Motivating Example

- Suppose we are trying to fit a production frontier, but have little information about its functional form:

$$Y_i = F(X_i)\Delta_i\tilde{\varepsilon}_i$$

  - Y_i is ith observation of output
  - X_i is ith observation of inputs
  - F is (unknown) production function
  - $\Delta_i \in [0,1]$ is technical efficiency
  - $\tilde{\varepsilon}_i > 0$ is observation error

Motivating Example: Data

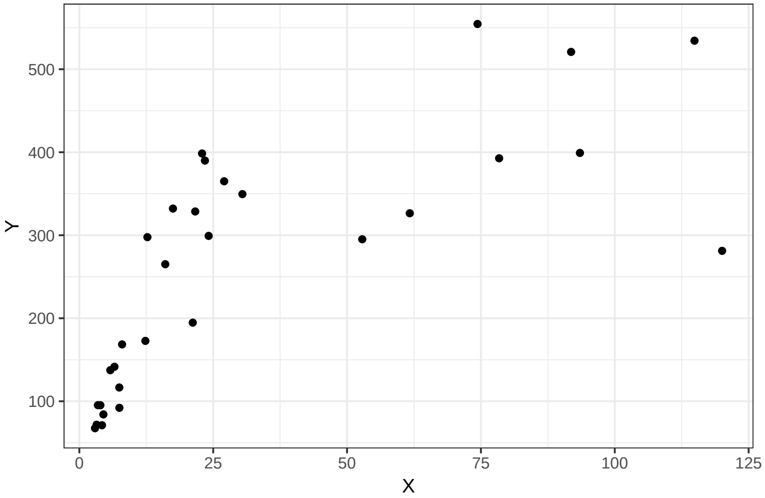

Figure: Simulated Data

Motivating Example: Cleaning Data

- Common first step: Log-transform:

$$y_i = f(X_i) + \delta_i + \varepsilon_i$$

  - $y_i = \log Y_i$
  - $f(X_i) = \log F(X_i)$
  - $\delta_i = \log \Delta_i \leq 0$
  - $\varepsilon_i = \log \tilde{\varepsilon}_i$
- Make distributional assumptions:
  - $\delta_i \sim N^{-}(0, \sigma_\delta)$
  - $\varepsilon_i \sim N(0, \sigma_\varepsilon)$

Motivating Example: Data

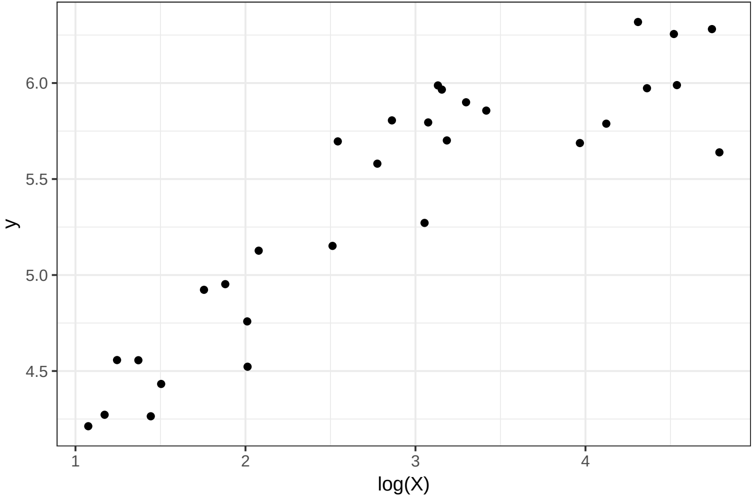

Figure: Log Simulated Data

Motivating Example: Functional Forms

- We still need to select a form for f
- Traditionally, parametric forms of f have been used
  - Log-linear: $f(X_i) = \log(X_i)\beta$
  - Translog: All log-inputs, squared log-inputs, and interactions between log-inputs
- Could also use a non-parametric specification
  - Du et al. (2013) use kernel smoothing to fit conditional mean of the data, use residuals to fit distributional parameters
- How can we select between these functional forms, especially given limited data?
  - Log-linear and translog forms can be compared using many methods
  - Difficult to compare with non-parametric methods

