UNIVERSITY OF SOUTHERN CALIFORNIA, COMPUTER SCIENCE

Machine Learning: HomeWork 4

Rohit Kondekar
740-581-9473

November 17, 2014

1 Question: Boosting

1.1 Gradient Calculation

$$\textbf{Least Squared Loss:} \quad L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

$$\textbf{Gradient of Loss:} \quad g_i = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}$$

$$g_i = -2(y_i - \hat{y}_i)$$

1.2 Weak Learner Selection

$$\gamma^* = \texttt{argmin}_{\gamma \in R} \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2$$

$$= \texttt{argmin}_{\gamma \in R} \sum_{i=1}^{n} (2(y_i - \hat{y}_i) - \gamma h(x_i))^2$$

Differentiating w.r.t γ and equating to zero

$$\sum_{i=1}^{n} -2h(x_i)[2(y_i - \hat{y}_i) - \gamma h(x_i)] = 0$$

$$\sum_{i=1}^{n} [-4h(x_i)(y_i - \hat{y}_i) + 2\gamma h(x_i)^2] = 0$$

$$2\gamma \sum_{i=1}^{n} h(x_i)^2 = 4 \sum_{i=1}^{n} h(x_i)(y_i - \hat{y}_i)$$

$$\gamma = \frac{2\sum_{i=1}^{n} h(x_i)(y_i - \hat{y}_i)}{\sum_{i=1}^{n} h(x_i)^2}$$

1.3 Step Size Selection

$$\alpha^* = \texttt{argmin}_{\alpha \in R} \sum_{i=1}^{n} (y_i - (\hat{y}_i + \alpha h^*(x_i)))^2$$

Differentiating w.r.t α and equating to zero

$$\sum_{i=1}^{n} -2h^*(x_i)(y_i - \hat{y}_i - \alpha h^*(x_i)) = 0$$

$$\sum_{i=1}^{n} [-2h^*(x_i)y_i + 2h^*(x_i)\hat{y}_i + 2\alpha h^*(x_i)^2] = 0$$

$$2\sum_{i=1}^{n} \alpha h^*(x_i)^2 = 2\sum_{i=1}^{n} [h^*(x_i)(y_i - \hat{y}_i)]$$

$$\alpha = \frac{\sum_{i=1}^{n} h^*(x_i)(y_i - \hat{y}_i)}{\sum_{i=1}^{n} h^*(x_i)^2}$$

2 Question: Neural Network

2.1

As the question says the network has logistic output and linear activation function in hidden layer, the prediction function can be written as, where w is weights of second layer and v are weights of first layer:

$$\begin{aligned} y &= \sigma(\sum_k w_k \times (\sum_i v_{ki} x_i)) \\ &= \frac{1}{1 + \exp(-\sum_k w_k \times (\sum_i v_{ki} x_i))} \\ &= \frac{1}{1 + \exp(-\sum_k \sum_i w_k v_{ki} x_i)} \\ &= \frac{1}{1 + \exp(-\sum_i w_i' x_i)} \quad \text{Where } w_i' = \sum_k w_k v_{ki} \end{aligned}$$

This shows thats its similar to logistic regression, with different weights combination. Even if there are multiple hidden layers, the equation would be in similar format.

2.2

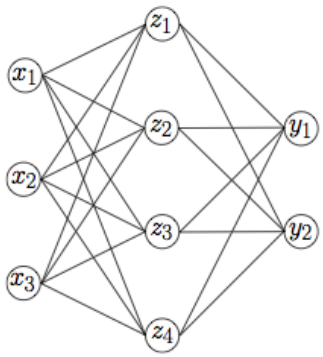

Figure 2.1: Neural Network

$$L(y, \hat{y}) = \frac{1}{2}\sum_{j=1}^{2} (y_j - \hat{y}_j)^2$$

First finding gradient for v_{jk}:

$$\frac{\partial L}{\partial v_{jk}} = \frac{\partial L}{\partial b_j}\frac{\partial b_j}{\partial v_{jk}} \quad \text{where } b_j = \hat{y}_j = \sum_k v_{jk} z_k$$

