Graphical Models For Complex Health Data (P8124)

Daniel Malinsky

Columbia University
dsm2128@cumc.columbia.edu

Parameter Learning/Estimation

Parameter learning (estimation)

In recent weeks we've considered abstract and mostly "non-parametric" properties of graphical models. This week we'll consider associating parametric models with graphs and learning/estimating the parameters for a **given** graph structure.

We'll talk about Bayesian networks and MRFs. BNs are quite straightforward, because the problem decomposes into a set of local learning problems. MRFs can seem more complicated, because the problem does not decompose nicely. However, some MRF models (Gaussian) have convenient parameterizations that make them easy to estimate with well-studied algorithms.

Maximum likelihood

Consider n iid observations of a single random variable $X^1, ..., X^n$, for example n tosses of a coin which may land $X^j \in \{H, T\}$. Let $\theta = p(X = H)$. For an observation sequence $\{H, H, T, H, T\}$ we calculate the probability of the observed data as a function of the unknown parameter θ:

$$\mathcal{L} \equiv \prod_{j=1}^{n} p(x^j) = \theta\theta(1-\theta)\theta(1-\theta)$$

The maximum of the likelihood is acheived at $\hat{\theta} = 0.60$.

Generally: $\hat{\theta}_{MLE} = \frac{\#heads}{\#heads+\#tails}$

Why the MLE?

From one perspective, we may prefer the MLE because it has some nice properties. These properties are very well-studied and typically form a large chunk of a class in estimation theory. Here we note that:

- Under certain conditions, the MLE is consistent, i.e., $\hat{\theta} \rightarrow^P \theta^*$
- The MLE is asymptotically normal: $\sqrt{n}(\hat{\theta} - \theta^*) \rightarrow N(0, \sigma^2_{MLE})$
- Among a certain class of estimators, the MLE is efficient, i.e., the asymptotic variance σ^2_{MLE} is as small as possible

These are all desirable properties, from a frequentist perspective. Note that there are important background conditions lurking here: the MLE is not consistent for all problems!

Why the MLE?

Here is another perspective. Consider the distribution at the true value of the parameter, θ^*. Recall the Kullback-Leibler (KL) divergence:

$$\begin{aligned} D_{KL}(p(x;\theta^*)||p(x;\theta)) &= \sum_x p(x;\theta^*) \log \frac{p(x;\theta^*)}{p(x;\theta)} \\ &= -H(p(x;\theta^*)) - \mathbb{E}[\log p(x;\theta)] \end{aligned}$$

The first term doesn't depend on θ and the second term is wrt $p(x;\theta^*)$. Maximizing the expected likelihood (asymptotically) minimizes KL-divergence from the true distribution.
We approximate this expectation by

$$\mathbb{E}[\log p(x;\theta)] \approx \frac{1}{n} \sum_{j=1}^{n} \log p(x^j;\theta)$$

Loss functions

Finally, we can view the situation from the perspective of minimizing some loss function $L(x, p)$ that measures the "cost" of using model distribution p on a particular instance x. (Common in ML literature.)

Our goal, from this perspective, is to find the model which minimizes the expected loss $\mathbb{E}[L(x, p)] \approx \frac{1}{n} \sum_{j=1}^{n} L(x, p)$. The expected loss $\mathbb{E}[L(x, p)]$ is also known as the *risk* and $\frac{1}{n} \sum_{j=1}^{n} L(x, p)$ the *empirical risk*. Notice this expectation is wrt the true distribution $p(x; \theta^*)$.

MLE corresponds to using the log-loss: $L(x, p) \equiv -\log p(x)$.

Other loss functions can be defined appropriate to specific tasks, for example the classification error ("0/1 loss") for a classification task: $\mathbb{E}[\mathbb{I}\{\exists y' \neq y : p(y'|x) \geq p(y|x)\}]$

Discrete Bayesian networks

Consider a BN model with $p(x) = \prod_{i=1}^{p} \theta_{x_i|x_{\text{Pa}(i)}}$. We have n observations $x^1, ..., x^n$ and likelihood:

$$\mathcal{L} = \prod_{j=1}^{n} \prod_{i=1}^{p} \theta_{x_i^j|x_{\text{Pa}(i)}^j}$$

Taking logs, setting derivative to zero...

$$\hat{\theta}_{x_i|x_{pa(i)}} = \frac{\#(x_i, x_{\text{Pa}(i)})}{\#(x_{\text{Pa}(i)})}$$

Discrete Bayesian networks

So, we have a closed-form solution for the MLE.

