1. (5 points) 1. You are given a data-set with 1000 data points generated from a mixture of some distribution in the file A2Q1Data.csv.

   (a) Determine which probabilisitic mixture could have generated this data (It is not a Gaussian mixture). Derive the EM algorithm for your choice of mixture and show your calculations. Write a piece of code to implement the algorithm you derived by setting the number of mixtures K = 4. Plot the log-likelihood (averaged over 100 random initializations) as a function of iterations.

Solution:
Data is generated from Bernoulli Distribution. All the data points are from $\{0, 1\}$.

Derivation of Bernoulli EM algorithm:
Mixture of discrete variables: Bernoulli distribution.
PMF of Bernoulli distribution: where "p" is the parameter to be estimated and the data points
$x_i \sim Bernoulli(p)$.

$$P(x; p) = p^{x_i}(1-p)^{1-x_i}$$

Bernoulli Mixture Model:
K is the number of Clusters. Z_i be the Latent variable.

$$Choose\ Z_i \in \{1, ..., K\},\ \forall i$$

$$P(Z_i = l) = \pi_l$$

be the mixing probability π_l of i^{th} data point belonging to l^{th} cluster. such that

$$\sum_{l=1}^{k} \pi_l = 1,\ \pi_l \geq 0$$

Parameters to estimates:

$$p_1, .., p_k$$

$$\pi_1, ...\pi_k,\ \Sigma_{l=1}^{k}\pi_l = 1$$

Let these parameters be θ Taking Maximum likelihood:

$$L(x_1, x_2, .., x_n, \theta) = \Pi_{i=1}^{n}\mathbf{P}(\mathbf{X}; \theta)$$

$$= \Pi_{i=1}^{n}\ \Sigma_{k=1}^{k}\ P(x_i, Z_i = k; \theta)$$

taking Log,

$$logL(x_1, x_2, .., x_n, \theta) = \Sigma_{i=1}^{n}\ log(\Sigma_{k=1}^{k}\ P(x_i, Z_i = k; \theta))$$

Log likelihood doesn't give close forms to these parameters due to summation within log.
By Using Jensen's Inequality: We, add λ_k^i be

$$logL(x_1, x_2, .., x_n, \theta) \ \geq \ \Sigma_{i=1}^n \ log(\Sigma_{k=1}^k \ \lambda_k^i \ \frac{P(x_i, Z_i = k; \theta)}{\lambda_k^i})$$

$$\geq \ \Sigma_{i=1}^n \ \Sigma_{k=1}^k \ \lambda_k^i \ (logP(x_i, Z_i = k; \theta) - log(\lambda_k^i))$$

Where, $P(x_i, Z_i = k; \theta) = P(z_i = k) * P(x_i/(z_i = k)) = \pi_k * p_k^{x_i} * (1 - p_k)^{(1-x_i)}$

$$\geq \ \Sigma_{i=1}^n \ \Sigma_{k=1}^k \ \lambda_k^i \ (logP(x_i, Z_i = k; \theta) - log(\lambda_k^i))$$

$$\geq \ \Sigma_{i=1}^n \ \Sigma_{k=1}^k \ \lambda_k^i \ (log(\pi_k * p_k^{x_i} * (1 - p_k)^{(1-x_i)}) - log(\lambda_k^i))$$

$$\geq \ \Sigma_{i=1}^n \ \Sigma_{k=1}^k \ \lambda_k^i \ ((log(\pi_k) + log(p_k^{x_i}) + log((1 - p_k)^{(1-x_i)}) - log(\lambda_k^i))$$

this is Log likelihood we use in EM algorithm
Finding Parameters θ in BMM:

$$p_1, .., p_k$$

$$\pi_1, ...\pi_k, \ \Sigma_{l=1}^k \pi_l = 1$$

$$logL(x_1, .., x_n, \theta) \ \geq \ \Sigma_{i=1}^n \ \Sigma_{k=1}^k \ \lambda_k^i \ ((log(\pi_k) + log(p_k^{x_i}) + log((1 - p_k)^{(1-x_i)}) - log(\lambda_k^i))$$

