ANLY601 Assignment2

Mengtong Zhang

January 2020

Collaborate with Shaoyu Feng and Yunjia Zeng

1 Convexity

1.1

$f(x)$ is convex.
First we prove $g(x) = \|x\|_p$ is convex. Using triangle inequality of norm, we have

$$\|\lambda v + (1-\lambda)w\|_p \leq \|\lambda v\|_p + \|(1-\lambda)w\|_p = \lambda\|v\|_p + (1-\lambda)\|w\|_p$$

Thus, $g(x)$ is convex for arbitrary $p > 0$.
$f(x) = \Sigma_{i=1}^{\infty}\|x\|_p$ is a linear combination of norm, thus $f(x)$ is also convex.

1.2

$f(d)$ is not convex.
Suppose $k(d) = k(x, x^{'}) = x - x^{'} = d$, $k(d) = k(x, x^{'}) = (x - x^{'})^2 = d^2$,

$$f(d) = d - d^2$$

$v = 0, w = 1, \lambda = 0.5$, $f(\lambda v + (1-\lambda)w) = f(0.5) = 0.25 > f(0) + f(1) = 0$
Thus, $f(d)$ is not convex.

1.3

$f(d)$ is not convex.
Suppose $k(d) = k(x, x^{'}) = x - x^{'} = d$, $k^{'}(d) = k(x, x^{'}) = x^{'} - x = -d$,

$$f(d) = -d^2 - b$$

$b = 0, v = -1, w = 1, \lambda = 0.5$, $f(\lambda v + (1-\lambda)w) = f(0) = 0 > f(-1) + f(1) = -2$
Thus, $f(d)$ is not convex.

1.4

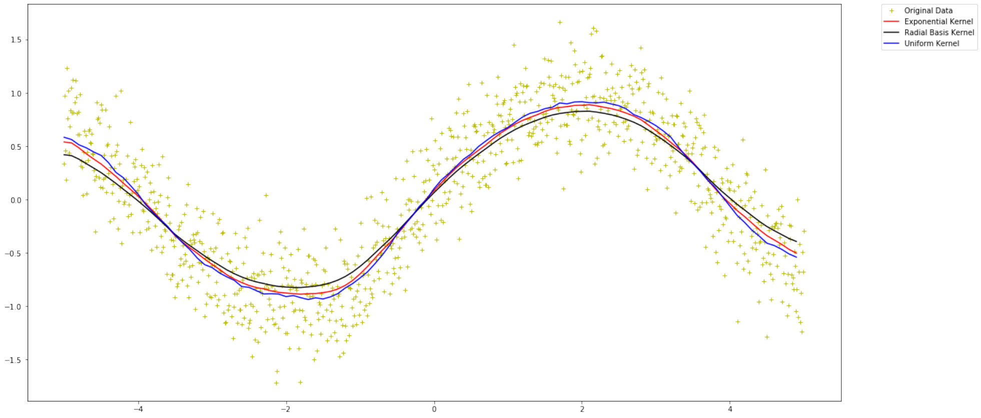
$f(x)$ is convex. $f(x) = \|x\|_p + min(0, -x)$
$\|\lambda v+(1-\lambda)w\|_p+min(0,-\lambda v-(1-\lambda)w) \leq \|\lambda v\|_p+\|(1-\lambda)w\|_p+min(0,-\lambda v)+min(0,-(1-\lambda)w) = \lambda(\|v\|_p + min(0,-v)) + (1-\lambda)(\|w\|_p + min(0,-w))$
From the definition of convexity, $f(x)$ is convex.

1.5

$f(x)$ is convex.
$\|\lambda v+(1-\lambda)w\|_p+max(0,\lambda v+(1-\lambda)w) \leq \|\lambda v\|_p+\|(1-\lambda)w\|_p+max(0,\lambda v)+max(0,(1-\lambda)w) = \lambda(\|v\|_p + max(0,v)) + (1-\lambda)(\|w\|_p + max(0,w))$
From the definition of convexity, $f(x)$ is convex.

