Understanding Priors by Sampling from the Grid

Imad Ali

2/2/2017

Contents

Introduction

The purpose of this note is two-fold. First, we want to emperically explore how prior distributions influence posterior distributions. Second, we want to explore the trade-off between model complexity and big data. Most techiques taught in Statistics deal with naive models and small data, while the machine learning community tends to deal with simple models and large data. Within machine learning, the deep learning community has been able to find solutions involving big data albiet at the expense of sacraficing model transparency. Bayesian approaches are particularly useful at capturing uncertainty with sparse data and complex models. With implementations like Stan, Bayesian methods are heading in a direction which will allow efficient computation on big data without without losing the transparency of declaring your model.

One Observation and One Parameter

Belief about the data

Assume you have a single observation y that you *believe* has been generated from the binomial distribution according to some unknown probability parameter θ and some known sample size n. For example, you have data on n (Bernoulli) experiments and information on how many of the n experiements resulted in a success, denoted y. In addition to the prior distribution being bound on the closed unit interval [0,1], you might also have some prior knowledge as to the distribution that θ resides in. This domain specific knowledge might encourage you to believe that θ is close to some set of values on the closed unit interval.

We can encode our belief about the model and prior information using Bayes' theorem:

$$p(\theta|y) = \frac{f(y|\theta)g(\theta)}{\int_\Omega f(y|\theta)g(\theta)\ d\theta}$$

Given that our single observation comes from the binomial distribution, the likelihood of our data $f(y|\theta)$ is simply the binomial probability mass function for y given θ. Formally,