$$\frac{\partial L}{\partial v_{jk}} = -(y_j - \hat{y}_j) z_k$$

Now finding gradient for w_{ki}:

$$\begin{aligned}
\frac{\partial L}{\partial w_{ki}} &= \frac{\partial L}{\partial a_k}\frac{\partial a_k}{\partial w_{ki}} \quad \text{where } a_k = \sum_k w_{ki} x_i \\
&= \frac{\partial L}{\partial a_k} x_i \\
&= \sum_j \frac{\partial L}{\partial b_j}\frac{\partial b_j}{\partial a_k} x_i \\
&= \sum_j (y_j - \hat{y}_j) \frac{\partial (v_{jk} tanh(a_k))}{\partial a_k} x_i \\
&= \sum_j^2 (y_j - \hat{y}_j) v_{jk} (1 - tanh^2(\sum_i^3 w_{ki} x_i)) x_i
\end{aligned}$$

3 Question: Clustering

3.1

$$\begin{aligned}
D &= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||_2^2 \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} (x_n - \mu_k)^T (x_n - \mu_k) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} (x_n^T x_n - x_n^T \mu_k - \mu_k^T x_n + \mu_k^T \mu_k) \\
\frac{\partial D}{\partial \mu_k} &= \sum_{n=1}^{N} r_{nk} (2\mu_k - 2x_n) = 0 \\
\sum_{n=1}^{N} r_{nk} \mu_k &= \sum_{n=1}^{N} r_{nk} x_n \\
\mu_k &= \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}
\end{aligned}$$

This shows that to minimize this loss/cost function, μ has to be the mean of the respective clusters.

3.2

$$D = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||_1$$

Intutively, we can see here that the cost function is bascically sum of the difference between μ and each data point.
If we sort the array and choose the last and first point (i.e. the points which are farthest), the point which is closest to all other points, will the point which lies in center. i.e. equidistant from all the points.

So if n is odd the the point will be at $\frac{i+j}{2}$th location. else if even it would be the the average of two middle elements.

It can be shown as:
If we have an even number of points, there will be two medians. A point between the two medians will have n/2 points to the left and n/2 points to the right, and a total sum-of-distances to those points of S.

If we move it one point to the left, S will go up by n/2 (since we're moving away from the right-most points) and down by n/2 (since we're moving towards the left-most points), so overall S remains the same. This holds true until we hit the left-most median point. When we move one left of the left-most median point, we now have (n/2 + 1) points to the right, and (n/2 - 1) points to the left, so S goes up by two. Continuing to the left will only increase S further.

By the same logic, all points to the right of the right-most median also have a higher S.

If we have an odd number of points, there is only one median. Using the same logic as above, we can show that it has the lowest value of S.

Mathematically:

$$\frac{\partial D}{\partial \mu_k} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} sign(x_n - \mu_k) = 0$$

Calculating for a perticular cluster of size m:

$$\sum_{m=1}^{M} sign(x_m - \mu_k) = 0$$
$$sign(x_m - \mu_k) = +1 \quad if\ x_m - \mu_k > 0$$
$$= -1 \quad if\ x_m - \mu_k < 0$$

Therefore: $\Psi(x_n | x_n - \mu_k > 0) - \Psi(x_n | x_n - \mu_k < 0) = 0$ where Ψ denotes number of elements.

This becomes zero pricisely at median.

4 Question: Mixture Models

4.1

$X_i \sim \exp(\lambda)$
$Y_i = min(X_i, c_i)$

Therefore, in this problem the observed variables are $Y_i = X_i : 0 < i <= r$
The hidden variables are $Y_i = X_i : r + 1 <= i <= n$
And the parameter is λ

$$\textbf{Likelihood:} \quad L(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i}$$
$$= \lambda^n e^{-\lambda \sum_i^n X_i}$$
$$\textbf{Log Likelihood:} \quad l(\lambda) = n \log(\lambda) - \lambda \sum_{i=1}^{n} X_i$$
$$= n \log(\lambda) - \lambda [\sum_{i=1}^{r} X_i + \sum_{j=r+1}^{n} X_j]$$