Estimate $\hat{p_k}$

Taking $\frac{\partial logL}{\partial p_k}$,

$$\sum_{i=1}^{n} (\frac{\lambda_k^i \ x_i}{p_k} - \frac{\lambda_k^i (1 - x_i)}{1 - p_k}) = 0$$

$$\sum_{i=1}^{n} \frac{\lambda_k^i \ x_i}{p_k} = \sum_{i=1}^{n} \frac{\lambda_k^i (1 - x_i)}{1 - p_k}$$

$$\frac{1 - p_k}{p_k} = \frac{\sum_{i=1}^n \ \lambda_k^i (1 - x_i)}{\sum_{i=1}^n \ \lambda_k^i \ x_i}$$

$$p_k = \frac{\sum_{i=1}^n \ \lambda_k^i \ x_i}{\sum_{i=1}^n \ \lambda_k^i (1 - x_i) + \sum_{i=1}^n \ \lambda_k^i \ x_i}$$

$$\boxed{\hat{p_k} = \frac{\sum_{i=1}^n \ \lambda_k^i \ x_i}{\sum_{i=1}^n \ \lambda_k^i}}$$

Estimate $\hat{\pi_k}$:

$$\sum_{i=1}^{n}\sum_{k=1}^{k} \lambda_k^i \ log(\pi_k) \ , \ S.T \ \Sigma_{l=1}^{k}\pi_l = 1$$

Constraint optimization problem : Lagrangian

$$\sum_{i=1}^{n}\sum_{k=1}^{k} \lambda_k^i \ log(\pi_k) + \gamma(\Sigma_{l=1}^{k}\pi_l - 1)$$

Taking $\frac{\partial logL}{\partial \pi_k}$,

$$\sum_{i=1}^{n} \frac{\lambda_k^i}{\pi_k} + \gamma = 0$$

$$\sum_{i=1}^{n} \ \lambda_k^i + \gamma \ \pi_k = 0 \ . \ .(1)$$

$$\sum_{i=1}^{n}\sum_{k=1}^{k} \lambda_k^i + \gamma \ \sum_{k=1}^{k} \pi_k = 0, \ \forall k$$

$\sum_{k=1}^{k} \lambda_k^i = 1$ and $\sum_{k=1}^{k} \pi_k = 1$

$$n + \gamma = 0$$

$$\gamma = -n$$

In (1),

$$\sum_{i=1}^{n} \lambda_k^i + (-n) \ \pi_k = 0$$

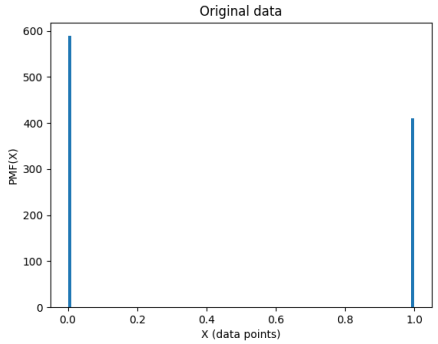
$$\boxed{\pi_k = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}}$$

Estimate λ_k^i

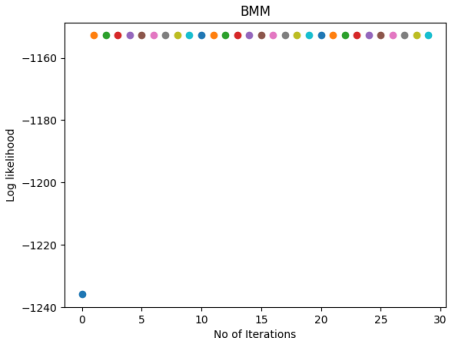
$$\lambda_k^i = \frac{P(x_i, z_i = k; \theta)}{P(x_i; \theta)}$$

$$\boxed{\lambda_k^i = \frac{\pi_k p_k^{x_i}(1-p_k)^{1-x_i}}{\sum_{l=1}^{k} \pi_l p_l^{x_i}(1-p_l)^{1-x_i}}}$$

Original Data



Log likelihood of data



(b) Assume that the same data was infact generated from a mixture of Gaussians with 4 mixtures. Implement the EM algorithm and plot the log-likelihood (averaged over 100 random initializations of the parameters) as a function of iterations How does the plot compare with the plot from part (i)? Provide insights that you draw from this experiment.