$$f(y|\theta) = \binom{n}{y}\theta^y(1-\theta)^{n-y}$$

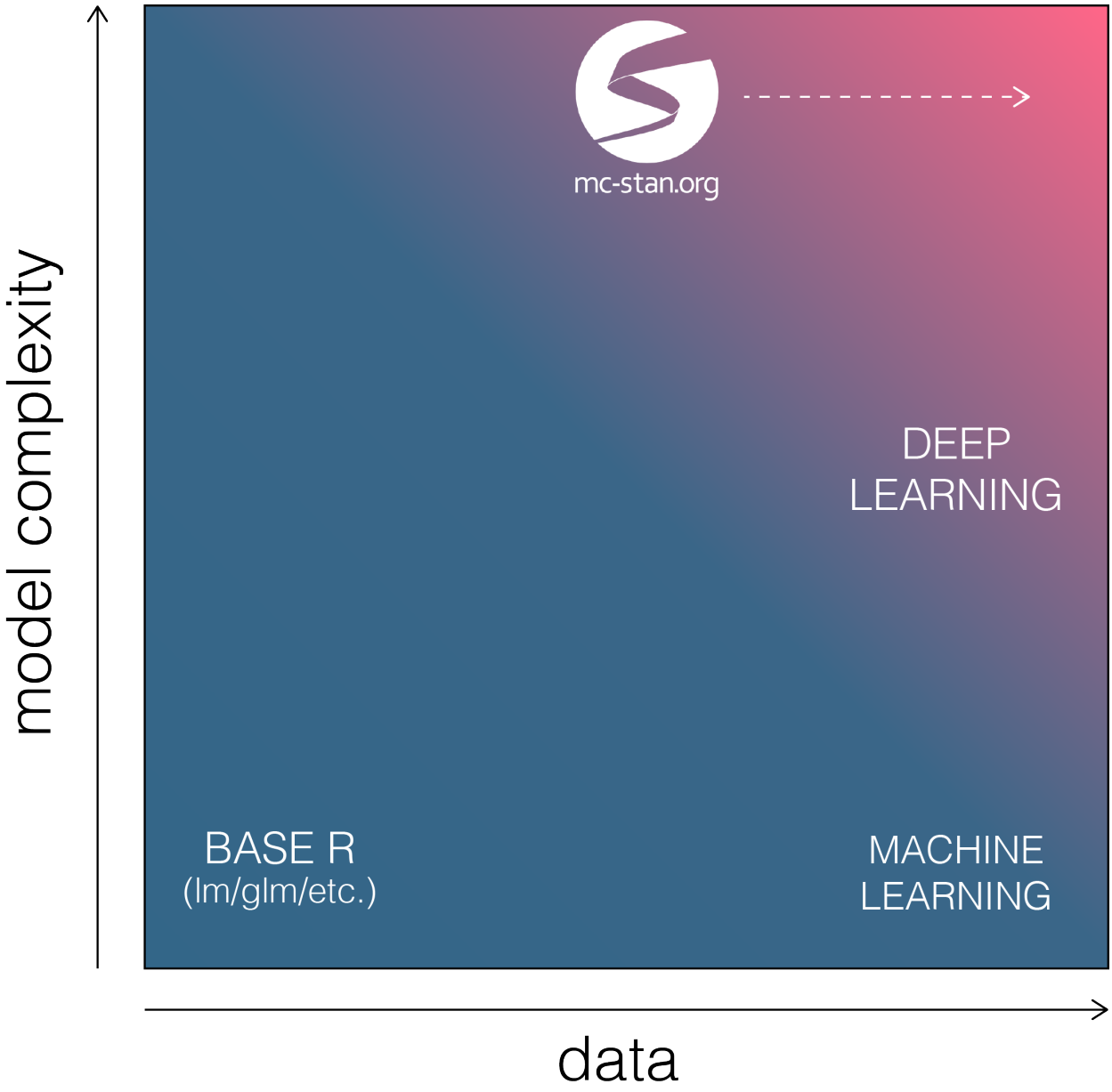

model complexity
mc-stan.org
DEEP
LEARNING
BASE R
(lm/glm/etc.)
MACHINE
LEARNING
data

Figure 1:

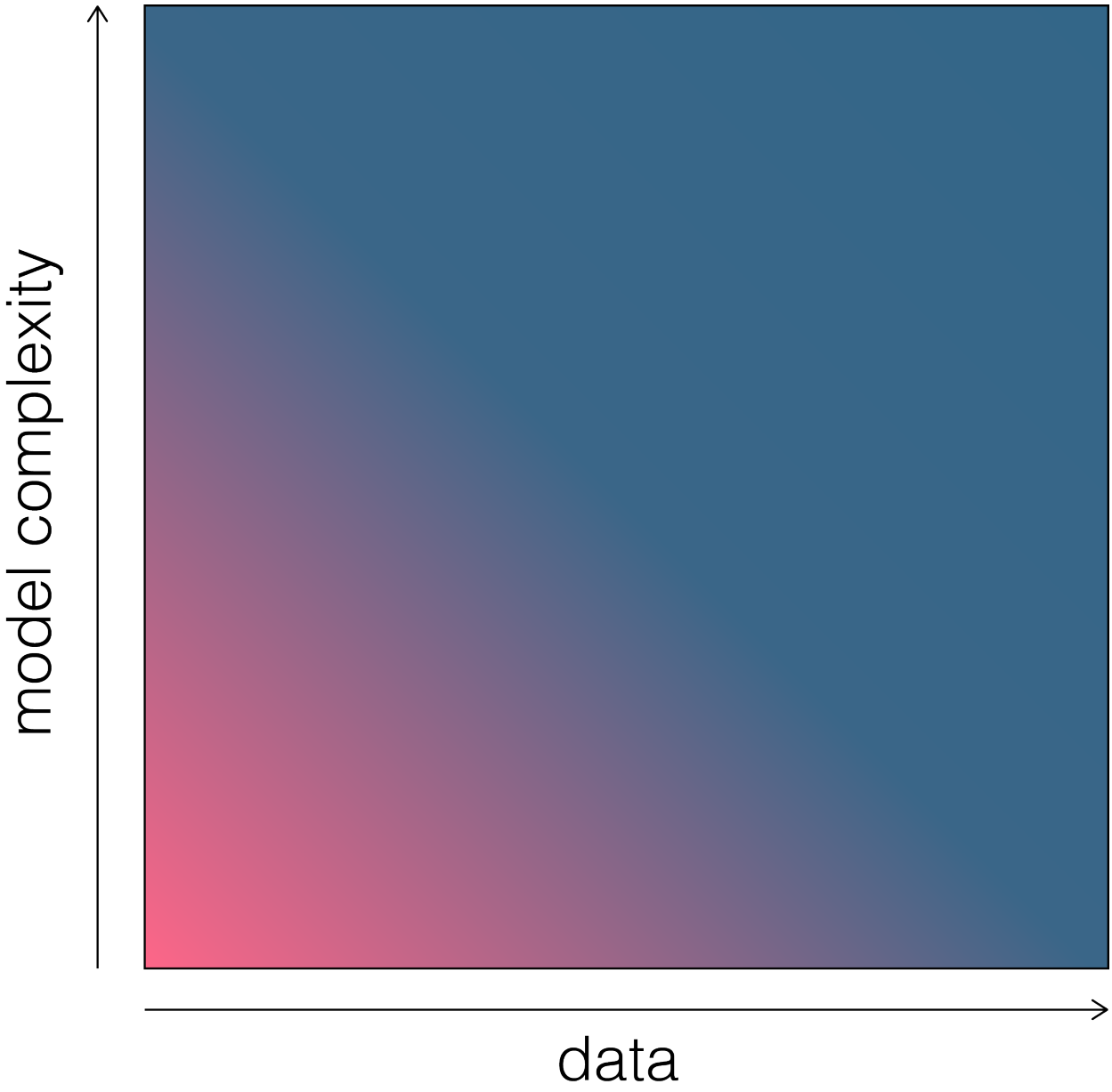


Figure 2:

Belief about the parameter

Our domain specific knowledge might lead us to consider various distributional specifications for our prior distribution $g(\theta)$. Here we will consider the beta, uniform, normal, and Cauchy priors distributions on θ:

$$
\begin{aligned}
g_b(\theta|\alpha,\beta) &= \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} \\
g_u(\theta) &= 1 \\
g_n(\theta|\mu,\sigma) &= \left[\sigma\sqrt{2\pi}\right]^{-1} e^{-\frac{(\theta-\mu)^2}{2\sigma^2}} \\
g_c(\theta|x_0,\gamma) &= \left\{\pi\gamma\left[1+\left(\frac{\theta-x_0}{\gamma}\right)^2\right]\right\}^{-1}
\end{aligned}
$$

Our posterior distribution of θ will adjust depending on the functional form used on the parameter θ.

Sampling from the grid

Sampling from the grid (or grid approximation) can be thought of as a "brute force" way to estimate your posterior distribution. Its name derives from the procedure using a grid of candidate parameter values (e.g. a vector, matrix, or tensor) to determine your posterior probabilities associated with each candidate parameter or candidate tuple of parameters. We can then use these posterior probabilities to sample from the grid of parameters with replacement. This will give us a distribution associated with the parameter(s).

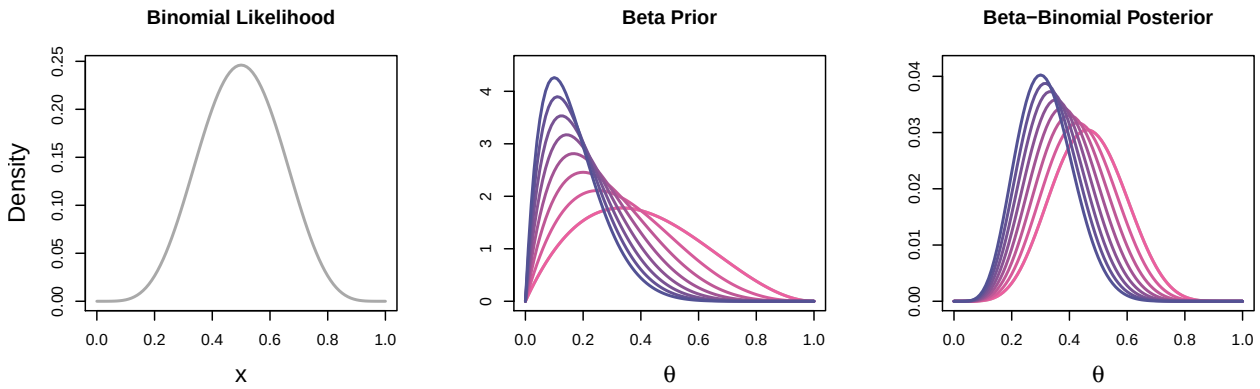
In order to sample from the grid we need to specify a function that can compute the posterior probability associated each candidate value (or at least the probability up to a normalizing constant). We also need a function to sample (with replacement) from the grid of candidate parameter values according to the posterior probability associated with each value. The `sample()` function in R can be uesd to accomplish this. However, in this section we do not sample, but rather plot the posterior distribution associated with each candidate parameter value. In other words, we are looking at the distribution that the samples converge to as the number of samples converges to infinity.

Below is the R code to calculate the posterior probabilities of each value of θ using beta, uniform, normal, and cauchy prior distributions, respectively. The first three lines setup the grid and the data.