2 Kernel Regression

2.1

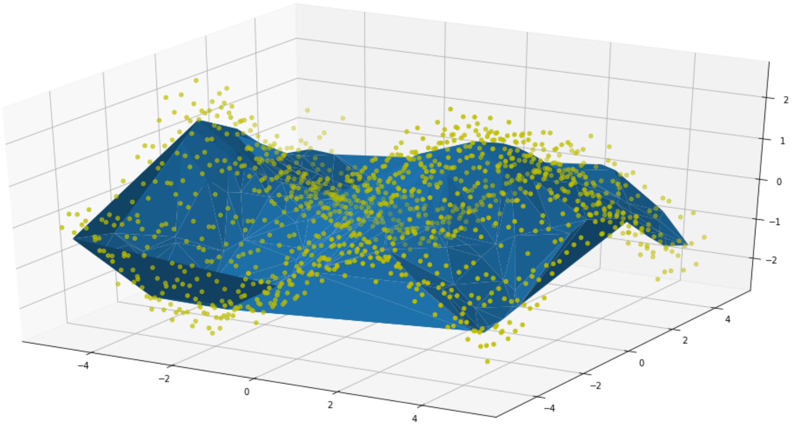


2.2

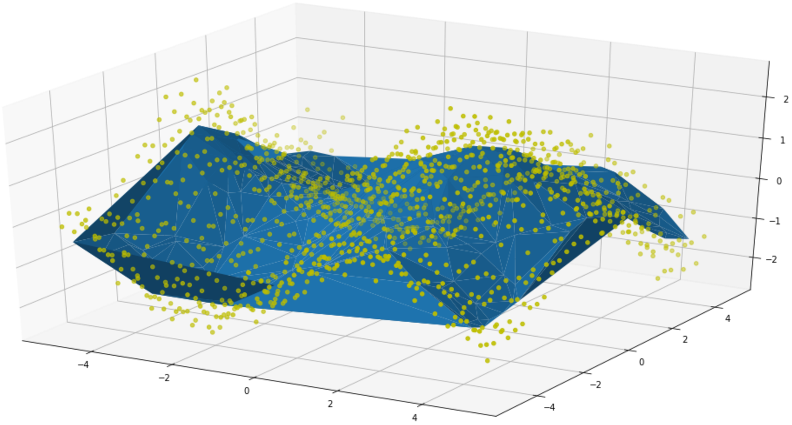
From the picture above, we can see that three kernels can all fit the original data quite well. Exponential Kernel and Radial Basis Kernel generate smooth curves but Uniform Kernel curve is composed with piecewise polylines.

2.3

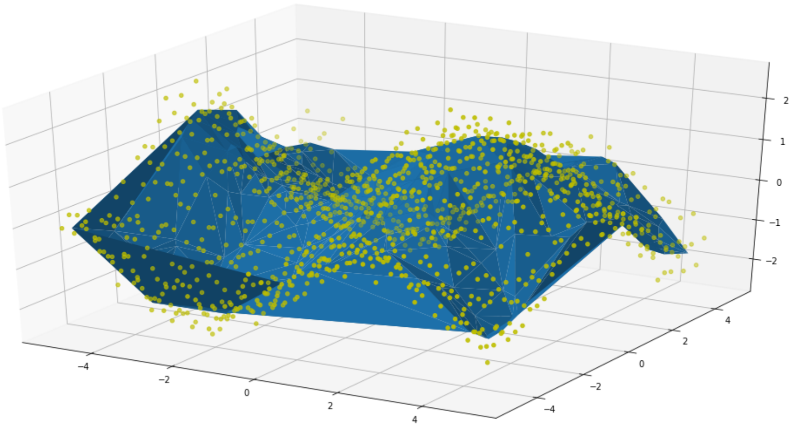
1. Here are 3D fit for three kernels: Exponential Kernel