Solution: Gaussian Mixture Model:
Let θ be parameters to estimates (μ, σ^2, π);

$$\mu_1, ..., \mu - k$$

$$\sigma_1^2, ...\sigma_k^2$$

$$\pi_1, ..., \pi_k$$

Taking Maximum likelihood:

$$L(x_1, x_2, .., x_n, \theta) = \Pi_{i=1}^n \mathbf{P}(\mathbf{X}; \theta)$$

$$= \Pi_{i=1}^n \ \Sigma_{k=1}^k \ P(x_i, Z_i = k; \theta)$$

taking Log,

$$logL(x_1, x_2, .., x_n, \theta) = \Sigma_{i=1}^n \ log(\Sigma_{k=1}^k \ P(x_i, Z_i = k; \theta))$$

Estimate $\hat{\mu_{kML}}$:
Taking $\frac{\partial logL}{\partial \mu_k}$,

$$\boxed{\hat{\mu_{ML}^k} = \frac{\sum_{i=1}^n \lambda_k^i \ x_i}{\sum_{i=1}^n \lambda_k^i}}$$

Estimate $\hat{\sigma_k^2} = S_k$ **:**
Taking $\frac{\partial logL}{\partial \sigma}$,

$$\boxed{\hat{\sigma_k^2} = S_k = \frac{\sum_{i=1}^n \lambda_k^i (x_i - \hat{\mu}_{kML})^2}{\sum_{i=1}^n \lambda_k^i}}$$

Estimate $\hat{\pi_k}$:
Taking $\frac{\partial logL}{\partial \pi_k}$,

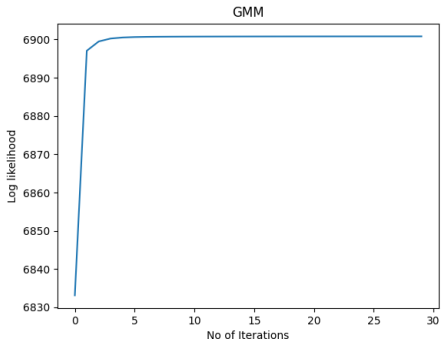
$$\boxed{\hat{\pi_k} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}}$$

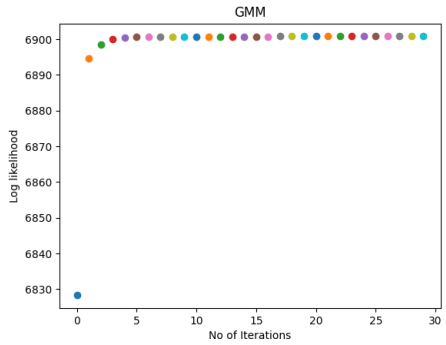
Estimate λ_k^i

$$\lambda_k^i = \frac{P(x_i, z_i = k; \theta)}{P(x_i; \theta)}$$

$$\boxed{\lambda_k^i = \frac{e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k}} \pi_k}{\sum_{l=1}^{k} e^{\frac{-(x_i - \mu_k)^2}{2\sigma_k}} \pi_k}}$$

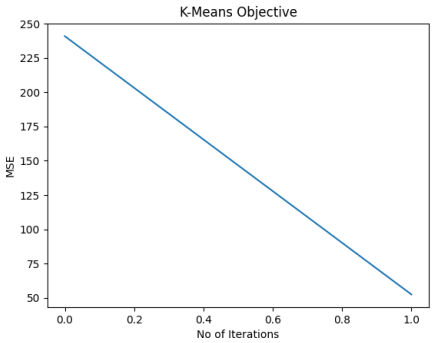
Gaussian Mixture Model:





(c) Run the K-means algorithm with K = 4 on the same data. Plot the objective of K means as a function of iterations.

Solution:
K - means as a function of iteration:



(d) Among the three different algorithms implemented above, which do you think you would choose to for this dataset and why?

Solution:
The data is best for K-Means. In K-means with any initialization Data is clustered in to two clusters. One of all 1's and other of 0's.
Rest 2 clusters are left empty with no data points in them.
Also the mean of Clusters is 0 1 0 0. with cluster 1 being Cluster of all 1's.

