# Assignment 2

*Arshia Singh*

*February 4, 2020*

# Question 1: Convexity

## 1

$f(x) = \sum_{i=1}^{\infty} ||x_i||_p; p > 0$

If the above funcion is a norm, we know that it is positive, definite, absolutely scaleable, and that it is subject to the triangle inequality. Applying this inequality ($||v + w|| \leq ||v|| + ||w||$ ) we know that norms are convex since:

$||\lambda x + (1 - \lambda)y|| \leq ||\lambda x|| + ||(1 - \lambda)y|| = \lambda||x|| + (1 - \lambda)||y||$

We now need to show that the p-norm is a norm, and therefore convex. However, if we look at the case of $n = 2$ with $x_1 = 1$ and $x_2 = 1$ we find that:

$f(x) = \sum_{i=1}^{\infty} ||x_i||_p = (\sum_{i=1}^{\infty} |x_i|^p)^{\frac{1}{p}}$

$||1 + 1|| = ||2|| = 2^{\frac{1}{p}} \geq 2 = 1 + 1 = ||1|| + ||1||$ if $0 < p < 1$

This shows that the triangle inequality does not hold for $0 < p < 1$ and that $f(x)$ is therefore not a norm for these values of p, and so NOT convex for all values of p.

## 2

$f(d) = k(x, x') - k'(x, x')$

In this case We can look at a specific stationary and positive definite kernel to see how this is not the case. For the radial basis kernel, which is stationary and positive definite, $k(d) = e^(\frac{-d^2}{2l^2})$ and $k'(d) = \frac{-d}{l^2} e^{-d^2} 2l^2$ . In this case, $k(d)$ is convex, but $k'(d)$ is not. When $k'(d)$ is subtracted from $k(d)$ the resulting function is NOT convex in this case - its second derivative changes signs.

## 3

$f(d) = k(x, x') \times k'(x, x') - b$

Let's again look at the radial basis kernel $k(d) = e^(\frac{-d^2}{2l^2})$ and $k'(d) = \frac{-d}{l^2} e^{-d^2} 2l^2$ . Now, $f(d) = \frac{-d}{l^2} e^(\frac{-2d^2}{2l^2}) - b$ and $f''(d) = \frac{e^{-d^2/l^2}(6dl^2 - 4d^3)}{l^6}$ which, as in part 2, changes signs, meaning the resulting function is NOT convex.

## 4

$f(x) = ||x||_p - max(0, x)$ for $p > 0$

Again, this is NOT convex for $0 < p < 1$ . Using the example from part 1 we have the case of $n = 2$ with $x_1 = 1$ and $x_2 = 1$ where we find that:

$f(x) = \sum_{i=1}^{\infty} ||x_i||_p + max(0, x) = (\sum_{i=1}^{\infty} |x_i|^p)^{\frac{1}{p}} + max(0, x)$

$||1 + 1|| - max(0, 1) = ||2|| - 1 = 2^{\frac{1}{p}} - 1 \geq 1 = 1 + 1 - max(0, 1) = ||1|| + ||1|| - max(0, 1)$ if $0 < p < 1$ .

This shows that the triangle inequality does not hold for $0 < p < 1$ and that $f(x)$ is therefore not a norm for these values of p, and so NOT convex for all values of p.

## 5

$f(x) = ||x||_p + max(0, x)$ for $p > 0$

Again, this is NOT convex for $0 < p < 1$ . Using the example from part 1 we have the case of $n = 2$ with $x_1 = 1$ and $x_2 = 1$ where we find that:

$f(x) = \sum_{i=1}^{\infty} ||x_i||_p + max(0, x) = (\sum_{i=1}^{\infty} |x_i|^p)^{\frac{1}{p}} + max(0, x)$

$||1 + 1|| + max(0, 1) = ||2|| + 1 = 2^{\frac{1}{p}} + 1 \geq 3 = 1 + 1 + max(0, 1) = ||1|| + ||1|| + max(0, 1)$ if $0 < p < 1$ .