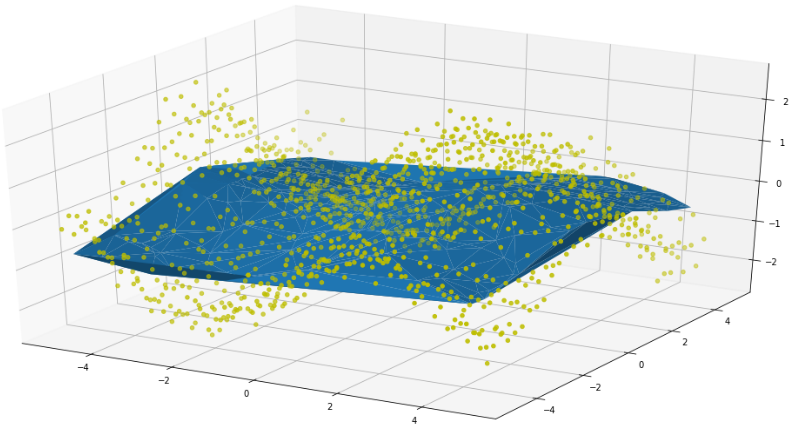
Radial Basis Kernel



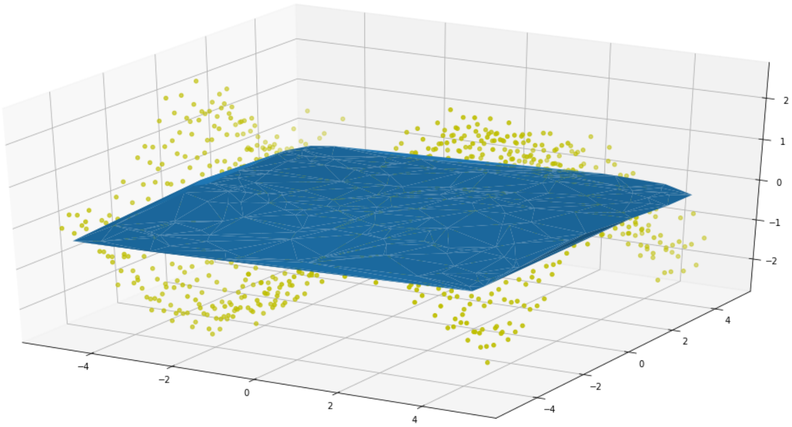
Uniform Kernel



2. If we set bandwidth=3, we will get:



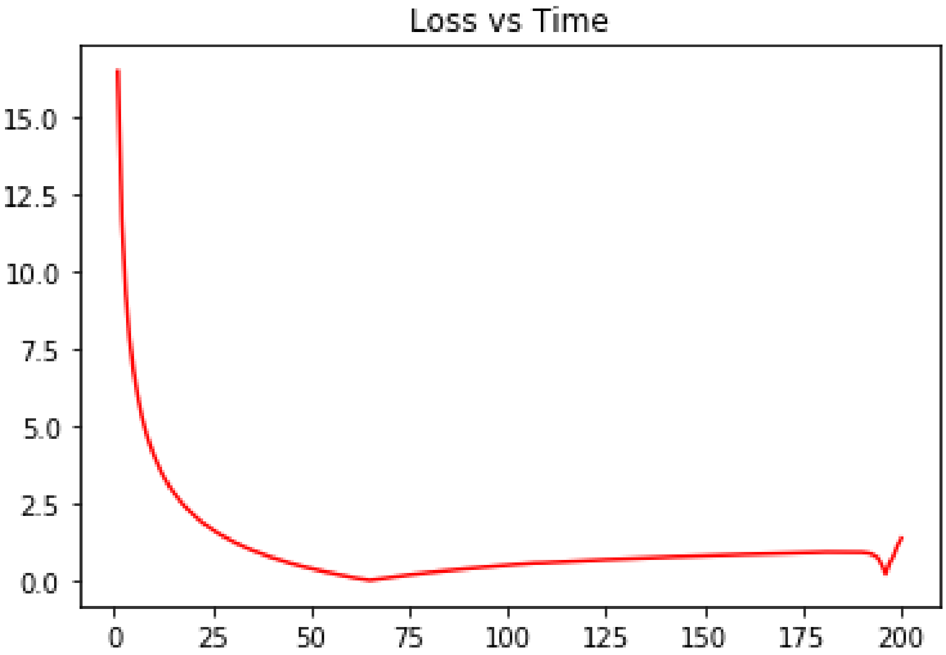
Set bandwidth=8:



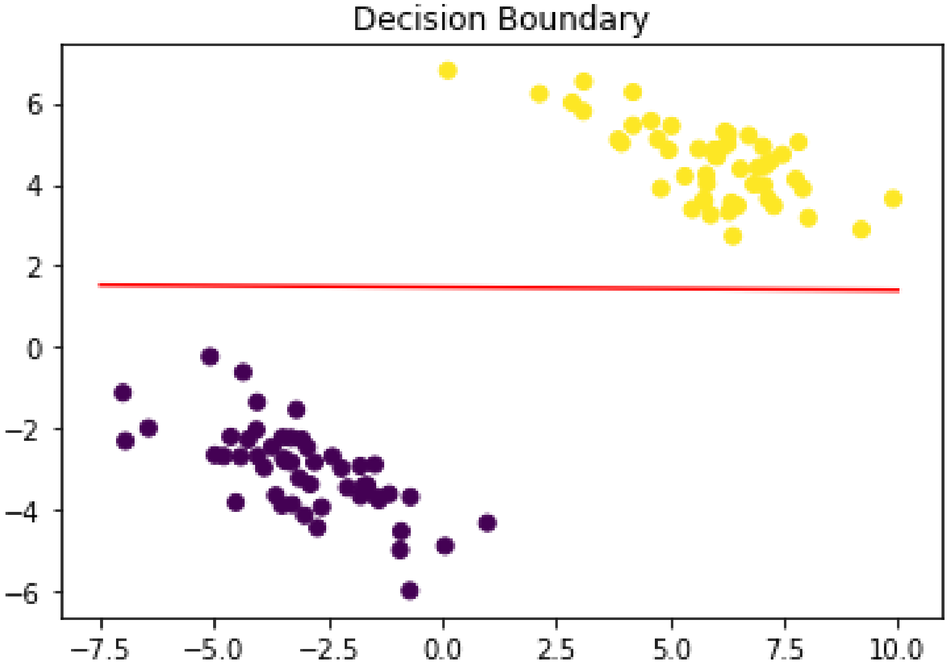
Easy to see that when we increase the width the surface becomes smoother.

3 Programming: Stochastic Subgradient Descent

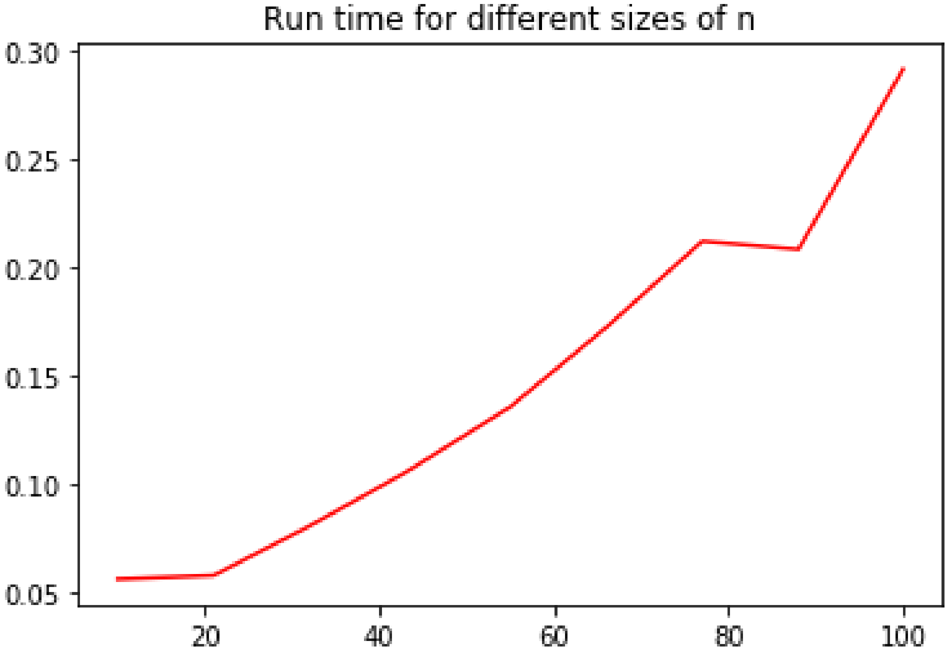
Loss Over Time:



Decision Boundary:



Run Time for Different Sizes of n:



4 Calculating the conjugate distributions

4.1

$P(\mu|\tau,\nu,\sigma,\mathbf{x}) \propto \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi\nu}}\mathbf{e}^{-\frac{(\mu-\tau)^{\mathbf{2}}}{\mathbf{2}\nu}} \prod_{\mathbf{i=1}}^{\mathbf{n}} \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi\sigma}}\mathbf{e}^{-\frac{(\mathbf{x_i}-\mu)^{\mathbf{2}}}{\mathbf{2}\sigma}} \propto \mathbf{e}^{\frac{(\mu-(\tau\sigma+\mathbf{n\bar{x}}\nu)/(\sigma+\mathbf{n}\nu))^{\mathbf{2}}}{\mathbf{2}\nu\sigma/(\sigma+\mathbf{n}\nu)}}$

$$\mu|\tau,\nu,\sigma,\mathbf{x} \sim \mathbf{Normal}(\frac{\tau\sigma+\mathbf{n\bar{x}}\nu}{\sigma+\mathbf{n}\nu}, \frac{\nu\sigma}{\sigma+\mathbf{n}\nu})$$

$P(\sigma^2|\alpha,\beta,\mu,\mathbf{x}) \propto \frac{\beta^\alpha}{\mathbf{\Gamma}(\alpha)}(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}})^{\alpha+\mathbf{1}}\mathbf{e}^{-\beta/\sigma^{\mathbf{2}}} \prod_{\mathbf{i=1}}^{\mathbf{n}} \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi\sigma}}\mathbf{e}^{-\frac{(\mathbf{x_i}-\mu)^{\mathbf{2}}}{\mathbf{2}\sigma^{\mathbf{2}}}} \propto (\frac{\mathbf{1}}{\sigma^{\mathbf{2}}})^{\alpha+\mathbf{1}+\mathbf{n}/\mathbf{2}}\mathbf{e}^{-(\mathbf{2}\beta+\mathbf{\Sigma_{i=1}^n}(\mathbf{x_i}-\mu)^{\mathbf{2}})/\mathbf{2}\sigma^{\mathbf{2}}}$

$$\sigma^2|\alpha,\beta,\mu,\mathbf{x} \sim \mathbf{InverseGamma}(\alpha+\mathbf{n}/\mathbf{2}, \beta+\mathbf{\Sigma_{i=1}^n}(\mathbf{x_i}-\mu)^{\mathbf{2}}/\mathbf{2})$$

4.2

$P(p_1,p_2,...,p_k|x_{ij}, i=1,2,...k, j=1,2,...,k) \propto \prod_{i=1}^n p_i^{\alpha_i-1} \prod_{i=1}^n \frac{n!}{x_{i1}!...x_{ik}!}p_1^{x_{i1}}...p_k^{x_{ik}} \propto \prod_{i=1}^n p_i^{\alpha_i+\Sigma_{j=1}^n x_{ji}-1}$

$p_1,p_2,...p_k|x_{ij}(i=1,2,...k, j=1,2,...,k) \sim Dirichlet(\alpha_1+\Sigma_{j=1}^n x_{j1}-1,...,\alpha_k+\Sigma_{j=1}^n x_{jk}-1)$

4.3

$P(\lambda|x_1,...,x_n) \propto \lambda^{\alpha-1}e^{-\lambda/\beta}\prod_{i=1}^n \lambda^{x_i}e^{-\lambda} \propto \lambda^{n\bar{x}+\alpha-1}e^{\lambda(-n-1/\beta)}$

$$\lambda|x_1,...,x_n \sim Gamma(n\bar{x}+\alpha, \frac{\beta}{\beta n+1})$$

5 Priors as regularizers

5.1

Suppose we are estimating $\beta = (\beta_1,...,\beta_p)$ with prior distribution of β_j as $N(0,\tau^2)$,
$\hat{\beta}_{MAP} = \arg\max_\beta P(\beta|y) = \arg\max_\beta \frac{P(y|\beta)P(\beta)}{P(y)} = \arg\max_\beta P(y|\beta)P(\beta) = \arg\max_\beta \log(P(y|\beta)P(\beta)) = \arg\max_\beta \log P(y|\beta) + \log P(\beta)$
$\arg\max_\beta \left[\log\prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2}{2\sigma^2}} + \log\prod_{j=0}^p \frac{1}{\tau\sqrt{2\pi}}e^{-\frac{\beta_j^2}{2\tau^2}}\right] = \arg\max_\beta \left[-\sum_{i=1}^n \frac{(y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2}{2\sigma^2} - \sum_{j=0}^p \frac{\beta_j^2}{2\tau^2}\right] = \arg\min_\beta \frac{1}{2\sigma^2}\left[\sum_{i=1}^n (y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2 + \frac{\sigma^2}{\tau^2}\sum_{j=0}^p \beta_j^2\right] = \arg\min_\beta \left[\sum_{i=1}^n (y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2 + \lambda\sum_{j=0}^p \beta_j^2\right]$
From above, we can see the target function of maximum posterior estimation is equivalent to ridge regression. Thus, The L2 penalty (ridge) is equivalent to a Normal prior.

5.2

$\arg\max_\beta \Big[\log \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2}{2\sigma^2}} + \log \prod_{j=0}^p \frac{1}{2b} e^{-\frac{|\beta_j|}{2b}} \Big]$
$= \arg\max_\beta \Big[-\sum_{i=1}^n \frac{(y_i-(\beta_0+\beta_1 x_{i,1}+...+\beta_p x_{i,p}))^2}{2\sigma^2} - \sum_{j=0}^p \frac{|\beta_j|}{2b} \Big]$
$= \arg\min_\beta \frac{1}{2\sigma^2} \big[\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}))^2 + \frac{\sigma^2}{b} \sum_{j=0}^p |\beta_j| \big]$
$= \arg\min_\beta \big[\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p}))^2 + \lambda \sum_{j=0}^p |\beta_j| \big]$ From above, we can see the target function of maximum posterior estimation is equivalent to LASSO regression. Thus, The L1 penalty (LASSO) is equivalent to a Laplace prior.