Motivating Example: Functional Forms

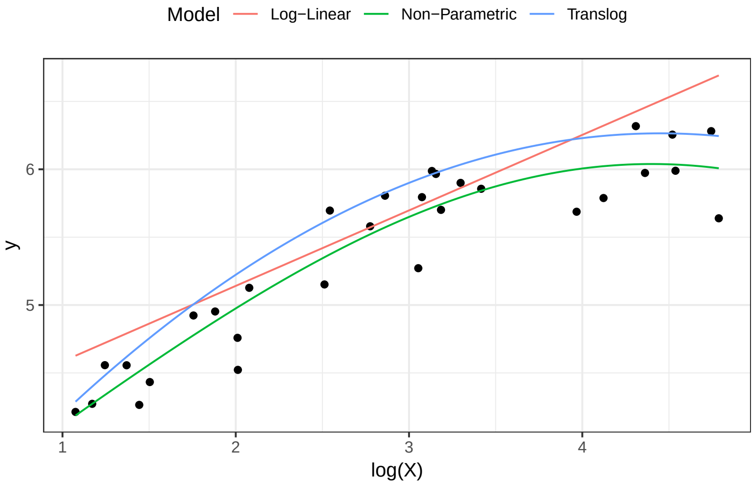

Figure: Fitted Frontiers

Model Selection

- Classical methods
  - Based on likelihood values
  - Some methods are restrictive in what models can be compared (e.g., nested models), others like information criteria are general
  - Require moderate to large sample sizes
  - **Problems:**
    - Kernel smoothing is not likelihood-based
    - Sample sizes may not be large enough
    - Classical methods are unreliable for estimating stochastic frontiers

Model Selection

- Cross-validation
  - Fit the model with one sub-sample, test accuracy against another sub-sample
  - Requires large sample sizes and a metric for accuracy
  - **Problem:** Sample sizes are not large enough

Model Selection

- Bayesian methods
  - Very general model comparison, can estimate the probability a model is correct from a set of exclusive models
    - Accounts for likelihood and numbers of parameters
  - No sample size restrictions in general
  - **Difficulties:**
    - Need a Bayesian analogue of kernel smoothing (Gaussian processes)
    - Calculating model probabilities is hard in general, existing methods are unsuitable for large models applied to small samples

Bayesian Model Selection

- Given a set of exclusive models $\{M_1, ..., M_K\}$, the probability that model M_k is the true model given data y is

$$\Pr(M_k|y) = \frac{m(y|M_k)p(M_k)}{\sum_{j=1}^{K} m(y|M_j)p(M_j)}$$

  - $m(y|M_k)$ is marginal likelihood of model M_k
  - $p(M_k)$ is prior probability that M_k is the true model
- Marginal likelihoods are difficult to compute in general

Marginal Likelihood Estimation

- Many methods have been developed to compute marginal likelihoods
  - If Gibbs or Metropolis-Hastings sampling are used, efficient methods developed in Chib (1995) and Chib and Jeliazkov (2001)
  - Laplace's method and Gaussian quadrature perform well for moderately-sized models
  - Bridge sampling is very efficient and is widely applicable (in theory)

Marginal Likelihood Estimation: Problems

- Existing methods are not well suited to marginal likelihood estimation for this model and data
  - Model is not of correct form for Gibbs sampling
  - Excessive convergence times for Metropolis-Hastings sampling
  - Models have too many parameters relative to sample size for Laplace's method, accuracy is degraded
  - Models have too many parameters for Gaussian quadrature to be computationally feasible
  - Bridge sampling is prone to numerical issues in large models, accuracy is degraded (more on this later)

Numerical Issues With Bridge Sampling

- Bridge sampling estimates marginal likelihood with

$$\hat{m}(y) = \frac{\frac{1}{N_2}\sum_{s=1}^{N_2} p\left(y|\theta_g^{[s]}\right) p\left(\theta_g^{[s]}\right) h\left(\theta_g^{[s]}\right)}{\frac{1}{N_1}\sum_{s=1}^{N_1} h\left(\theta_y^{[s]}\right) g\left(\theta_y^{[s]}\right)}$$

  - θ denote parameters
  - $p(y|\theta)$ is the likelihood
  - $p(\theta)$ are priors over parameters
  - $g(\theta)$ is the proposal distribution, chosen by the researcher
  - $h(\theta) = (r_1 p(y|\theta)p(\theta) + r_2 \hat{m}(y)g(\theta))^{-1}$
  - θ_g are parameter samples taken from g (total of N_2 samples)
  - θ_y are parameter samples taken from the posterior distribution (total of N_1 samples)