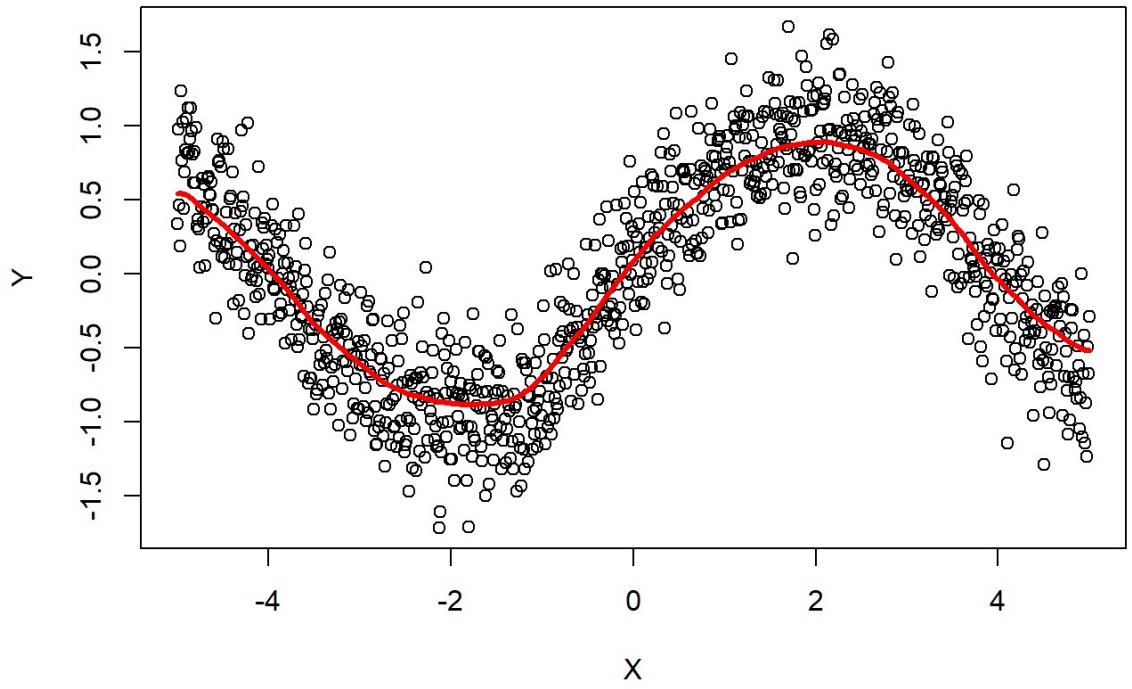
This shows that the triangle inequality does not hold for $0 < p < 1$ and that $f(x)$ is therefore not a norm for these values of p, and so NOT convex for all values of p.

# Question 2: Kernel Regression

## Part 1

### Kernel 1 - Exponential

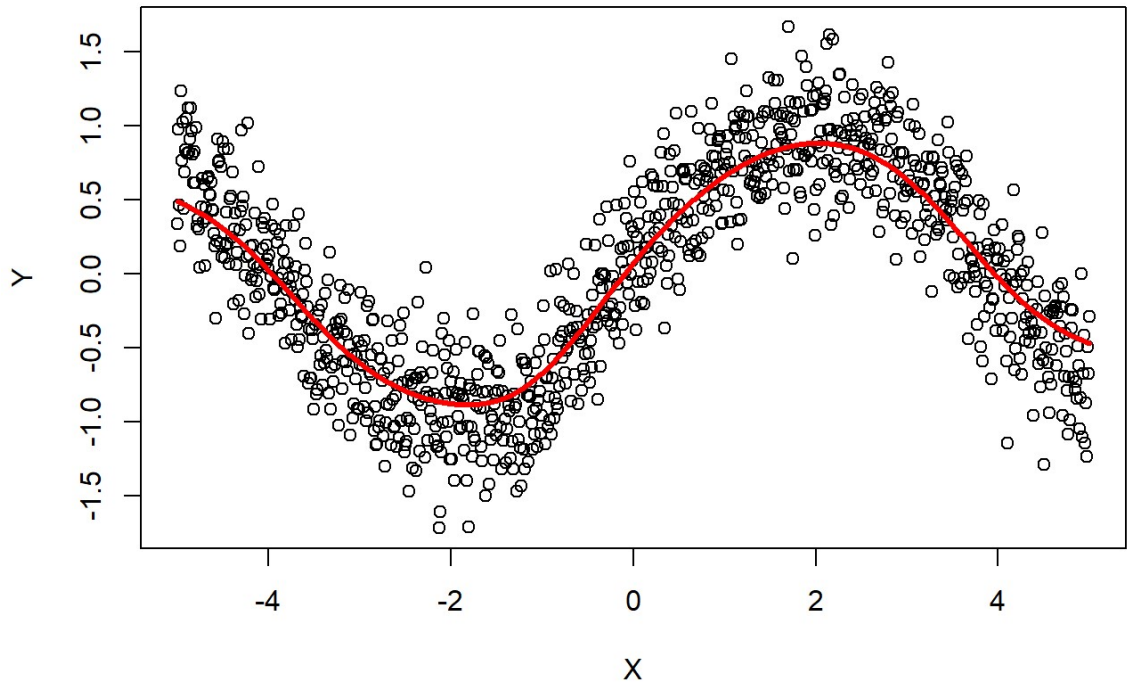
```
data <- read.csv("kernel_regression_1.csv")

X <- data$x
Y <- data$y
kreg <- function(x, X, Y) {
  Kx <- sapply(X, function(Xi) exp(-3*abs(x-Xi)))
  W <- Kx / rowSums(Kx)
  drop(W %*% Y)
}

xGrid <- seq(-5, 5, l = 1001)

plot(X, Y)
lines(xGrid, kreg(x = xGrid, X = X, Y = Y), col = 2, lwd=3)
```