6 General Questions?

6.1

Posterior distribution is a distribution of parameter based on the observed data. But posterior predictive distribution is the predictive distribution of data given the posterior distribution of parameter.

6.2

Posterior predictive distribution. Because it is the distribution of data but posterior distribution is for the parameter.

6.3

First we calculate the MLE of μ and σ^2.

$$f(x_1, x_2, \dots, x_n|\sigma, \mu) = \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = (\frac{1}{\sigma\sqrt{2\pi}})^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2}$$

$\log(f(x_1, x_2, \dots, x_n|\sigma, \mu)) = \log((\frac{1}{\sigma\sqrt{2\pi}})^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2})$
$= n \log \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$
$= -\frac{n}{2} \log(2\pi) - n \log\sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$

$$\frac{d\mathcal{L}}{d\mu} = -\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \mid_\mu = 0$$

$$\frac{1}{2\sigma^2} \sum_{i=1}^n (2\hat{\mu} - 2x_i) = 0$$

$$\hat{\mu}_{MLE} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\frac{d\mathcal{L}}{d\sigma} = -\frac{n}{\sigma} + \sum_{i=1}^n (x_i - \mu)^2 \sigma^{-3} = 0$$

$$\hat{\sigma}^2_{MLE} = \frac{\sum_{i=1}^{n}(x_i - \hat{\mu})^2}{n}$$

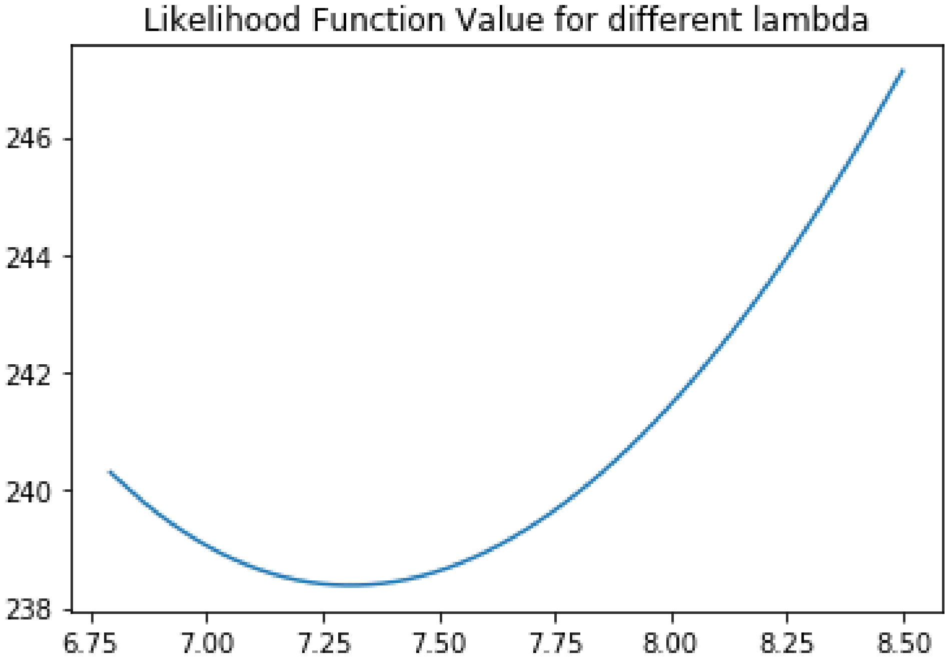
Then from the first question of question4, we know:

$$\hat{\mu}_{MAP} = \frac{\alpha\sigma + n\bar{x}\beta}{\sigma + n\beta} \to \bar{x} = \hat{\mu}_{MLE}(n \to \infty)$$

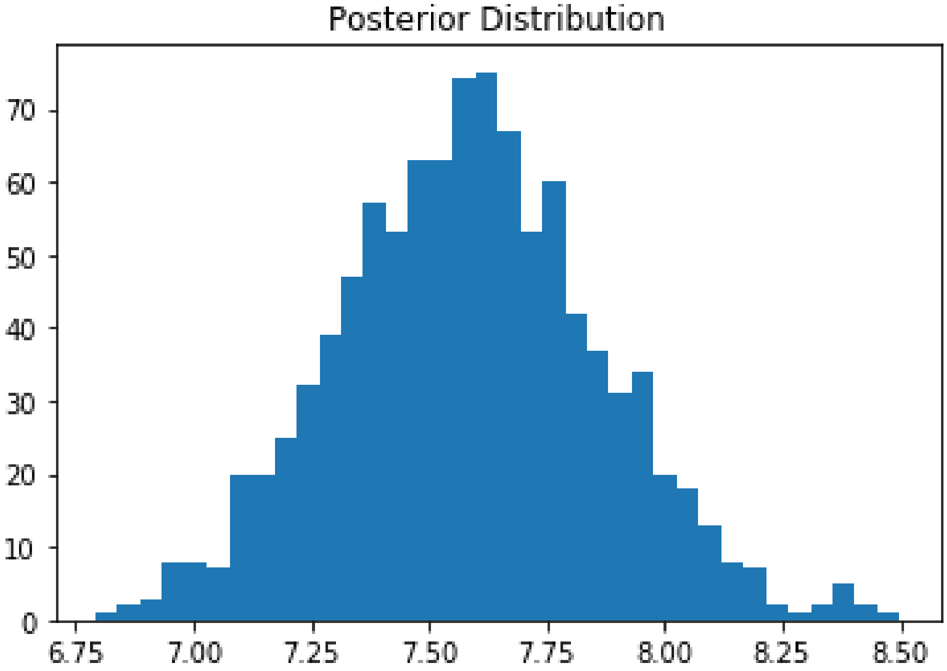
$$\hat{\sigma^2}_{MAP} = \frac{\nu + \Sigma_{i=1}^{n}(x_i - \mu)^2/2}{\tau + n/2 - 1} \to \frac{\Sigma_{i=1}^{n}(x_i - \mu)^2}{n} = \hat{\sigma^2}_{MLE}(n \to \infty)$$

