:K}$, $\theta_k =$ probability of k showing up on dice, then
  - $p(D|\theta) = \frac{N!}{N_1! \cdots N_K!} \prod_{k=1}^{K} \theta_k^{N_k} \propto \prod_{k=1}^{K} \theta_k^{N_k}$ (multinomial likelihood)
- Dirichlet prior with hyperparameters $\alpha = \{\alpha_k\}_{k=1:K}$, $\alpha_k > 0$
  - $p(\theta|\alpha) = \mathrm{Dir}(\theta|\alpha) = \begin{cases} \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} & \text{if } \theta_k \geq 0, \sum_{k=1}^{K} \theta_k = 1 \\ 0 & \text{otherwise} \end{cases}$



$\theta_k \geq 0, \sum_{k=1}^{K} \theta_k = 1$
θ called simplex



$\alpha = (2, 2, 2)$



$\alpha = (20, 2, 2)$

Dirichlet-Multinomial Model

- Consider N rolls of K-sided dice. Let $D = \{N_k\}_{k=1:K}$, $N_k = \#$ times k shows up on dice. Let $\theta = \{\theta_k\}_{k=1:K}$, $\theta_k =$ probability of k showing up on dice, then
  - $p(D|\theta) = \frac{N!}{N_1! \cdots N_K!} \prod_{k=1}^{K} \theta_k^{N_k} \propto \prod_{k=1}^{K} \theta_k^{N_k}$ (multinomial likelihood)
- Dirichlet prior with hyperparameters $\alpha = \{\alpha_k\}_{k=1:K}$, $\alpha_k > 0$
  - $p(\theta|\alpha) = \text{Dir}(\theta|\alpha) = \begin{cases} \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} & \text{if } \theta_k \geq 0, \sum_{k=1}^{K} \theta_k = 1 \\ 0 & \text{otherwise} \end{cases}$
- Dirichlet Posterior

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k - 1}$$
$$= \text{Dir}(\theta|N_1 + \alpha_1, \cdots, N_K + \alpha_K)$$

  - Posterior obtained by adding prior hyperparameters to observed counts. Therefore $\alpha = \{\alpha_k\}_{k=1:K}$ called pseudo-counts
  - Posterior same form as prior $\implies$ Dirchlet is conjugate prior of multinomial

MAP, ML Estimates

- For $\mathrm{Dir}(x|\gamma = \gamma_1, \cdots, \gamma_K)$, mode $x_k = \frac{\gamma_k - 1}{\sum_k \gamma_k - K}$
- Posterior $p(\theta|D) = \mathrm{Dir}(\theta|N_1 + \alpha_1, \cdots, N_K + \alpha_K)$
- Maximum-A-Posteriori (MAP) estimate $\hat{\theta}_{MAP} = \mathrm{argmax}_\theta \, p(\theta|D)$

$$\hat{\theta}_k^{MAP} = \frac{N_k + \alpha_k - 1}{N + \sum_k \alpha_k - K}$$

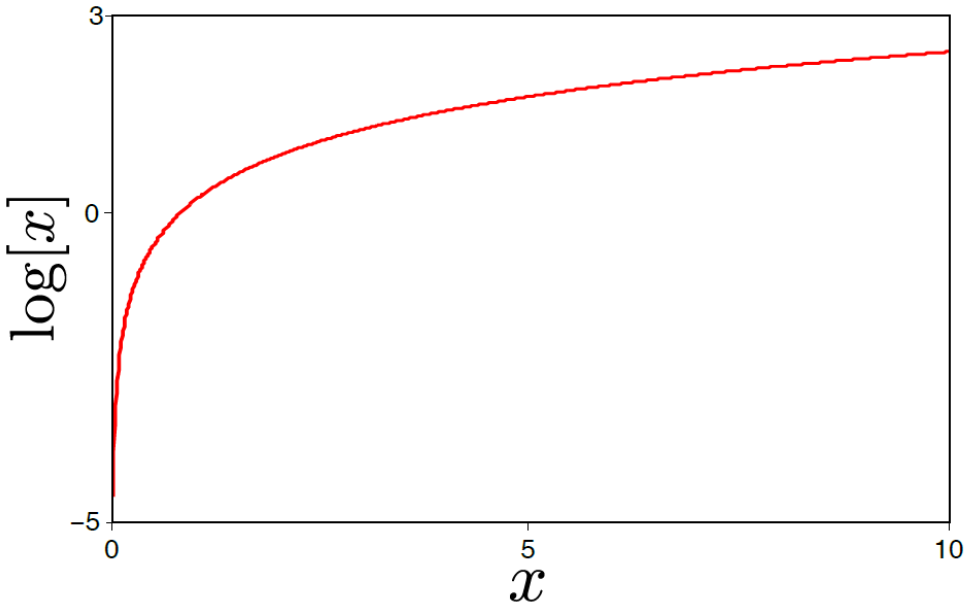
- Maximum-likelihood (ML) estimate

$$\hat{\theta}_k^{ML} = \frac{N_k}{N} \text{ by setting } \alpha_k = 1 \text{ (uniform prior)}$$

Derivation of Dirichlet Mode

- $\mathrm{Dir}(\theta|\alpha = \alpha_1, \cdots, \alpha_K)$, want to show mode $\theta_k = \frac{\alpha_k - 1}{\sum_k \alpha_k - K}$

$$\theta_{mode} = \underset{\theta}{\operatorname{argmax}} \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \quad \text{s.t.} \quad \sum_k \theta_k = 1, \theta_k \geq 0$$