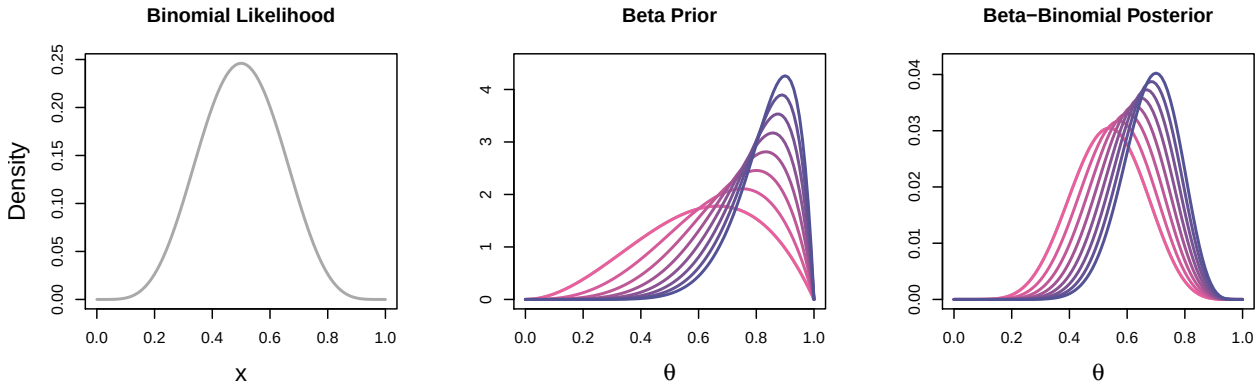
```
prob <- seq(0, 1, by=0.01)  # grid of candidate parameter values (theta)
x <- 5                      # number of successes
n <- 10                     # number of trials

# beta prior
binom_beta <- function(x, n, theta, alpha, beta) {
  lik <- dbinom(x, n, theta)
  prior <- dbeta(theta, alpha, beta)
  post <- (lik * prior) / sum(lik * prior)
  return(list('lik' = lik, 'prior' = prior, 'post' = post))
}
# uniform prior
binom_unif <- function(x, n, theta, alpha, beta) {
  lik <- dbinom(x, n, theta)
  prior <- dunif(theta, alpha, beta)
  post <- (lik * prior) / sum(lik * prior)
  return(list('lik' = lik, 'prior' = prior, 'post' = post))
}
# normal prior
```

```
binom_norm <- function(x, n, theta, loc, scale) {
  lik <- dbinom(x, n, theta)
  prior <- dnorm(theta, loc, scale)
  post <- (lik * prior) / sum(lik * prior)
  return(list('lik' = lik, 'prior' = prior, 'post' = post))
}
# cauchy prior
binom_cauchy <- function(x, n, theta, loc, scale) {
  lik <- dbinom(x, n, theta)
  prior <- dcauchy(theta, loc, scale)
  post <- (lik * prior) / sum(lik * prior)
  return(list('lik' = lik, 'prior' = prior, 'post' = post))
}
```

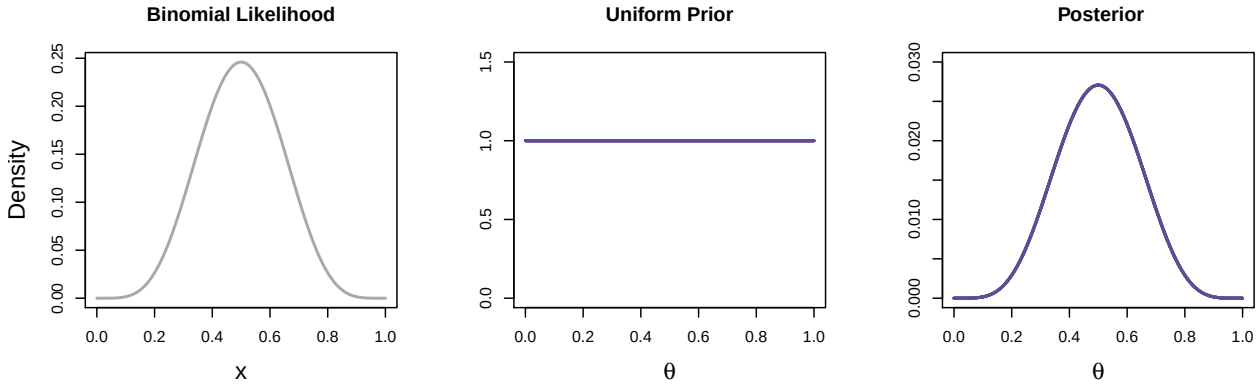
The code below runs through the `binom_beta()` function which uses the $Beta(\alpha, \beta_i)$ prior distribution with $\alpha = 2$ and increasing values for $\beta_i \in [2, 11]$. We then plot the distribution of the likelihood of the data, the distribution of the prior on θ, and the posterior distribution of θ. The figure below illustrates the relationship between the prior and the posterior as the Beta prior belief on probability shifts towards zero.