The likelihood decomposes into product of terms which can be maximized independently (global decomposability).

Note: the number of data points used to estimate $\theta_{x_i|x_{\text{Pa}(i)}}$ is $\#(x_{\text{Pa}(i)})$. As the number of parents grows, the number of different parent assignments grows exponentially. Therefore, the number of available data points for each parameter shrinks exponentially. This is called "data fragmentation" and is a serious limit to learning discrete BNs. The problem is more severe when the parents can take on a large number of different values.

$\Rightarrow$ In practice, may limit the number of parents for each node (enforces "sparsity," likely introduces bias).

Discrete pairwise MRFs

Consider the pairwise model

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{i\sim k} \theta_{ik} x_i x_k\right)$$

where $Z(\theta) = \sum_x \exp\left(\sum_{i\sim k} \theta_{ik} x_i x_k\right)$

Here I'm following Hastie et al. Sec. 17.4. They assume a "constant node" $X_0 \equiv 1$ with edges to all other nodes, which gives the terms $\theta_i x_i \ \forall i$.

Discrete pairwise MRFs

The log-likelihood is:

$$\ell(\theta) = \log \mathcal{L} = \sum_{j=1}^{n} \left[\sum_{i \sim k} \theta_{ik} x_i^j x_k^j - \Phi(\theta) \right]$$

where $\Phi(\theta) = \log \sum_x \exp\left(\sum_{i \sim k} \theta_{ik} x_i x_k\right)$

The gradient is:

$$\frac{\partial \ell(\theta)}{\partial \theta_{ik}} = \sum_{j=1}^{n} x_i^j x_k^j - n \frac{\partial \Phi(\theta)}{\partial \theta_{ik}}$$

$$\frac{\partial \Phi(\theta)}{\partial \theta_{ik}} = \sum_x x_i x_k \cdot p(x; \theta) = E_\theta[X_i X_k]$$

Discrete pairwise MRFs

So the MLE is the solution to:

$$\frac{1}{n}\sum_{j=1}^{n} x_i^j x_k^j - E_\theta[X_i X_k] = 0 \qquad \text{“moment matching”}$$

How do we solve these equations? The simplest algorithm is iterative proporitional fitting (IPF):

- Initialize the parameter vector $\theta^0 = (1, ..., 1)$
- For each θ_{ik}, update the value by holding rest fixed:
  $\theta_{ik}^{t+1} = \theta_{ik}^t \cdot \frac{\frac{1}{n}\sum_{j=1}^{n} x_i^j x_k^j}{E_{\theta^t}[X_i X_k]}$
- Repeat until convergence
- Can prove that each step improves $\ell(\theta^t)$ and converges to MLE
- Not very efficient for large graphs

More general categorical MRFs

The same general strategy works for categorical MRFs beyond pairwise MRFs: log-linear models w/ larger cliques (need to iterative over cliques) and with more than 2 categories for each variable.

The gradient equations will always look like a "moment matching" condition, this is generally true for exponential family models where sufficient statistics are set equal to their expectations under the model. E.g. starting from:

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{C \in \mathcal{C}} \theta_C^T \psi_C(x_C)\right)$$

MLE will need to solve:

$$\frac{1}{n}\sum_{j=1}^{n} \psi_C(x_C^j) - E_\theta[\psi_C(x_C)] = 0$$

Obtaining MLE

However, for large graphs IPF is slow (why?)...