Numerical Issues With Bridge Sampling

$$\hat{m}(y) = \frac{\frac{1}{N_2}\sum_{s=1}^{N_2} p\left(y|\theta_g^{[s]}\right) p\left(\theta_g^{[s]}\right) h\left(\theta_g^{[s]}\right)}{\frac{1}{N_1}\sum_{s=1}^{N_1} h\left(\theta_y^{[s]}\right) g\left(\theta_y^{[s]}\right)}$$

- For large numbers of parameters, $p(\theta)$ and $g(\theta)$ can be very small positive numbers
- Thus, $h(\theta)$ can take on very large values
- Terms in each sum can be very large or very small, are eventually truncated because of finite machine precision, making numerator and denominator inaccurate
- Inaccuracies are magnified by division of numerator by denominator
- Will show an example of biased marginal likelihood estimates that result

Iterative Kernel Density Estimation

- The marginal likelihood of model M_k can be written as

$$m(y|M_k) = \frac{f(y|\theta, M_k)p(\theta|M_k)}{p(\theta|y, M_k)}$$

  - $f(y|\theta, M_k)$ is the likelihood, defined by model
  - $p(\theta)$ is the prior density, defined by researcher
  - $p(\theta|y, M_k)$ is posterior density, must be estimated
- As noted by Chib (1995), this relationship holds for all θ, only need to estimate at one point θ^*, such as the posterior mean

Iterative Kernel Density Estimation

- Markov Chain Monte Carlo (MCMC) produces samples of $\theta | y, M_k$, could theoretically use these samples to estimate posterior density
- Some problems:
  - Traditional kernel density estimators produce biased density estimates
    - Traditional estimators overestimate in low density regions and underestimate in high density regions
    - Adaptive kernel density estimation corrects for this issue (Portnoy and Koenker, 1989)
  - Kernel density estimation suffers from the curse of dimensionality
    - Traditional estimators are largely infeasible for more than six dimensions
    - Adaptive kernel density estimation is unreliable for more than a few dimensions

Iterative Kernel Density Estimation

- To address the curse of dimensionality, first denote the parameter vector as $\theta = (\theta_1, \theta_2, ..., \theta_P)'$
- Posterior density can be written as

$$\begin{aligned}
p(\theta|y) &= p(\theta_1, ..., \theta_P|y) && (1)\\
&= p(\theta_1|\theta_2, ..., \theta_P, y) \times p(\theta_2, ..., \theta_P|y) && (2)\\
&= p(\theta_1|\theta_2, ..., \theta_P, y) \times p(\theta_2|\theta_3, ..., \theta_P, y) && (3)\\
&\quad\ \times p(\theta_3, ..., \theta_P|y) && (4)\\
&= ... && (5)\\
&= p(\theta_1|\theta_2, ..., \theta_P, y) \times p(\theta_2|\theta_3, ..., \theta_P, y) && (6)\\
&\quad\ \times ... \times p(\theta_P|y). && (7)
\end{aligned}$$

- The density of a P-dimensional vector is broken into P one-dimensional densities

Iterative Kernel Density Estimation

The iterative kernel density estimator has the following algorithm:

1. Draw samples of $\theta|y$ using an MCMC algorithm.
2. Choose θ^* from a high-density region of $\theta|y$, such as the sample mean or maximum a posteriori.
3. Estimate the log-density of $\theta_P|y$ at θ_P^* using adaptive KDE, denoting that value $\ln \hat{p}(\theta_P^*|y)$.
4. For each i from $P-1, ..., 1$:
    4.1 Re-estimate the model, setting $(\theta_{i+1}, ..., \theta_P) = (\theta_{i+1}^*, ..., \theta_P^*)$, to obtain draws of $(\theta_1, ..., \theta_i)|(\theta_{i+1}^*, ..., \theta_P^*), y$.
    4.2 Estimate the log-density of $\theta_i|\theta_{i+1}^*, ..., \theta_P^*, y$ at θ_i^* using adaptive KDE, denoting that value $\ln \hat{p}(\theta_i^*|\theta_{i+1}^*, ..., \theta_P^*, y)$.
5. Find the sum of each of the estimated partial log-densities to arrive at an estimate for the overall log-posterior density, denoted $\ln \hat{p}(\theta^*|y)$.