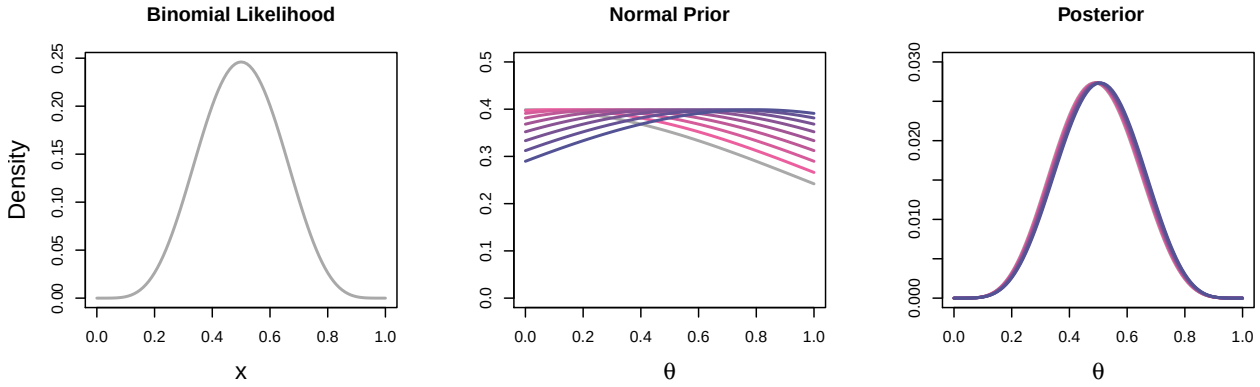
The figure below shows the relationship of the prior and the posterior as the Beta prior belief on probability shifts towards one. In this case the prior is using $\alpha_i \in [2, 11]$ and $\beta = 2$.



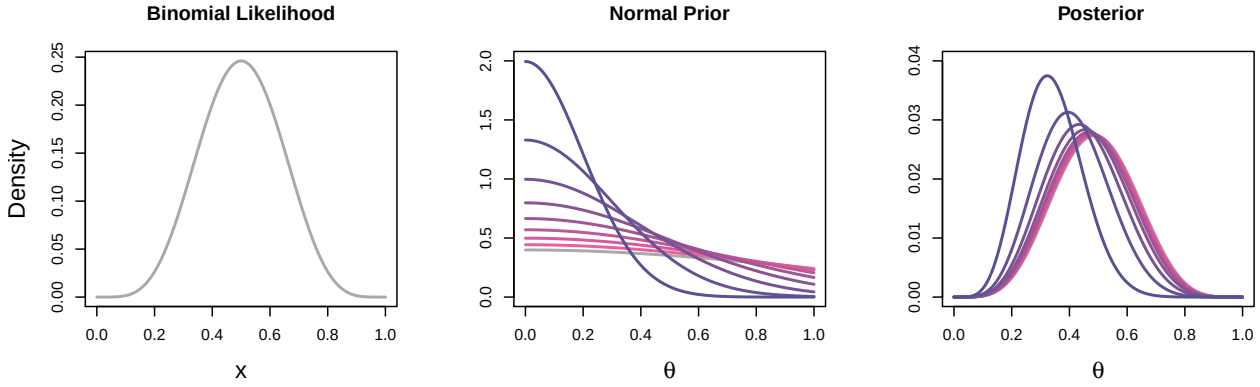
Using a $Unif(0, 1)$ prior does not change the shape of the posterior (i.e. no prior information is being encoded in the model since the uniform probability density function is a constant).



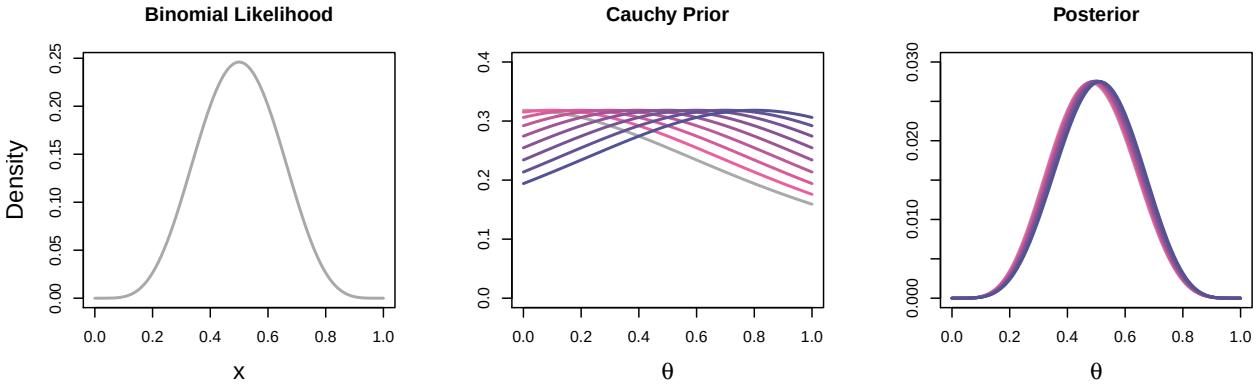
The figure below shows the relationship between the prior and the posterior when using a truncted $\mathcal{N}(\mu_i, \sigma)$ prior with $\mu_i \in [0, 0.8]$ and $\sigma = 1$.