... so approximate procedures have been developed (based on mean-field equations or Gibbs sampling, to be discussed later)

Gaussian graphical models

Gaussian distributions have special properties:

- $X_i \perp\!\!\!\perp X_j | X_S \iff \rho_{ij.S} = 0$
- Only "second moment" information, no higher-order dependencies (everything is pairwise)
- (μ, Σ) completely describes the distribution
- "closed under marginalization and conditioning"

In Lecture0, we saw that $\rho_{ij} = \frac{cov(X_i, X_j)}{\sigma_i \sigma_j}$

$\rho_{ij.S} = \frac{\rho_{ij.S_0} - \rho_{is.S_0}\rho_{js.S_0}}{\sqrt{1-\rho^2_{is.S_0}}\sqrt{1-\rho^2_{js.S_0}}}$ where $s \in S$ and $S_0 = S \setminus s$ [recursive definition]

Gaussian Bayesian networks

$$p(x_i|\,\text{Pa}(X_i,\mathcal{G})) \sim N(\beta_0 + \beta_{i_1} x_{i_1} + ... + \beta_{i_k} x_{i_k}, \sigma_i^2)$$

$$\text{where } \text{Pa}(X_i,\mathcal{G}) = \{X_{i_1}, ..., X_{i_k}\} \text{ and } \forall i \in V.$$

[note that each X_i may have a different number of parents i_k]

$$\ell_i \equiv \log \mathcal{L}_i = \sum_{j=1}^{n} \left[-\frac{1}{2}\log(2\pi\sigma_i^2) - \frac{1}{2}\frac{1}{\sigma_i^2}(\beta_0 + \beta_{i_1} x_{i_1}^j + ... + \beta_{i_k} x_{i_k}^j - x_i^j)^2 \right]$$

MLE parameters can be obtained by ordinary linear regression of each X_i on $\text{Pa}(X_i,\mathcal{G})$.

Gaussian MRFs

$$p(x;\mu,\Sigma) = (2\pi)^{-p/2}(\det\Sigma)^{-1/2}\exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

It is convenient to write the multivariate Gaussian distribution as a function of the precision matrix $K = \Sigma^{-1}$.

$$p(x;\mu,K) = \exp\{\mu^T Kx - tr(K,\frac{1}{2}xx^T) - \frac{p}{2}\log(2\pi) + \frac{1}{2}\log\det(K) - \frac{1}{2}\mu^T K\mu\}$$

This connects the distribution directly to independence constraints since

$$K_{ij} = 0 \iff X_i \perp\!\!\!\perp X_j | \{X \setminus \{X_i, X_j\}\}$$

Gaussian distributions

Let $X \sim N(\mu, \Sigma)$ and partition $X = (X_A, X_B)$. Write $\mu = (\mu_A \;\; \mu_B)'$ and $\Sigma = \begin{pmatrix} \Sigma_{A,A} & \Sigma_{A,B} \\ \Sigma_{B,A} & \Sigma_{B,B} \end{pmatrix}$

(1) The marginal distribution of X_A is $N(\mu_A, \Sigma_{A,A})$

(2) The conditional distribution of $X_A | X_B = x_B$ is $N(\mu_{A|B}, \Sigma_{A|B})$ where

$$\mu_{A|B} = \mu_A + \Sigma_{A,B}\Sigma_{B,B}^{-1}(x_B - \mu_B)$$

and

$$\Sigma_{A|B} = \Sigma_{A,A} - \Sigma_{A,B}\Sigma_{B,B}^{-1}\Sigma_{B,A}$$

"closed under marginalization and conditioning"

Note: some matrix algebra (Schur complements) yields $K_{A,A} = \Sigma_{A|B}^{-1}$.

Gaussian distributions

Theorem. Let $X \sim N(\mu, \Sigma)$. Then

(1) $X_i \perp\!\!\!\perp X_j$ iff $\Sigma_{ij} = 0$

(2) $X_i \perp\!\!\!\perp X_j | \{X \setminus \{X_i, X_j\}\}$ iff $K_{ij} = 0$

(1) follows from previous slide. (2) follows from applying previous slide with $A = \{i, j\}$ and $B =$ rest so $\Sigma_{\{i,j\}|\text{rest}} = (K_{\{i,j\},\{i,j\}})^{-1}$ and $X_i \perp\!\!\!\perp X_j | \{X \setminus \{X_i, X_j\}\}$ iff $\Sigma_{\{i,j\}|\text{rest}} = (K_{\{i,j\},\{i,j\}})^{-1}$ is diagonal or equivalently $K_{ij} = 0$.

Gaussian MRFs

The Gaussian log-likelihood can be written

$$\ell(\mu, K) \propto \frac{n}{2} \log\det(K) - \frac{n}{2} tr(SK) - \frac{n}{2}(\bar{x} - \mu)^T K(\bar{x} - \mu)$$

where $S = \frac{1}{n}\sum_{j=1}^{n}(x^j - \bar{x})(x^j - \bar{x})^T$.