####

Kernel 2 - Radial Basis function

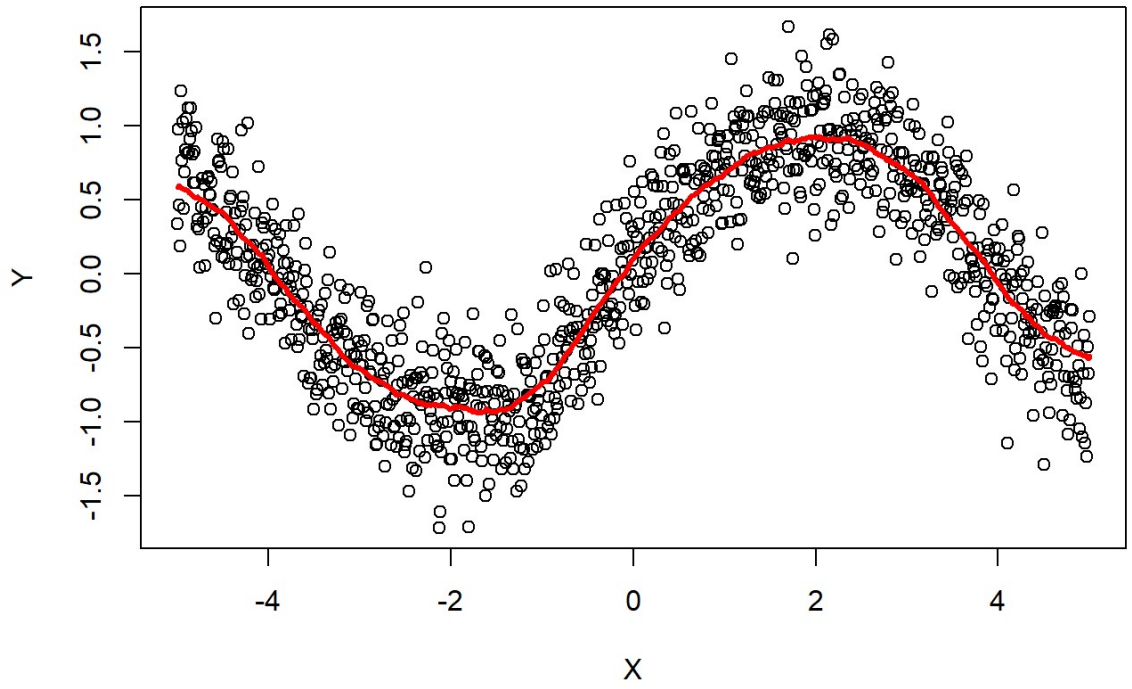
```
kreg <- function(x, X, Y) {
  Kx <- sapply(X, function(Xi) exp(-2*abs(x-Xi)^2))
  W <- Kx / rowSums(Kx)
  drop(W %*% Y)
}

set.seed(12345)
xGrid <- seq(-5, 5, l = 1001)

plot(X, Y)
lines(xGrid, kreg(x = xGrid, X = X, Y = Y), col = 2, lwd=3)
```




####

Kernel 3

```
kreg <- function(x, X, Y) {
  Kx <- sapply(X, function(Xi) abs(x-Xi)<0.5)
  W <- Kx / rowSums(Kx)
  drop(W %*% Y)
}

set.seed(12345)
xGrid <- seq(-5, 5, l = 1001)

plot(X, Y)
lines(xGrid, as.numeric(kreg(x = xGrid, X = X, Y = Y)), col = 2, lwd=3)
```