Simulations: Multivariate Normal Linear Model

$$y = X\beta + \varepsilon \tag{8a}$$
$$\varepsilon \sim N(0, \sigma^2 I_{100}) \tag{8b}$$
$$\tag{8c}$$
$$X_1 = \mathbf{1}_{100} \tag{8d}$$
$$X_i \sim U(-10, 10); \qquad i = 2, 3 \tag{8e}$$
$$\beta = (-2, 5, 3)' \tag{8f}$$
$$\sigma = 25 \tag{8g}$$

- Marginal likelihood is estimable via Gibbs sampling and with iterative kernel density estimation
  - Estimate marginal likelihood using Gibbs sampling and iterative kernel density estimation 500 times each
  - Test for equality of means

Simulations: Multivariate Normal Linear Model

Model	# Trials	Gibbs/Chib	Iterative KDE	Mean Test p-value
Multivariate Linear	500	-481.353 (0.154) Iter = 5,000	-481.348 (0.078) Iter = 5,000	0.493

Table: Comparison of Gibbs and Iterative KDE

Simulations: Large Multivariate Normal Linear Model

$$
\begin{aligned}
y &= X\beta + \varepsilon && \text{(9a)}\\
\varepsilon &\sim \; N(0, \sigma^2 I_{100}) && \text{(9b)}\\
& && \text{(9c)}\\
X_1 &= \mathbf{1}_{100} && \text{(9d)}\\
X_i &\sim U(-10, 10); \qquad i = 2, ..., 50 && \text{(9e)}\\
\beta_i &\sim U(-10, 10); \qquad i = 1, ..., 50 && \text{(9f)}\\
\sigma &= 25 && \text{(9g)}
\end{aligned}
$$

- Large model has potential to bias bridge sampling estimates
- Marginal likelihood estimated with Gibbs sampling, iterative kernel density estimation, and bridge sampling 100 times each

Chib	Iterative KDE	Bridge	IKDE = Chib p-value	Bridge = Chib p-value
-606.927 (0.195)	-606.88 (0.24)	-607.094 (0.014)	0.125	1.777×10^{-13}

Table: Comparison of Chib, Iterative KDE, and Bridge Sampling

- Bias in bridge sampling estimates can result in up to 5% difference in model probabilities

Other Simulation Results

- Simulations also performed for probit models, showing similar results
- Like bridge sampling, Laplace's method, and similar estimators, iterative kernel density estimation is widely applicable
  - Used to distinguish probit from logit models in simulations
  - Used to test different forms of the production function in stochastic frontier model (our motivating example)

Production Function Selection

- Recall the stochastic frontier model:
  - $y_i = \log Y_i$
  - $f(X_i) = \log F(X_i)$
  - $\delta_i = \log \Delta_i \leq 0$
  - $\varepsilon_i = \log \tilde{\varepsilon}_i$
- Distributional assumptions:
  - $\delta_i \sim N^-(0, \sigma_\delta)$
  - $\varepsilon_i \sim N(0, \sigma_\varepsilon)$
- Functional forms for f:
  - Log-linear
  - Translog
  - Non-parametric

Production Function Selection: Data

- Sample dataset describing cereal production of 29 countries in 2012, from the World Bank
- Output: Metric tons of cereal grains produced
- Inputs: Area of land used for cereal production, fertilizer consumption, freshwater withdrawals, average rainfall, total number of people working in agriculture

Production Function Selection

- Gibbs sampling can be used to estimate log-linear and translog forms
- Non-parametric (Gaussian process) form must be estimated using another method
- Non-parametric form has many parameters, data are relatively limited
  - Iterative kernel density estimation is the only method appropriate for estimating marginal likelihood in this example

Production Function Selection: Results

Model	# Parameters	Log Likelihood	Log Prior	Log Posterior	Log Marginal Likelihood
Log-Linear	8	-3.734	-10.225	15.249	-29.208
Translog	23	-0.699	-23.666	44.62	-68.984
GP	37	2.68	-27.937	21.458	-46.716

Table: Marginal Likelihood Estimates

-