7 Programming a Gibbs Sampler

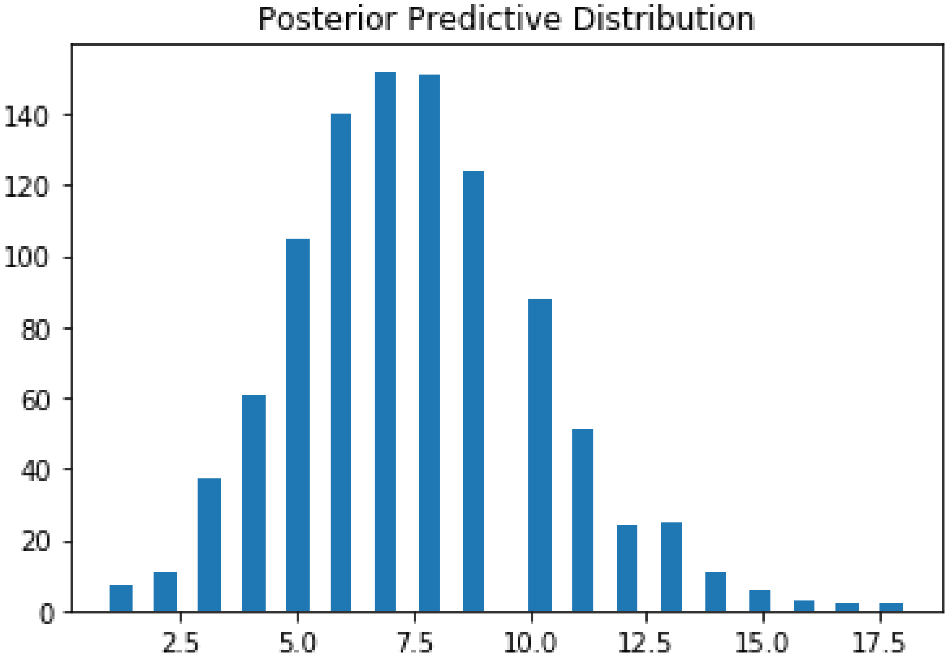
1. Likelihood vs λ:



2. Posterior Distribution:



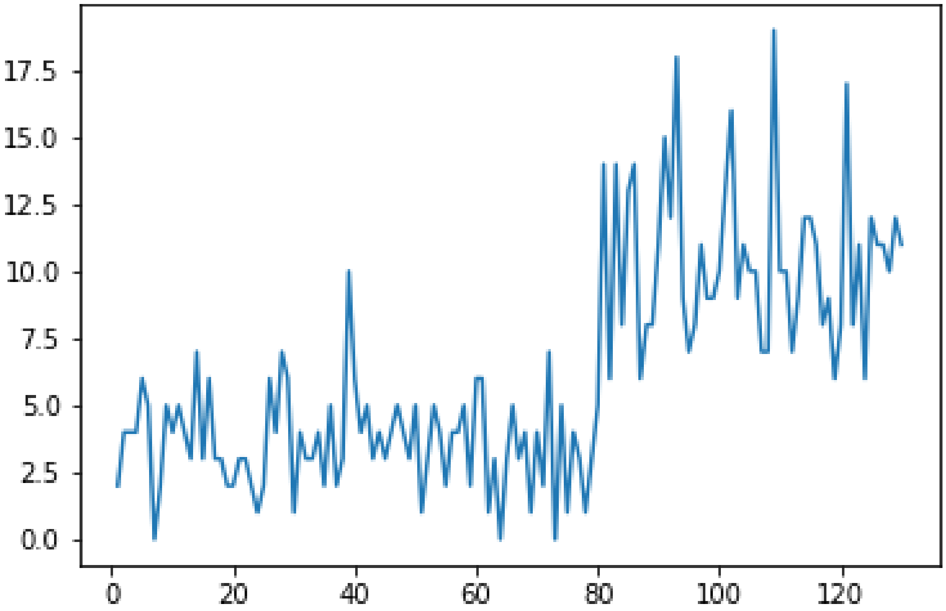
3. Posterior Predictive distribution:



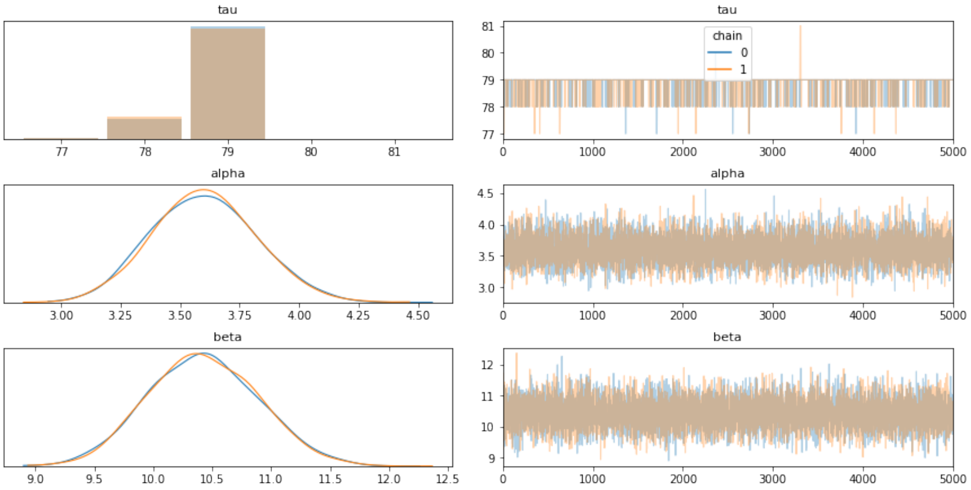
Since there is no dependencies in gibbs sampling process, we do not need to consider the convergence problem.