# Part 2

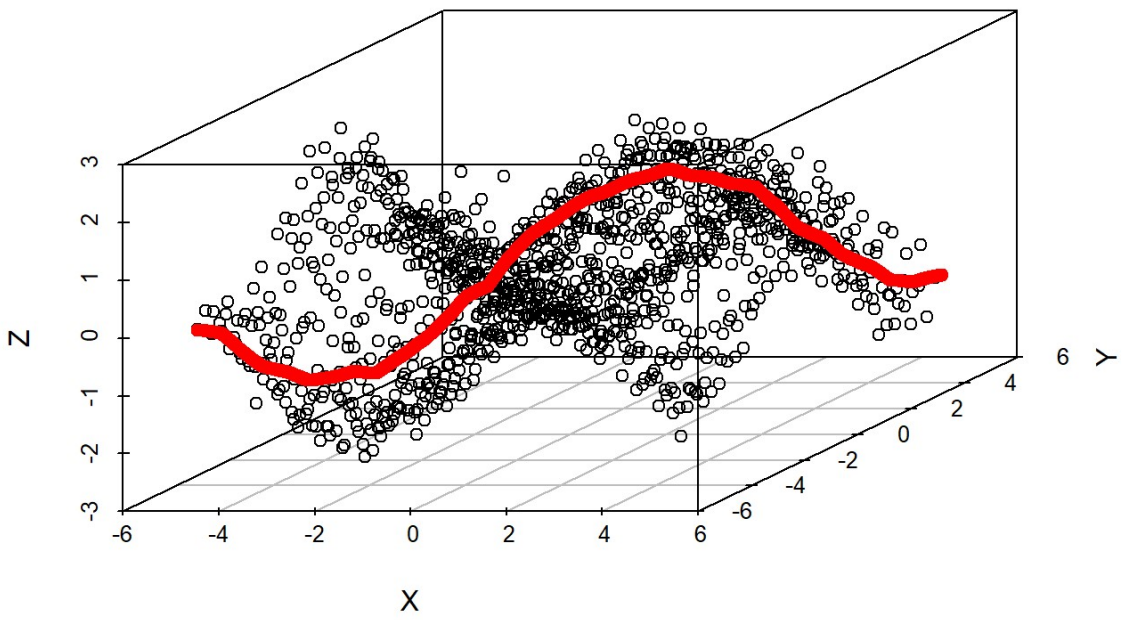
The uniform kernel produces the least smooth estimates, while the radial basis function seems to produce the most smooth estimate function. The uniform and exponential graphs also have a more flat section around -2 whereas the radial basis function produces a more rounded function. Other than that, they seem to produce similar looking graphs.