Since a given MRF $\mathcal{G}$ does not constrain μ we have that $\hat{\mu} = \bar{x}$ and focus on estimating K.

Recall that $X_i \not\perp X_j$ in $\mathcal{G}$ iff $K_{ij} = 0$. So our optimization problem reduces to:

maximize: $\log\det(K) - tr(SK)$
subject to: $K_{ij} = 0$ for $X_i \not\perp X_j$ in $\mathcal{G}$

Gaussian MRFs

maximize: $\log\det(K) - tr(SK)$
subject to: $K_{ij} = 0$ for $X_i \not\perp X_j$ in $\mathcal{G}$

There are many ways of tackling this convex optimization problem. One way is to use **coordinate descent**. You give an initial guess to the parameters and then update one parameter at a time (there exists a closed-form solution) while holding the rest fixed.

MLE with coordinate descent

Algorithm 9.2 Coordinate descent on K

Input: Graph $G = (V, E)$, sample covariance matrix S, and precision ϵ.
Output: MLE $\hat{K}$.

1: Let $K^0 = \mathrm{Id}$.
2: Cycle through $(u, v) \in E$ and solve the following optimization problem:

$$\begin{aligned} \underset{K \succeq 0}{\text{maximize}} \quad & \log\det(K) - \operatorname{trace}(KS) \\ \text{subject to} \quad & K_{i,j} = K^0_{i,j} \ \text{ for all } (i,j) \in (V \times V) \setminus \{(u,u),(v,v),(u,v)\} \end{aligned}$$

and update $K^1 := K$.
3: **if** $\|K^0 - K^1\|_1 < \epsilon$ **then**
4: let $\hat{K} := K^1$
5: **else**
6: let $K^0 := K^1$ and return to line 2.
7: **end if**

Solution to max prob in step 2:

$$K_{A,A} = (S_{A,A})^{-1} + K_{A,B} K_{B,B}^{-1} K_{B,A}$$

where $A = \{u, v\}$ and $B = V \setminus A$

Faster optimization algorithms

There are more sophisticated methods that are designed to scale up to very large graphs. Quasi-Newton methods (e.g. BFGS or L-BFGS in `optim()` function in R) use an approximation to the Hessian of the objective function and various numerical tricks to make the problem feasible in high-dimensions.[1]

And for some special cases of graph structures (e.g. *chordal* graphs) there exists a closed form solution for the entire matrix $\hat{K}$.

[1] See Uhler (2018) "Gaussian graphical models: An algebraic and geometric perspective" in *Handbook of Graphical Models*, CRC Press. Also see this tech report for a comparison of optimization methods: here [link]

Confidence intervals for MLE estimates

$\hat{\theta}_{MLE}$ is one number for any given data set, it represents a "guess" at what is the true parameter value, θ^*. But, how do we represent our uncertainty about the estimate?

Instead of producing only a single value, we may produce an interval estimate: a confidence interval. We then report the interval $\hat{C} \equiv (\hat{\theta}_{MLE} - \hat{\delta}, \hat{\theta}_{MLE} + \hat{\delta})$ for some $\hat{\delta}$ estimated from the data. Note that $\hat{C}$ is a random quantity (a function of random variables), just like $\hat{\theta}$ is!

How do we determine $\hat{\delta}$, and what is the right "length" of the interval?

Confidence intervals for MLE estimates

Recall we said $\sqrt{n}(\hat{\theta}_{MLE} - \theta^*) \to N(0, \sigma^2_{MLE})$. Another way of putting this is that $\frac{(\hat{\theta}_{MLE} - \theta^*)}{se} \to N(0,1)$ where se is the standard error.

$se = 1/\sqrt{nI(\theta)}$ where $I(\theta) = -\mathbb{E}[\frac{\partial^2 \ell(\theta)}{\partial \theta^2}]$ (called the Fisher Information). Thus we can calculate the se pretty straightfowardly.

Let $\hat{se} = 1/\sqrt{nI(\hat{\theta}_{MLE})}$. It also holds that $\frac{(\hat{\theta}_{MLE} - \theta^*)}{\hat{se}} \to N(0,1)$.