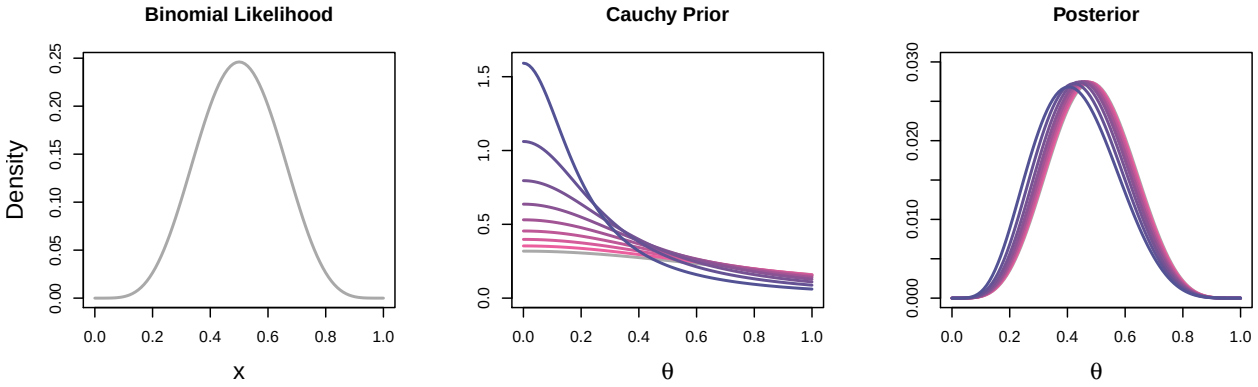
The figure below shows the relationship between the prior and the posterior when using a truncted $\mathcal{N}(\mu, \sigma_i)$ prior with $\mu = 0$ and $\sigma_i = [0.2, 1]$.



The figure below shows the relationship between the prior and the posterior when using a truncted $Cauchy(x_{0_i}, \gamma)$ prior with $x_{0_i} \in [0, 0.8]$ and $\gamma = 1$.



The figure below shows the relationship between the prior and the posterior when using a truncted $Cauchy(x_0, \gamma_i)$ prior with $x_0 = 0$ and $\gamma \in [0.2, 1]$.



One Observation and Multiple Parameters

```
y <- 0 # rnorm(1, 0, 1)
mu_grid <- seq(-10, 10, by=0.1)
sd_grid <- seq(0.1, 10, by=0.1)

post_norm <- function(y, mu_grid, sd_grid) {
  lik_fun <- function(mu, sd) {               # likelihood function
    dnorm(y, mu, sd)
  }
  prior_mu <- dnorm(mu_grid, 0 ,1)            # prior on mu
  prior_sd <- dnorm(sd_grid, 0, 1)            # prior on sd
  prior <- outer(prior_mu, prior_sd)          # outer prod for priors
  lik <- outer(mu_grid, sd_grid, "lik_fun")   # outer prod proc through lik_fun()
  post <- (lik * prior) / (sum(lik * prior))  # posterior probability grid
  return(post)
}

# evaluate full posterior grid
post_full <- post_norm(y, mu_grid, sd_grid)
# samples of mu
post_mu <- sample(mu_grid, size = 10000, replace = TRUE, prob = rowSums(post_full))
# samples of sd
post_sd <- sample(sd_grid, size = 10000, replace = TRUE, prob = colSums(post_full))
```