# Part 3

## 1

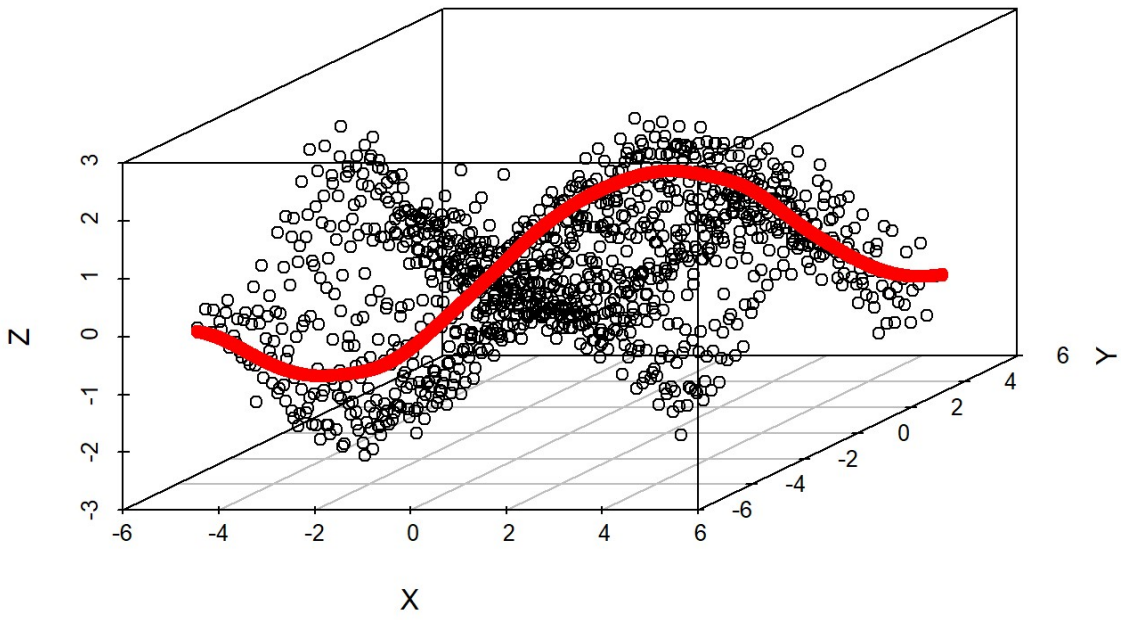
### Exponential

```
library(scatterplot3d)
data <- read.csv("kernel_regression_2.csv")

X <- data$x
Z <- data$z
Y <- data$y
kreg <- function(x, X, y, Y, Z) {
  Kx <- sapply(X, function(Xi) exp(-3*abs(x-Xi)))
  Ky <- sapply(Y, function(Yi) exp(-3*abs(y-Yi)))
  W <- Kx*Ky / rowSums(Kx*Ky)
  drop(W %*% Z)
}

xGrid <- seq(-5, 5, l = 1156)
yGrid <- seq(-5, 5, l = 1156)

s3d <- scatterplot3d(X,Y,Z)
s3d$points3d(xGrid, yGrid, z=kreg(x=xGrid, X = X, y=yGrid, Y = Y, Z=Z), col=2)
```



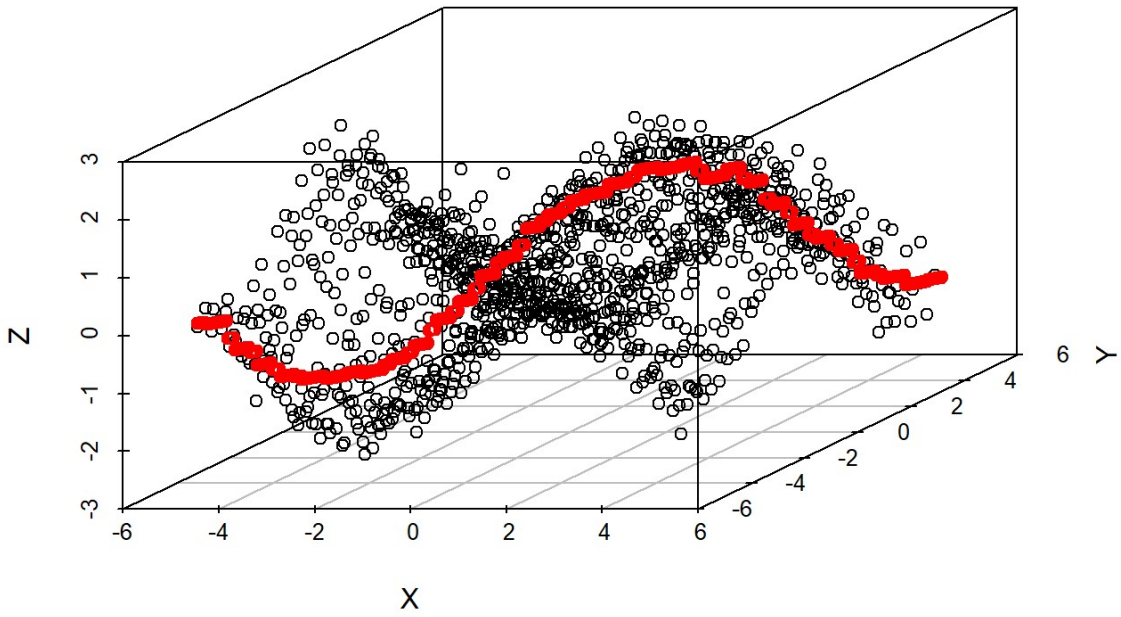
## Radial Basis

```
kreg <- function(x, X, y, Y, Z) {
  Kx <- sapply(X, function(Xi) exp(-2*abs(x-Xi)^2))
  Ky <- sapply(Y, function(Yi) exp(-2*abs(y-Yi)^2))
  W <- Kx*Ky / rowSums(Kx*Ky)
  drop(W %*% Z)
}

s3d <- scatterplot3d(X,Y,Z)
s3d$points3d(xGrid, yGrid, z=kreg(x = xGrid, X = X, y = yGrid, Y = Y, Z=Z), col=2)
```