4.2

E Step is where we take the Expectation of Log Likelihood. Assume $Z = X_{r+1}.....X_n$ i.e. the unoberved data.

$$\begin{aligned}
E_{Z|Y}[l(\lambda)] &= E_{Z|Y}[n\log(\lambda) - \lambda[\sum_{i=1}^{r} X_i + \sum_{j=r+1}^{n} X_j]] \\
&= n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - E_{Z|Y}[\sum_{j=r+1}^{n} X_j] \\
&= n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} E_{Z|Y}[X_j] \\
&= n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} E_{X_j|Y_j}[X_j] \\
&= n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} E_{X_j|Y_j}[X_j > c_j] \\
&= n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} (c_j + \frac{1}{\lambda^k}) \quad \textbf{Because of memoryless property of exponential distribution.}
\end{aligned}$$

$$Q(\lambda, \lambda^k) = E_{Z|Y}[l(\lambda)] = n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} (c_j + \frac{1}{\lambda^k})$$

4.3

In M-step we will maximize the $Q(\lambda, \lambda^k)$ w.r.t λ i.e.

$$\lambda^{k+1} = argmax_\lambda \; n\log(\lambda) - \lambda\sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} (c_j + \frac{1}{\lambda^k})$$

Differentiating it w.r.t λ we get:

$$\begin{aligned}
\frac{n}{\lambda} - \sum_{i=1}^{r} X_i - \sum_{j=r+1}^{n} c_j - \frac{n-r}{\lambda^k} &= 0 \\
\lambda &= \frac{n}{\sum_{i=1}^{r} X_i + \sum_{j=r+1}^{n} c_j + \frac{n-r}{\lambda^k}} \\
\lambda^{k+1} &= \frac{n}{\sum_{i=1}^{r} X_i + \sum_{j=r+1}^{n} c_j + \frac{n-r}{\lambda^k}}
\end{aligned}$$

We will have to do this iteratively till it converges, i.e. till this condition is not satisfied $\lambda^{k+1} = \lambda^k$