The logarithm is a monotonic transformation.
Hence, the position of the peak stays in the same place
But the log probability easier to work with

Derivation of Dirichlet Mode

- $\text{Dir}(\theta|\alpha = \alpha_1, \cdots, \alpha_K)$, want to show mode $\theta_k = \frac{\alpha_k - 1}{\sum_k \alpha_k - K}$

$$\theta_{mode} = \underset{\theta}{\text{argmax}} \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \quad \text{s.t.} \quad \sum_k \theta_k = 1, \theta_k \geq 0$$

$$= \underset{\theta}{\text{argmax}} \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_k \quad \text{s.t.} \quad \sum_k \theta_k = 1, \theta_k \geq 0$$

- Lagrangian: $l(\theta, \lambda) = \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_k + \lambda(1 - \sum_k \theta_k)$
  - Want to solve for $\frac{\partial l}{\partial \lambda} = \frac{\partial l}{\partial \theta_1} = \cdots = \frac{\partial l}{\partial \theta_K} = 0$
  - Note that $\frac{\partial l}{\partial \lambda} = (1 - \sum_k \theta_k) = 0$ yields original constraint
- $\frac{\partial l}{\partial \theta_i} = \frac{\alpha_i - 1}{\theta_i} - \lambda = 0 \implies \theta_i = \frac{1}{\lambda}(\alpha_i - 1)$
  - Since $\sum_i \theta_i = 1 \implies \sum_i \frac{1}{\lambda}(\alpha_i - 1) = 1 \implies \lambda = \sum_i \alpha_i - K$
  - Therefore $\theta_k = \frac{\alpha_k - 1}{\sum_k \alpha_k - K}$

Predicting Future Dice Rolls

- Suppose we are given training data $D = \{N_1, ..., N_k\}$ dice rolls
- Let $\tilde{x}$ be outcome of next dice roll. What is the probability that $\tilde{x} = i$?
- Strategy 1 (ML estimation): $p(\tilde{x} = i) = \theta^i_{ML}$
  - Note that θ_{ML} is a vector, so θ^i_{ML} is the i-th element of vector
- Strategy 2 (MAP estimation): $p(\tilde{x} = i) = \theta^i_{MAP}$
- Strategy 3 (Posterior Predictive Distribution)
  - Problem with ML/MAP strategies is that they are not quite optimal
  - Above question says "Suppose we are given D, what is probability of $\tilde{x} = i$", so the probability we want to evaluate is $p(\tilde{x} = i|D)$, which is not exactly equal to θ^i_{ML} or θ^i_{MAP}

Posterior Predictive Distribution

- For $\text{Dir}(x|\gamma_1, \cdots, \gamma_K)$, mean $E(x_i) = \frac{\gamma_i}{\sum_k \gamma_k}$
- Posterior $p(\theta|D) = \text{Dir}(\theta|N_1 + \alpha_1, \cdots, N_K + \alpha_K)$. $\tilde{x}$ = outcome of single future dice roll

$$
\begin{aligned}
p(\tilde{x} = i|D) &= \int p(\tilde{x} = i, \theta|D) d\theta \quad \text{(multi-dimensional integral over simplex)} \\
&= \int p(\tilde{x} = i|\theta, D) p(\theta|D) d\theta \\
&= \int p(\tilde{x} = i|\theta) p(\theta|D) d\theta \\
&= \int \theta_i p(\theta|D) d\theta \\
&= E(\theta_i|D) \\
&= \frac{N_i + \alpha_i}{N + \sum_k \alpha_k}
\end{aligned}
$$

Conditional Independence

- Posterior predictive distribution even more important in multinomial than binomial case because data more sparse as we partition data into more categories

Bag of words example

- Apply Dirichlet-multinomial model to predict next word in text sequence.
- Text sequence: `Mary had a little lamb, little lamb, little lamb, Mary had a little lamb, its fleece as white as snow`
- Dictionary:

mary	lamb	little	big	fleece	white	black	snow	rain	unk
1	2	3	4	5	6	7	8	9	10

  - "unk" stands for unknown (e.g., "had", "its")
- Ignore punctuation and stop words (e.g., a, as, the), count word frequency

Token	1	2	3	4	5	6	7	8	9	10
Word	mary	lamb	little	big	fleece	white	black	snow	rain	unk
Count	2	4	4	0	1	1	0	1	0	3

- For $\alpha_i = 1$, we have $p(\tilde{x} = i) = \frac{N_i + \alpha_i}{N + \sum_k \alpha_k} = \frac{N_i + 1}{16 + 10}$

$$p(\tilde{x}|D) = \left(\frac{3}{26}, \frac{5}{26}, \frac{5}{26}, \frac{1}{26}, \frac{2}{26}, \frac{2}{26}, \frac{1}{26}, \frac{2}{26}, \frac{1}{26}, \frac{4}{26}\right)$$

  - Most likely words are "little" and "lamb"
  - "big", "black", "rain" non-zero probability even though not observed

Summary

- Probabilistic Signal Detection / Estimation
  - MAP & ML estimation
  - Two approaches of modeling posterior distribution: Generative & Discriminative Models
  - Parameters of probabilistic models can themselves be treated as random variables and estimated accordingly
- Conjugate priors
  - Beta-Binomial generative model (binomial parameters are beta distributed)
  - Dirichlet-Multinomial generative models (multinomial parameters are dirichlet distributed)
- Predicting future coin toss / dice rolls
  - Posterior predictive distribution, Plug-In Approximation, Black Swan

Optional Reading

- SP Chapters 4.5
- KM Chapter 3.1 to 3.4
  - More examples in the chapter
  - Beware of typos