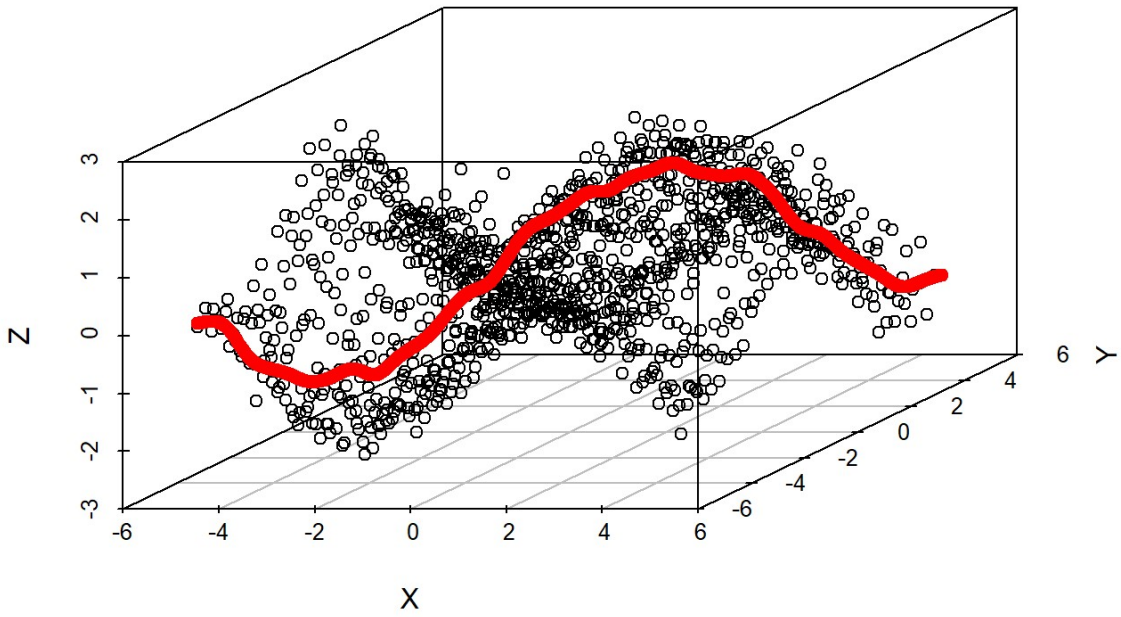

## Uniform

```
kreg <- function(x, X, y, Y, Z) {
  Kx <- sapply(X, function(Xi) abs(x-Xi)<0.5)
  Ky <- sapply(Y, function(Yi) abs(y-Yi)<0.5)
  W <- Kx*Ky / rowSums(Kx*Ky)
  drop(W %*% Z)
}

s3d <- scatterplot3d(X,Y,Z)
s3d$points3d(xGrid, yGrid, z=kreg(x = xGrid, X = X, y = yGrid, Y = Y, Z=Z), col=2)
```




2

```
kreg <- function(x, X, y, Y, Z) {
  Kx <- sapply(X, function(Xi) exp(-8*abs(x-Xi)^2))
  Ky <- sapply(Y, function(Yi) exp(-8*abs(y-Yi)^2))
  W <- Kx*Ky / rowSums(Kx*Ky)
  drop(W %*% Z)
}

s3d <- scatterplot3d(X,Y,Z)
s3d$points3d(xGrid, yGrid, z=kreg(x = xGrid, X = X, y = yGrid, Y = Y, Z=Z), col=2)
```




Here

we can see that with an increase in the bandwidth, the kernel became much more sensitive to the variance of the data - the new surface of fitted values is not nearly as smooth as when the bandwidth was 2.

# Question 4: Calculating the conjugate distributions

## 1

$\mu \sim N(\tau, \nu)$ , $\sigma^2 \sim InverseGamma(\alpha, \beta)$ where $X \sim N(\mu, \sigma^2)$ and $\tau, v, \alpha, \beta$ are all constant.