2. (5 points) You are given a data-set in the file A2Q2Data train.csv with 10000 points in (R^{100}, R) (Each row corresponds to a datapoint where the first 100 components are features and the last component is the associated y value).

   (a) Obtain the least squares solution W_{ML} to the regression problem using the analytical solution.

Solution: Least squares solution w_{ML} using analytical method is:

$$w_{ML} = (X^T X)^{-1} X^T Y$$

I have added a columns of $1s$ to my data as 101^{st} feature in order to bias it. This is done to normalise the dataset.
Here, $X = (500 \times 101)$ data points and $Y = (500 \times 1)$ output values.

   (b) Code the gradient descent algorithm with suitable step size to solve the least squares algorithms and plot $\|w^t - w_{ML}\|_2$ as a function of t. What do you observe?

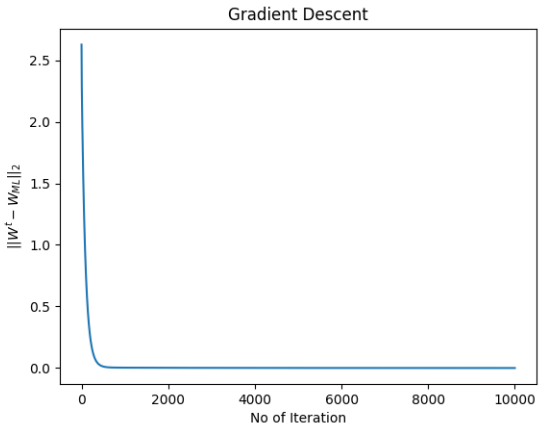
Solution: Gradient Descent algorithm is used to find the minima or slope of function of w. Consider we run a total of T iteration, and we are currently in the t^{th} iteration. Then:

$$w^{t+1} = w^t - \eta \ \nabla f(w^t)$$

The algorithm follows the above step until convergence.
In the above equation, η = Learning rate or step size. We have used $\eta = 0.000007$ in our algorithm.

Observations:
The value of learning rate used is optimal because it is the largest step value for which gradient descent algorithm does not grow exponentially. The algorithm converges after ~ 1000 iterations for $\eta = 0.000007$.
For values of learning rate which are too small, the algorithm takes a much longer time to reach convergence. Values such as $\eta = 0.0000071$ were tested and convergence was sub-optimal. Typically, the algorithm converged after ~ 10000 iterations for $\eta = 0.000007$.
For values of $\eta > 0.0001$, the algorithm behaves erratically and becomes exponential. For example, $\eta = 0.001$ produces an exponential graph.

The main takeaway from the gradient descent algorithm is that it is an ideal fit for datasets where number of data points is less.
we compute gradient step considering all n data points, and this is not feasible for large datasets with more than a million data points.

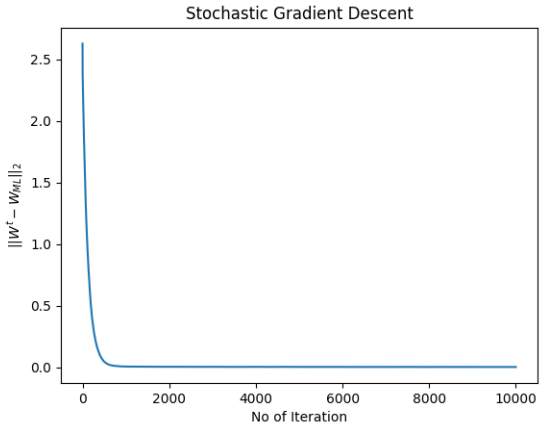


(c) Code the stochastic gradient descent algorithm using batch size of 100 and plot $\|w^t - w_{ML}\|_2$as a function of t. What are your observations? function of t. What are your observations?

Solution: Stochastic gradient descent differs from gradient descent in the sense that it uses only a small "bunch" of the original data points, to calculate step size for next iteration. This is called a batch of the data, and for each iteration we sample the same number of data points uniformly at random. Here size of batch $N = 100$.