So we can choose our interval to be $\hat{C} \equiv (\hat{\theta}_{MLE} - z_{\alpha/2}\hat{se}, \hat{\theta}_{MLE} + z_{\alpha/2}\hat{se})$ where $z_{\alpha/2}$ is the $\alpha/2$ quantile of $N(0,1)$, $z_{0.5/2} \approx 1.96$.

$P(\theta^* \in \hat{C}) \to 1 - \alpha$ as $n \to \infty$. Often we choose $\alpha = 0.05$ to get a 95% confidence interval. This is the probability that $\hat{C}$ contains the true value.

Bayesian estimation

One alternative to MLE is computation of a Bayesian posterior distribution:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

That is, in place of a single "guess" $\hat{\theta}_{MLE}$, we want a probability distribution over possible values of θ, representing our beliefs about θ given our prior belief $p(\theta)$ updated by the data.

Once you have the posterior, you may be interested in using the mean/median/mode of $p(\theta|D)$ as a "guess" of θ.

Bayesian estimation for coin flips

Consider our coin flip example again, for which we derived the MLE estimator.

Now we specify a prior distribution for θ. One choice is $\theta \sim Beta(\theta; \alpha_H, \alpha_T)$, i.e.,

$$p(\theta) = \frac{\theta^{\alpha_H - 1}(1-\theta)^{\alpha_T - 1}}{B(\alpha_H, \alpha_T)}$$

where $B(\cdot)$ is the beta function and α_H, α_T are parameters governing our prior, called hyperparameters (you have to specify values for these). Then:

$$p(\theta | N_H, N_T) = \frac{\theta^{N_H + \alpha_H - 1}(1-\theta)^{N_T + \alpha_T - 1}}{B(N_H + \alpha_H, N_T + \alpha_T)}$$

where N_H, N_T denote the number of heads and tails, respectively.

Bayesian estimation for a discrete Bayesian network

We have to make some assumptions/decisions about our prior $p(\theta)$. It is common to assume *parameter independence*:

$$p(\theta) = \prod_{i=1}^{n} p(\theta_{x_i|x_{pa(i)}})$$

then we have:

$$p(\theta|D) = \frac{1}{p(D)} \prod_{i=1}^{n} \theta_{x_i|x_{pa(i)}} p(\theta_{x_i|x_{pa(i)}})$$

the posterior is a product of local terms.

Bayesian estimation for a discrete Bayesian network

How do you calculate this thing?

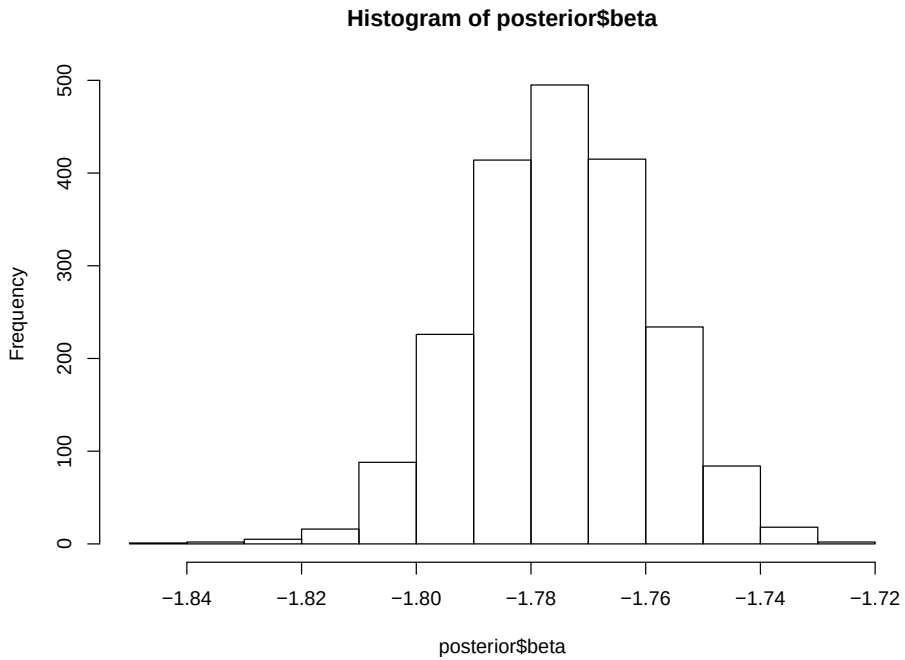
$$p(\theta|D) = \frac{1}{p(D)} \prod_{i=1}^{n} \theta_{x_i|x_{pa(i)}} p(\theta_{x_i|x_{pa(i)}})$$