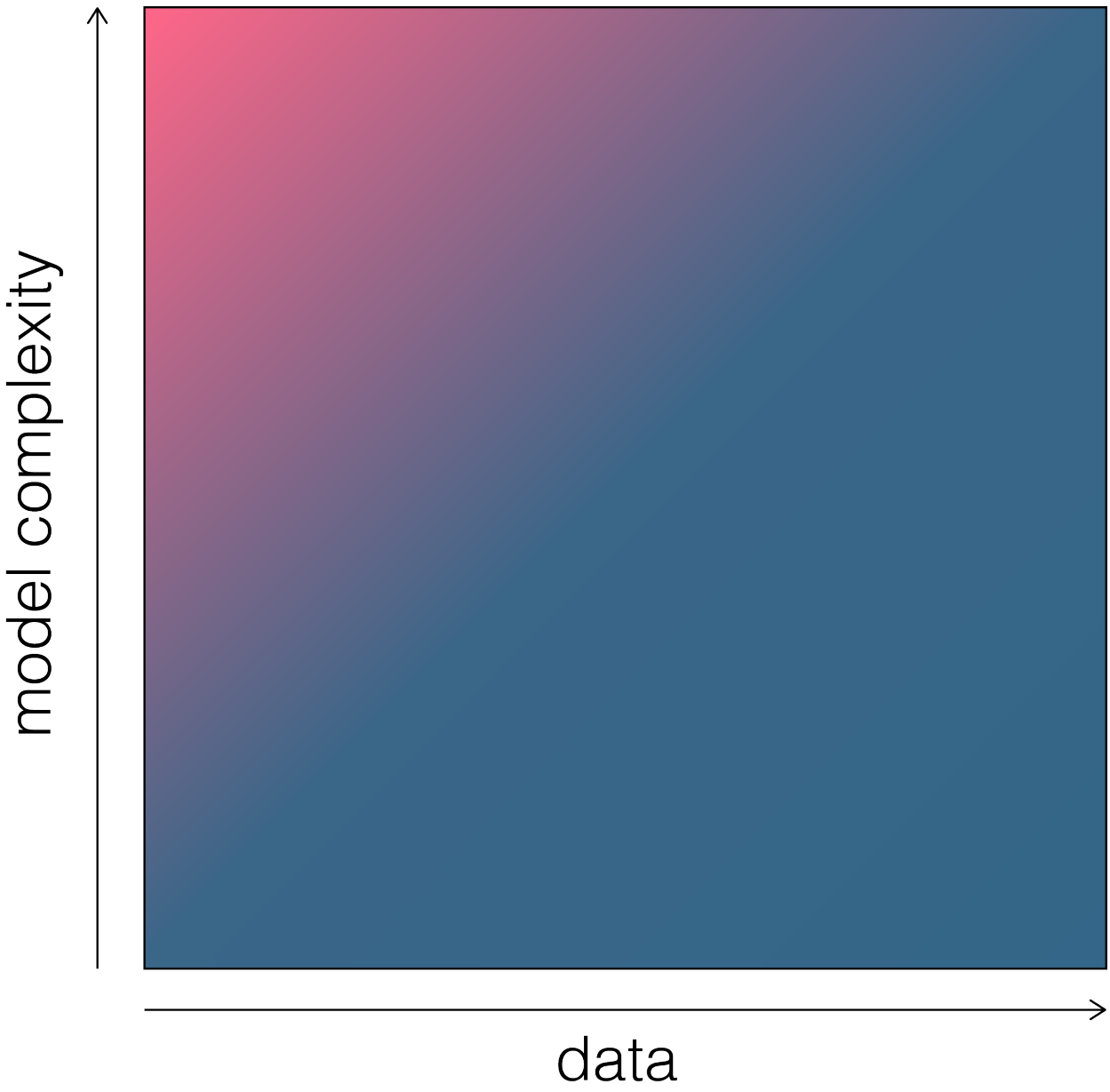
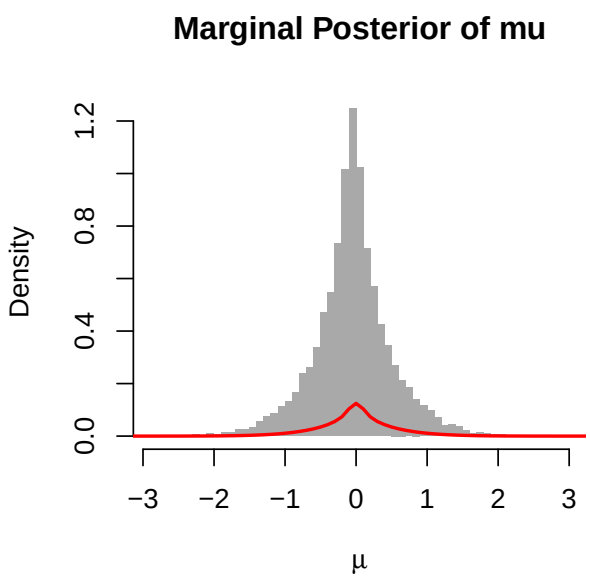
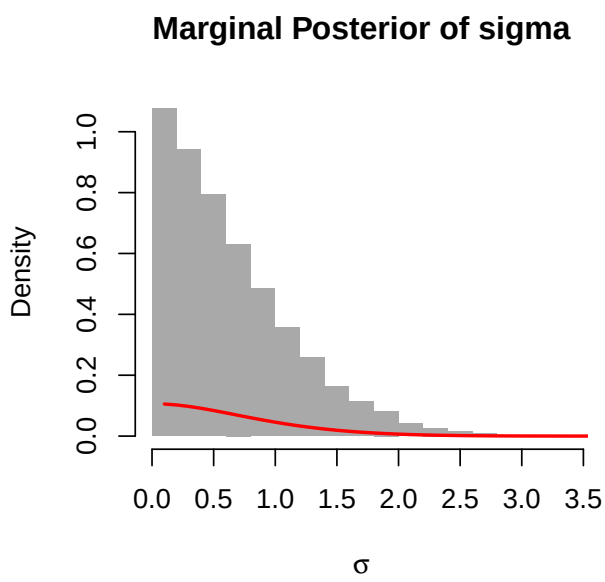
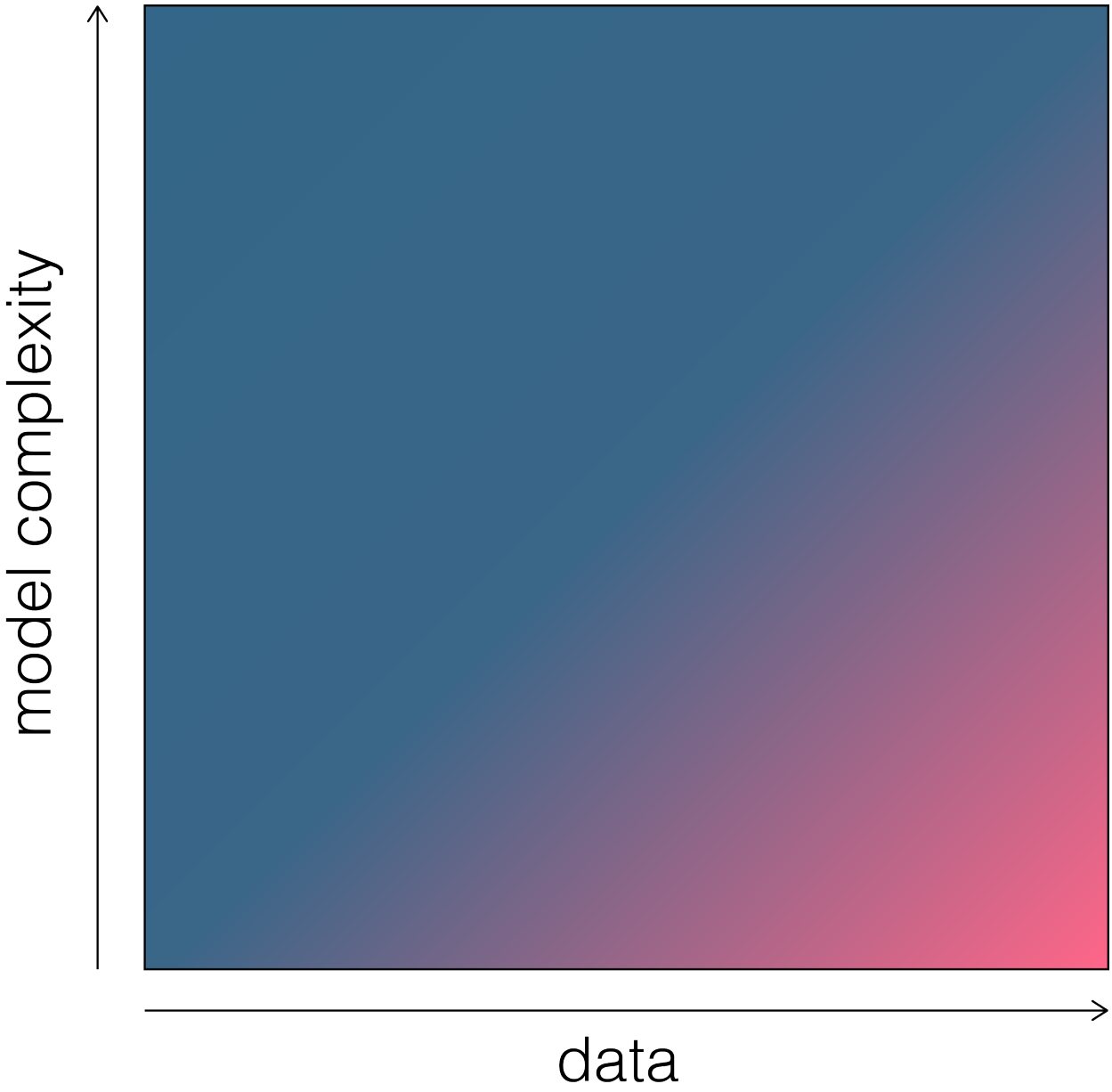


Figure 3:


Marginal Posterior of mu
Density
0.0
0.4
0.8
1.2
−3
−2
−1
0
1
2
3
μ


Marginal Posterior of sigma
Density
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
σ

Multiple Observations One Parameter



Multiple Observations Multiple Parameters