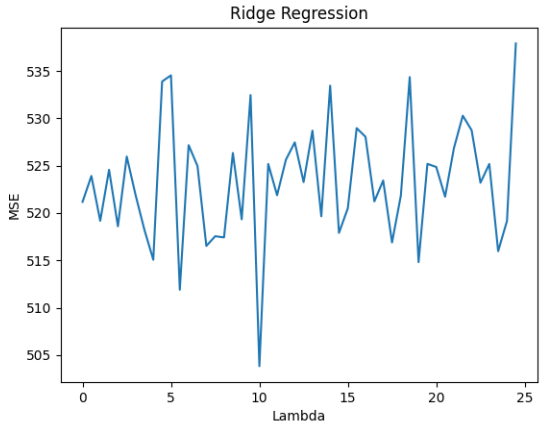
Observations:
Stochastic gradient descent is computationally much faster since it uses fewer data points per iteration to calculate step size. Here, we use a value of $N = 100$. The faster computation comes at the cost of convergence, since stochastic gradient descent takes longer to converge than gradient descent. For $\eta = 0.0001$, GD converges in ~ 1000 iterations while SGD takes ~ 13000 iterations to converge. Therefore, it is a computationally faster but convergently slower algorithm.
SGD behaves similar to GD algorithm with respect to η values. If η is too small, the algorithm will take longer to converge, whereas for if η is too large the algorithm may not converge at all.
Key takeaway for SGD algorithm is that it is a very efficient algorithm for very large datasets. For example, if number of data points n is in order of millions, gradient descent algorithm would be computationally very complex. However, SGD algorithm with batch size $N \sim$ *Hundreds or Thousands* would be the al-

gorithm of choice.



(d) Code the gradient descent algorithm for ridge regression. Cross-validate for various choices of λ and plot the error in the validation set as a function of λ. For the best λ chosen, obtain w_R. Compare the test error (for the test data in the file *A2Q2Data_test.csv*) of w_R with w_{ML}. Which is better and why?

Solution:



Test error for $w_{ML} = 185.3757511443717$.
Test error for $W_R = 182.69831723883442$.
It can be seen that W_R is a better estimator because it renders a lesser test error for dataset *A2Q2Data_test.csv*.
The reason for the better performance of Ridge Regression as opposed to MLE is that MLE is an unbiased estimator, Ridge Regression introduces a certain amount of bias in the estimator by levying a penalty. The hyperparameter λ allows us to tune the regression model to optimise it further.
Therefore, the Ridge Regression model is most efficient and optimal for data prediction.

3. (5 points) Consider estimating the mean of a uni-variate Gaussian random variable with mean μ and variance 1. Assume you have access to a single sample y from this distribution. What is the maximum likelihood estimator μ_{ML} for μ? Now consider shrinking your estimator μ_{ML} to a new estimator $\mu_{new} = \frac{\mu_{ML}}{2}$. Can this shrunk estimator have lesser MSE than the ML estimator? If yes, under what condition on μ can this happen? If no, prove the same.

Solution:
Sample $y \sim N(\mu, 1)$
PDF of Gaussian Distribution :

$$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

We derive Maximum likelihood $\mu_{\hat{ML}}$:

$$L(x_1, x_2, .., x_n, \mu) = \Pi_{i=1}^{n} \mathbf{f}(\mathbf{x_i}; \mu, \sigma)$$

$$L(x_1, x_2, .., x_n, \mu) = \Pi_{i=1}^{n} \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi}\sigma} \mathbf{e}^{\frac{-(\mathbf{x_i}-\mu)^\mathbf{2}}{\mathbf{2}\sigma^\mathbf{2}}}$$

Next, we note that log is a monotonically increase function, so we can maximize the log-likelihood:

$$log(L(x_1, x_2, .., x_n, \mu)) = \sum_{i=1}^{n} Log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(x_i - \mu)^2}{2\sigma^2}$$

We take derivatives of this with respect to μ and find:

$$\frac{\partial log(L(x_1, x_2, .., x_n, \mu))}{\partial \mu} = \sum_{i=1}^{n} -\frac{(x_i - \mu)}{\sigma^2} = 0$$

$$\sum_{i=1}^{n} \mu = \sum_{i=1}^{n} x_i$$

$$\mu\ n = \sum_{i=1}^{n} x_i$$

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

$$\hat{\mu}_{ML} = y$$

$E[\hat{\mu}_{ML}] = E[y] = \mu$ ....(1)