As we discussed a few weeks ago, there are certain special prior distributions which are convenient because the posterior can be computed in closed form. Specifically, the Dirichlet prior distribution for θ is the conjugate prior for multinomial discrete data, and so the posterior can be calculated exactly: it is also Dirichlet. (There are other examples like this for certain parametric families.)

What if you have a more complicated parametric family, or your beliefs don't correspond to the conjugate prior for that parametric family? Then computing the posterior exactly can be quite hard, or impossible. What could you do instead? Sample from the posterior with Monte Carlo methods we will discuss later.

Sampling from a posterior

Instead of calculating the posterior explicitly for certain parametric families, it has become more common in recent years to use approximate inference techniques to sample from the posterior. This involves setting up a Markov chain which has as its stationary distribution the (hard to explicitly calculate) posterior distribution of interest. $\Rightarrow$ this is exactly what you'll do with Stan in your homework assignment!



Credal intervals for Bayesian estimators

There is a concept analagous to (but importantly different from!) confidence intervals in Bayesian statistics. If I have a posterior distribution $p(\theta|D)$ I can find values (a, b) such that:

$$P(\theta \in (a, b)|D) = \int_a^b p(\theta|D)d\theta = 1 - \alpha$$

this is called a credal interval or posterior interval. Notice that because we're treating θ as a random variable (something which has probability distribution, reflecting our beliefs), this is indeed a probability statement about θ.

The frequentist confidence interval was a probability statement about a *random* interval, and a *fixed* parameter value θ^*. $P(\theta^* \in \hat{C}) \rightarrow 1 - \alpha$ is a different statement which may or may not be satisfied by a given credal interval.

MAP estimation

You may also consider giving up on the Bayesian desire for a posterior distribution over θ, and instead be satisfied with an estimate/"guess" for θ based on the posterior. MAP = "maximum a posteriori"

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \log p(\theta|D)$$

MAP estimation

You may also consider giving up on the Bayesian desire for a posterior distribution over θ, and instead be satisfied with an estimate/"guess" for θ based on the posterior. MAP = "maximum a posteriori"

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \log p(\theta|D)$$

$$\begin{aligned}\arg\max_{\theta} \log p(\theta|D) &= \arg\max_{\theta} \log \frac{p(D|\theta)p(\theta)}{p(D)} \\ &= \arg\max_{\theta} \left(\log p(D|\theta) + \log p(\theta)\right)\end{aligned}$$

which is like maximizing the log-likelihood + an extra term which is your (log) prior, a regularization term. A priori likely values of θ get a "boost" and a priori unlikely values suffer a cost.

MAP estimation

MAP inference is thus no harder than MLE (assuming we can calculate the prior we specified without any trouble). From the perspective of minimizing some loss function, this amounts to replacing the risk

$$\mathbb{E}[L(x, \theta)]$$

with

$$\mathbb{E}_{\pi}[L(x, \theta)]$$

where the expectation is taken with respect to the prior distribution $\pi = p(\theta)$. This is called the Bayes risk (or negative expected utility).

Overview

MLE:

- Has some nice statistical properties. Gives us a good "guess" $\hat{\theta}$ for parameter θ and allows us to construct an interval which contains the true value with high probability.
- In discrete DAGs, relatively straightfoward but there is the problem of data fragmentation.
- In Gaussian DAGs, can just use regression.
- In MRFs, it is generally difficult because of the partition function. There are some ways of getting around this.
- In Gaussian MRFs, it is a (convex) constrained optimization problem.

Bayes:

- Produces a distribution over θ values, depending on the specified prior. May take the mean/median/mode of posterior, or construct a credal interval.
- In common parametric families, conjugate priors give posteriors you can calculate analytically.
- If you can't calculate the posterior, may use approximate inference methods to obtain samples from the posterior.
- May also do MAP estimation, which is like MLE regularlized by the prior.