8 Change points models

First we plot the raw data as below:



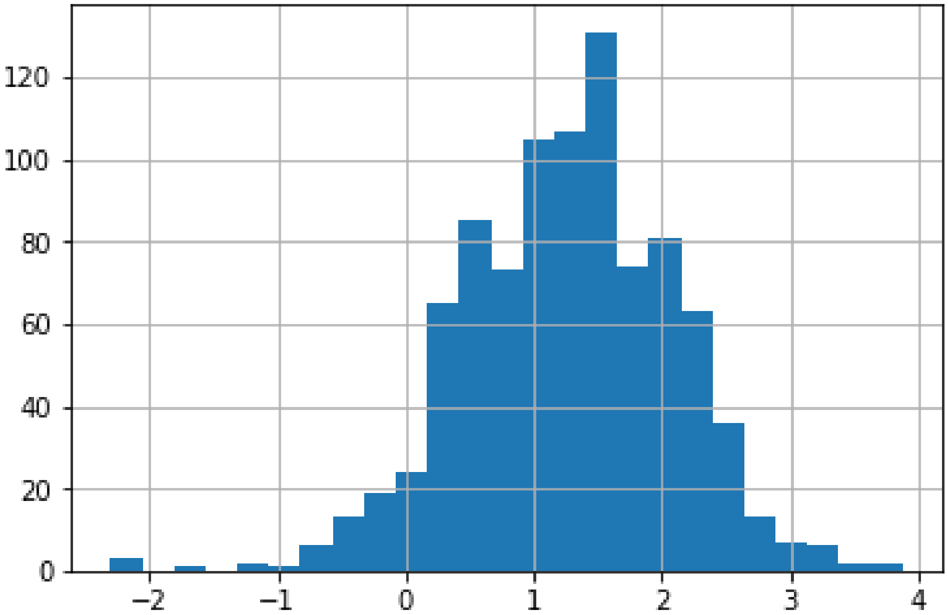
Then after applying change points model:



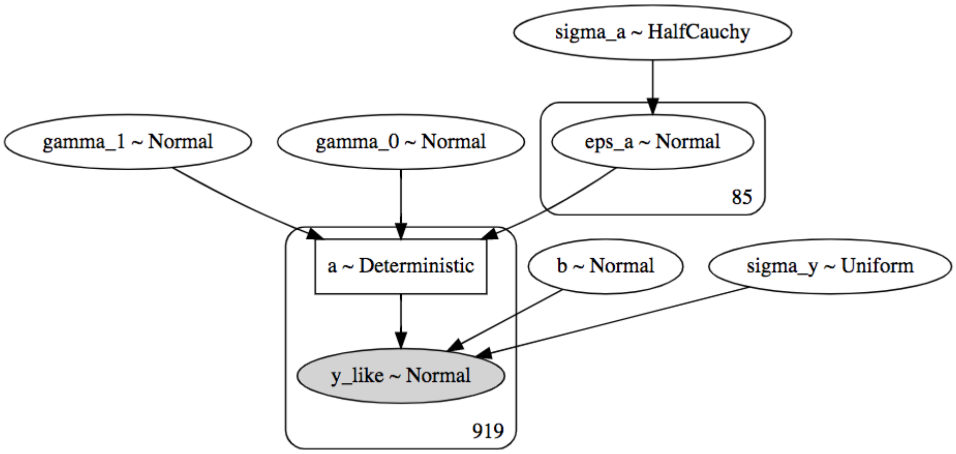
From the above plot, we can see from the likelihood plot(the first one) that the change point occurs at 79 with a very high probability. From the trace plots of α and β we know that the sampler has converged.

9 Programming a hierarchical model using PYMC3

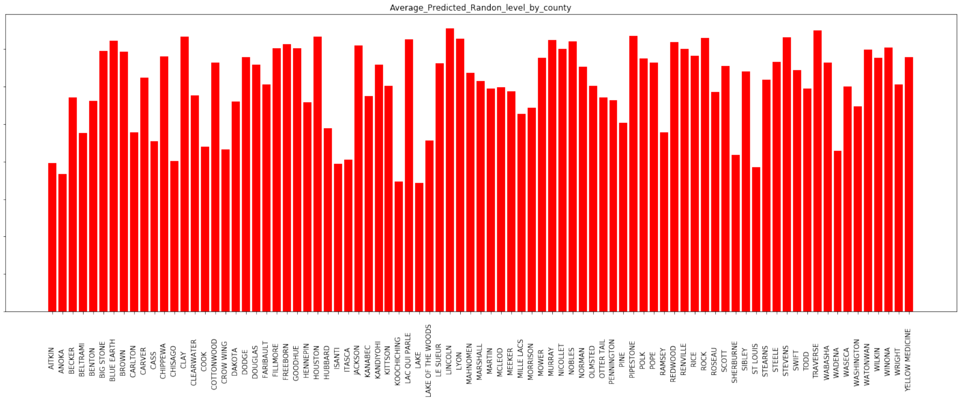
First we plot the distribution of radon levels in MN (log scale):