$Var(\hat{\mu}_{ML}) = Var(y) = 1....(2)$
Shrinking $\hat{\mu}_{ML}$ to $\frac{\hat{\mu}_{ML}}{2}$

$$\hat{\mu}_{new} = \frac{\hat{\mu}_{ML}}{2}$$

Mean Square Error :
$\hat{\mu}_{ML}$:

$$MSE = E[(\hat{\theta} - E[\theta])^2] + (E[\hat{\theta}] - \theta)^2$$

$$MSE_{ML} = Var(\hat{\mu}_{ML}) + (Bias)^2$$

$$MSE_{ML} = Var(\hat{\mu}_{ML}) + (E[\hat{\mu}_{ML}] - \mu)^2$$

$$MSE_{ML} = Var(y) + (E[y] - \mu)^2$$

$$MSE_{ML} = 1 + 0 \quad \ldots from\ (1)\ and\ (2)$$

$$MSE_{ML} = 1$$

As $\hat{\mu}_{ML}$ is unbiased.
$\hat{\mu}_{new}$:

$$MSE_{new} = Var(\hat{\mu}_{new}) + (Bias)^2$$

$$MSE_{new} = Var(\frac{\hat{\mu}_{ML}}{2}) + (E[\frac{\hat{\mu}_{ML}}{2}] - \mu)^2$$

$$MSE_{new} = \frac{1}{4}Var(\hat{\mu}_{ML}) + (\frac{1}{2}E[\hat{\mu}_{ML}] - \mu)^2$$

$$MSE_{new} = \frac{1}{4} + (\frac{\mu}{2} - \mu)^2$$

$$MSE_{new} = \frac{1}{4} + \frac{\mu^2}{4} \quad \ldots from\ (1)\ and\ (2)$$

$$MSE_{new} = \frac{\mu^2 + 1}{4}$$

To have $MSE_{new} < MSE_{ML}$

$$\frac{\mu^2 + 1}{4} < 1$$

$$\mu^2 < 3$$

$$\therefore \mu \in [-\sqrt{3}, \sqrt{3}]$$

4. (5 points) Play Question: In the above question, given a interval [a, b] where $a < b$, can you construct an estimator whose MSE is lower in the given interval than the MSE of the maximum likelihood estimator in the same interval? Can you prove why your estimator achieves this property?

Solution:
Sample $y \sim N(\mu, 1)$
PDF of Gaussian Distribution :

$$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

We derive Maximum likelihood $\hat{\mu_{ML}}$:

$$L(x_1, x_2, .., x_n, \mu) = \Pi_{i=1}^{n} \mathbf{f}(\mathbf{x_i}; \mu, \sigma)$$

$$L(x_1, x_2, .., x_n, \mu) = \Pi_{i=1}^{n} \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi}\sigma} \mathbf{e}^{\frac{-(\mathbf{x_i}-\mu)^\mathbf{2}}{\mathbf{2}\sigma^\mathbf{2}}}$$

Next, we note that log is a monotonically increase function, so we can maximize the log-likelihood:

$$log(L(x_1, x_2, .., x_n, \mu)) = \sum_{i=1}^{n} Log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(x_i - \mu)^2}{2\sigma^2}$$

We take derivatives of this with respect to μ and find:

$$\frac{\partial log(L(x_1, x_2, .., x_n, \mu))}{\partial \mu} = \sum_{i=1}^{n} -\frac{(x_i - \mu)}{\sigma^2} = 0$$

$$\sum_{i=1}^{n} \mu = \sum_{i=1}^{n} x_i$$

$$\mu\ n = \sum_{i=1}^{n} x_i$$

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

$$\hat{\mu}_{ML} = y$$

$E[\hat{\mu}_{ML}] = E[y] = \mu$ ....(1)
$Var(\hat{\mu}_{ML}) = Var(y) = 1$....(2)