For n observations the data generating function is: $P(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

and the conjugate priors are:

$P(\mu|\tau, \nu) \propto (\nu^2)^{-1/2} e^{-(\mu-\tau)^2/(2\nu^2)}$

$P(\sigma^2|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-\alpha-1} e^{-\beta/\sigma^2}$

The conditional posterior is:

$\sigma^2_{post} = n\sigma^2 + \nu$ The marginal posterior is defined as:

$P(\mu, \sigma^2|x, \tau, \nu, \alpha, \beta) \propto P(\sigma^2|\alpha, \beta)\Pi(\mu|\tau, \nu)P(x|\mu, \sigma^2)$

$P(\mu, \sigma^2|x, \tau, \nu, \alpha, \beta) \propto ((\sigma^2)^{-\alpha-1} e^{-\beta/\sigma^2})((\nu^2)^{-1/2} e^{-(\mu-\tau)^2/(2\nu^2)})(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}})$

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

so the posterior is:

$$\mu, \sigma^2|x \sim Normal - InverseGamma(\frac{\nu\tau + n\bar{x}}{\tau + n}, \nu + n, \alpha + n/2, \beta + \frac{1}{2}\sum_{i=1}^n (x_i - \bar{x})^2 + \frac{n\nu}{\nu + n}\frac{(\bar{x} - \tau)^2}{2})$$

## 2

$p \sim Dirichlet(a_1, \ldots, a_k)$ where $X\ Multinomial(p)$ where $p = (p_1, \ldots p_k)$
For n observations the data generating function is:
$p(x|p) = p_1^{x_1} p_2^{x_2} \ldots p_k^{x_k}$ And the conjugate prior is:
$p(p|a) = K(a) p_1^{a_1 - 1} p_2^{a_2 - 1} \ldots p_k^{a_k - 1}$ where $K(a) = \frac{\Gamma(\sum_{k=1}^{K} a_k)}{\Pi_{k=1}^{K} \Gamma(a_k)}$ So the posterior is:
$p(p|x, a) \propto p_1^{\sum_{n=1}^{N} x_{n1} + a_1 - 1} p_2^{\sum_{n=1}^{N} x_{n2} + a_2 - 1} \ldots p_k^{\sum_{n=1}^{N} x_{nk} + a_k - 1}$ $p|x \sim Dirichlet(a + \sum_{i=1}^{n} x_i)$

## 3

$\lambda \sim Gamma(\alpha, \beta)$ where $X \sim Poisson(\lambda)$
For n observations the data generating function is:
$P(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$
and the conjugate prior is:
$\Pi(\lambda) = \frac{\lambda^{\alpha - 1} exp(\frac{-\lambda}{\beta})}{\Gamma(\alpha)\beta^{\alpha}}$
The posterior is defined as:
$Pi(\lambda|x) = \frac{P(\lambda, x)}{\Pi(x)} = \frac{P(x|\lambda)\Pi(\lambda)}{P(x)}$ In this case we have that:
$\Pi(\lambda|x) = \frac{\Pi_{i=1}^{n} \lambda^{x_i} e^{-\lambda} \frac{\lambda^{\alpha - 1} exp(-\lambda/\beta)}{\Gamma(\alpha)\beta^{\alpha}}}{P(x)}$
$\Pi(\lambda|x) \sim \lambda^{\sum_{i=1}^{n} x_i} e^{n\lambda} \lambda^{\alpha - 1} exp(-\lambda/\beta)$
$\Pi(\lambda|x) \sim \lambda^{\bar{x} + \alpha - 1} exp(-\lambda(n + 1/\beta))$
$\Pi(\lambda|x) \sim \lambda^{\bar{x} + \alpha - 1} exp(-\lambda(\frac{n\beta + 1}{\beta}))$
So the posterior is:
$\lambda|x\ Gamma(\bar{n}x + \alpha, \frac{\beta}{n\beta + 1})$

# Question 6: General Questions

## 1

Explain in your own words the difference between the posterior distribution and posterior predictive distribution.

The posterior distribution is the “current” distribution of a parameter given prior information, or data. It describes the distribution of the “current” parameter’s possible values. When using a conjugate prior, the expected value of the posterior distribution is a weighted average of prior weights and observed values used to describe a random variable or unknown quantity. The posterior predictive distribution, on the other hand, describes the distribution of “future” data. It tells us what the distribution of unsampled data under the parameter would look like, assuming the samples are homogenous.

## 2

Which one would you use to predict future values of X? Explain your rationale.

I would use the posterior predictive distribution because it tells us what the distribution of unsampled data are likely to look like.