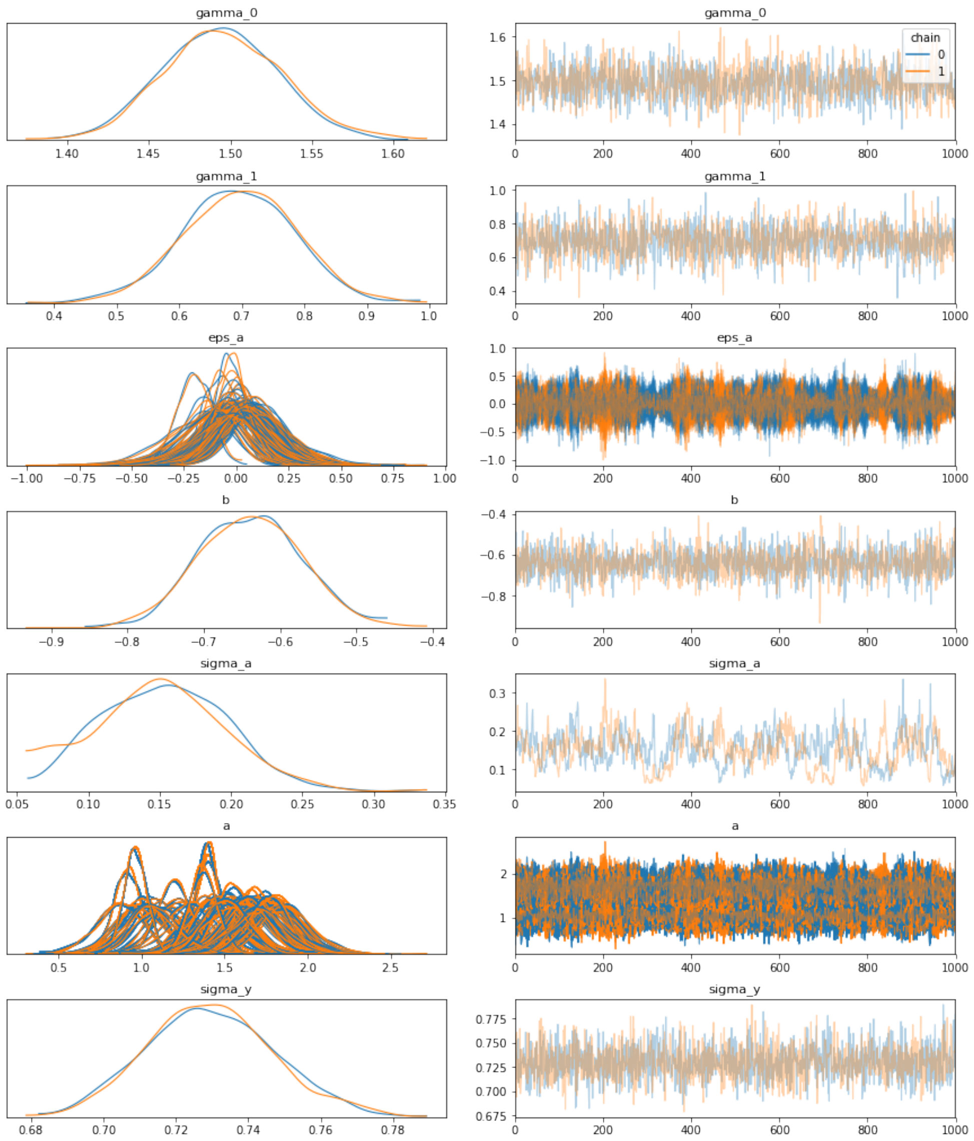
We incorporate a house-level predictor (floor or basement) as well as a county-level predictor (uranium). And the architecture of the model is shown as below:



And using the model, our prediction shows that STEVENS is the county with highest level of radon and KOOCHICHING has the lowest. Here is a prediction of all counties:



And we also use trace plots to diagnose the convergence. From the trace plots of each parameter below, our sampler has converged.



10 Interview Questions

10.1

a. SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form, usually it will reduce the dimension of input. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

b. A slack variable is a variable that is added to an inequality constraint to transform it into an equality. Slack variables can be defined as the ratio between the distance from a training point to a marginal hyperplane, and half of margin. Large penalty will lead to smaller margin. So if we want our model to be very sensitive to errors, we need to put large penalty.

c. RBF kernel function is special because it range in 0 and 1 and only increase with the distance between two points. So it is invariant to the translation. In this case, slack variable will no longer be a problem.

d. Suppose now our target function is:

$$min||w^2|| + c\sum_{i=1} n\zeta_i$$

If we write $||w^2|| = \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i^T x_j) + b*$, a dual form of the problem will be:

$$w_{\alpha \geq 0} \sum_i \alpha_i - 1/2 \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

Solving the dual problem, we obtain the α_i (here $\alpha_i = 0$ or all but a few points) - the support vectors). In order to classify a query point, we calculate

$$w^T x + w_0 = \left(\sum_{i=1}^{n} \alpha_i y_i x_i \right)^T x + w_0 = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + w_0$$

This term is very efficiently calculated if there are only few support vectors. Further, since we now have a scalar product only involving data vectors, we may apply the kernel trick.

e. n^2 when C is relatively small and n^3 when C is very large.

10.2

10.3

The hierarchical models allow easy addition and deletion of new information. Data at the top of the Hierarchy is very fast to access. When data is sparse, there is no sophisticated relationships between data. In this case, hierarchical model will capture the underlying pattern in a more efficient way and provide better model fit.