Mean Square Error :
$\hat{\mu}_{ML}$:

$$MSE = E[(\hat{\theta} - E[\theta])^2] + (E[\hat{\theta}] - \theta)^2$$

$$MSE_{ML} = Var(\hat{\mu}_{ML}) + (Bias)^2$$

$$MSE_{ML} = Var(\hat{\mu}_{ML}) + (E[\hat{\mu}_{ML}] - \mu)^2$$

$$MSE_{ML} = Var(y) + (E[y] - \mu)^2$$

$$MSE_{ML} = 1 + 0 \quad ... \; from \; (1) \; and \; (2)$$

$$MSE_{ML} = 1$$

Let's consider our New Estimator:

$$\hat{\mu}_{new} = \frac{\hat{\mu}_{ML}}{K}$$

shrinking our $\hat{\mu}_{ML}$ by a constant factor of K.
$\hat{\mu}_{new}$:

$$MSE_{\hat{\mu}_{new}} = Var(\hat{\mu}_{new}) + (E[\hat{\mu}_{new}] - \mu)^2$$

$$MSE_{\hat{\mu}_{new}} = Var(\frac{\hat{\mu}_{ML}}{K}) + (E[\frac{\hat{\mu}_{ML}}{K}] - \mu)^2$$

$$MSE_{new} = \frac{1}{K^2} Var(\hat{\mu}_{ML}) + (\frac{1}{K} E[\hat{\mu}_{ML}] - \mu)^2$$

$$MSE_{new} = \frac{1}{K^2} * 1 + (\frac{1}{K} * \mu - \mu)^2 \quad .. \; from \; (1) \; and \; (2)$$

$$MSE_{new} = \frac{1}{K^2} * 1 + (\frac{\mu - \mu K}{K})^2$$

$$MSE_{new} = \frac{1 + (\mu - K\mu)^2}{K^2}$$

MSE of this estimator to give lesser value than our Maximum likelihood estimator.

$$MSE_{new} < MSE_{ML}$$

$$\frac{1 + (\mu - K\mu)^2}{K^2} < 1$$

$$\mu^2 (1 - K)^2 < K^2 - 1$$

$$\mu^2 < \frac{K^2 - 1}{(1 - K)^2}$$

$$\mu^2 < \frac{(K - 1)(K + 1)}{(1 - K)(1 - K)}$$

$$\mu^2 < \frac{K + 1}{K - 1}$$

$$\mu \in \pm \sqrt{\frac{K + 1}{K - 1}}$$

Considering interval [a, b]

$$\sqrt{\frac{K + 1}{K - 1}} = |a| + |b| \quad ... \; (3)$$

and

$$-\sqrt{\frac{K+1}{K-1}} = -(|a| + |b|) \quad ... (4)$$

In (3),

$$\sqrt{\frac{K+1}{K-1}} = |a| + |b| \quad ... (3)$$

Taking Squaring both the side.

$$\frac{K+1}{K-1} = (|a| + |b|)^2 \quad ... (3)$$

solving Equation (3) for K

$$K = -\frac{|a|^2 + 2|a||b| + |b|^2 + 1}{-|a|^2 - 2|a||b| - |b|^2 + 1}$$

Our New Estimator will have MSE lower than the MSE of the maximum likelihood estimator in the [a, b] interval when K is $-\frac{|a|^2+2|a||b|+|b|^2+1}{-|a|^2-2|a||b|-|b|^2+1}$.

If K gets undefined when either $a = 0$ or $b = 0$ our function then becomes:
Let $x^2 = |a|^2 + 2|a||b| + |b|^2$ and $K = \frac{x^2+1}{-x^2+1}$ then we use the function $K = \frac{x^2+2}{-x^2+2}$

For Example:

$$\mu \in [-\sqrt{\frac{K+1}{K-1}}, \sqrt{\frac{K+1}{K-1}}]$$

Suppose our interval [a,b] is [-5,7]

$$\mu \in [-\sqrt{\frac{K+1}{K-1}}, \sqrt{\frac{K+1}{K-1}}] = [-(|a| + |b|), (|a| + |b|) \,]$$

$$\mu \in [-12, 12]$$

which is the superset of our interval [-5,7],we can say that MSE of $\hat{\mu}_{new}$ will be lower than MSE of $\hat{\mu}_{ML}$ in this range when K is $-\frac{|a|^2+2|a||b|+|b|^2+1}{-|a|^2-2|a||b|-|b|^2+1}$.