## 3

# Interview Questions

## 1

Support vector machines classify data via decision boundaries. If the classes are linearly separable SVM will choose the boundary that maximizes the margin between the classes. Maximizing the margin involves minimizing a constrained optimization problem for a quadratic function using the lagrangian. This is the primal form of the optimization problem, which in high dimensional spaces can be extremely intensive computationally because it requires all the points to be mapped in this higher dimension. If we instead only compute the dot-product, or kernel function, between each pair of points in the new dimension we don't have to map all the points and can use the outputs to determine the support vectors, and therefore the maximizing-margin hyperplane. This is the dual version of the optimization. As n, the number of dimensions, increases, the primal (which solves a quadratic optimization) becomes increasingly expensive to compute, while the dual remains relatively inexpensive. This also applies to the classification of new points - in the primal each new point must be evaluated in n-dimensions while the dual allows us to avoid explicit mapping in favor of comparing the point to the existing support vectors. This is the kernel trick - it allows the decision boundary to be non-linear by transforming the existing non-linear feature space into dimensions that do allow for linearly separable data using inner products. In the new feature space that uses the transformed variables the classes are now separable by a linear hyperplane, without a need to map those points explicitly.
The slack variable determines how strong the condition of separability must be - the larger the slack variable, the higher the tolerance of the algorithm for "misclassified" points that fall on the wrong side of the margin. This is useful in cases where the boundary between two classes overlaps or if there are outliers that fall well into another class's region. It is helpful in these cases to graph the data to determine whether it is more prudent to use a simpler kernel with slack, or a kernel like the radial basis in high dimensions to find a separable boundary. In the case of the radial basis kernel we don't have to worry about setting a slack variable because the radial basis kernel can project into infinite space and can therefore always find a separable boundary for the classes. It is important to find balance between perfect separability and slack in order to avoid either overfitting the data or allowing for too much error when classifying new points.

## 2

Show that for the class of distributions in the regular exponential family that the mean update function is a weighted average of the prior distribution and observations. Hint: See Statistical Machine Learning, by Han Liu and Larry Wasserman, 2014, pg. 312

If $p(.|\theta)$ is in the regular exponential family then the density can be expressed as:
$p(x|\theta) = exp(\theta^T x - A(\theta))$
The conjugate prior for this density is:
$\pi_{x_0,n_0}(\theta) = \frac{exp(n_0 x_0^T \theta - n_0 A(\theta))}{\int exp(n_0 x_0^T \theta - n_0 A(\theta)) d\theta}$
This is conjugate because:
$p(x|\theta)\pi_{x_0,n_0}(\theta) = exp(\theta^T x - A(\theta)) exp(n_0 x_0^T \theta - n_0 A(\theta))$
$p(x|\theta)\pi_{x_0,n_0}(\theta) \propto \pi_{\frac{x}{1+n_0} + \frac{n_0 x_0}{1+n_0}, 1+n_0}(\theta)$
Therefore the updates look like:
$p(\theta|X_1, \ldots, X_n) = \pi_{\frac{n\bar{X}}{n+n_0} + \frac{n_0 x_0}{n+n_0}, n+n_0}(\theta)$
$p(\theta|X_1, \ldots, X_n) \propto exp((n+n_0)(\frac{n\bar{X}}{n+n_0} + \frac{n_0 x_0}{n+n_0})^T \theta - (n+n_0)A(\theta))$
with:
$x_0' = \frac{x}{1+n_0} + \frac{n_0 x_0}{1+n_0}$
Therefore updates for families in these distributions use a weighted average of the prior distribution and observations.

# 3

Why do hierarchical models provide better model fits and regularization when data is sparse?
In the case of sparse data, hierarchical models (which typically use a form of partial pooling) can provide better model fits and regularization relative to non-hierarchal approaches such as pooling or no pooling. When we don't use any form of pooling, each group, or even data point, is treated distinctly. This can cause problems for sparse data because it can lead to high variance for sparse groups and can ignore relationships that might exist between the groups, meaning the modeling is not leveraging all possible information to create a good fit. In approaches with full pooling, the population is assumed to be homogenous rather than heirarchical in its structure. This can have an extremely strong regularization effect of the data, as all data points are pulled towards the mean of the entire group. This also leads to an overestimation of variance relative to a partial pooling, or heirarchical approach. When data is modeled through group relationships we are able to reduce the variance for groups with sparse data while adequately regularizing without negating group information.