5 Question: Programming

5.1

5.2

5.2.1

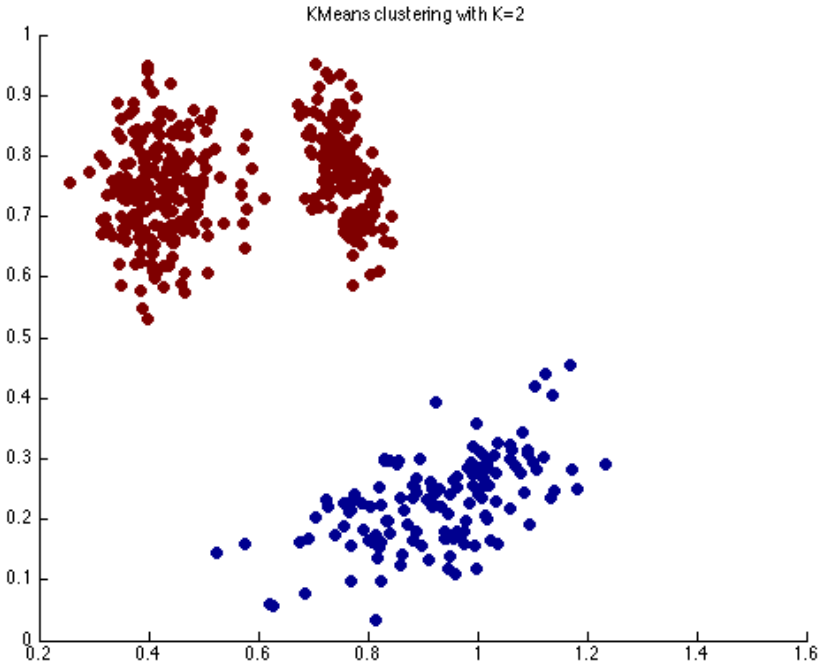

Figure 5.1: KMeans Clustering with K=2

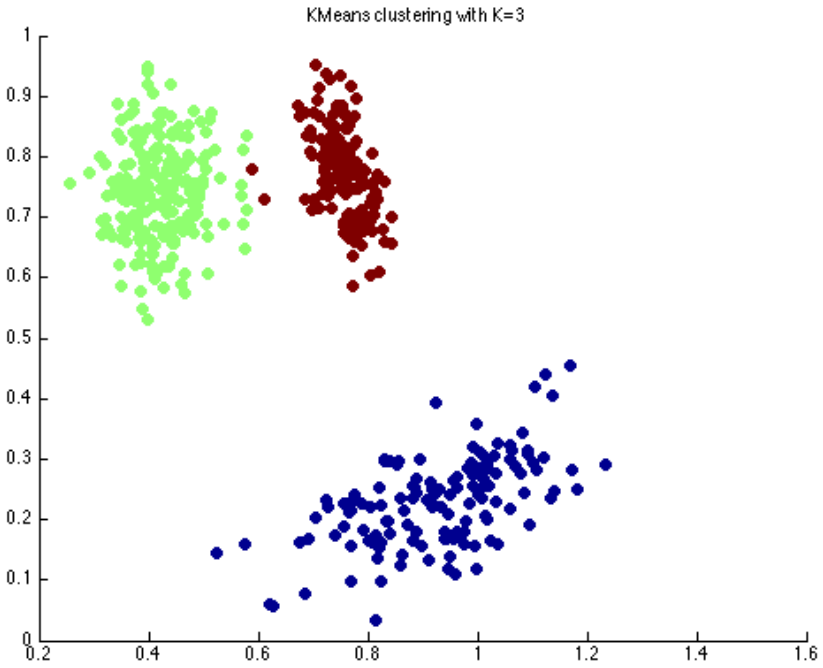

Figure 5.2: KMeans Clustering with K=3

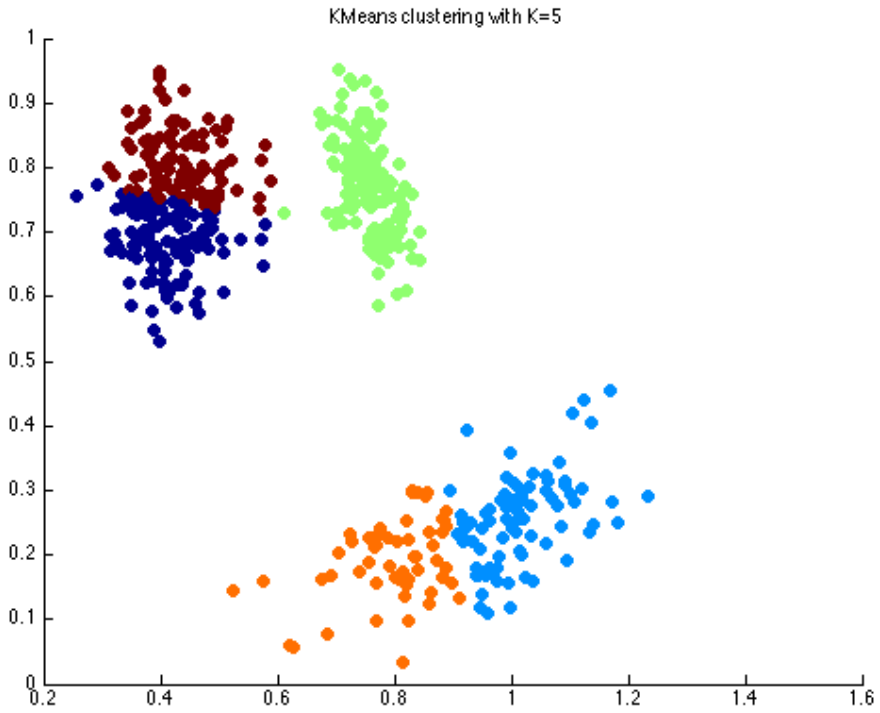

Figure 5.3: KMeans Clustering with K=5

5.2.2

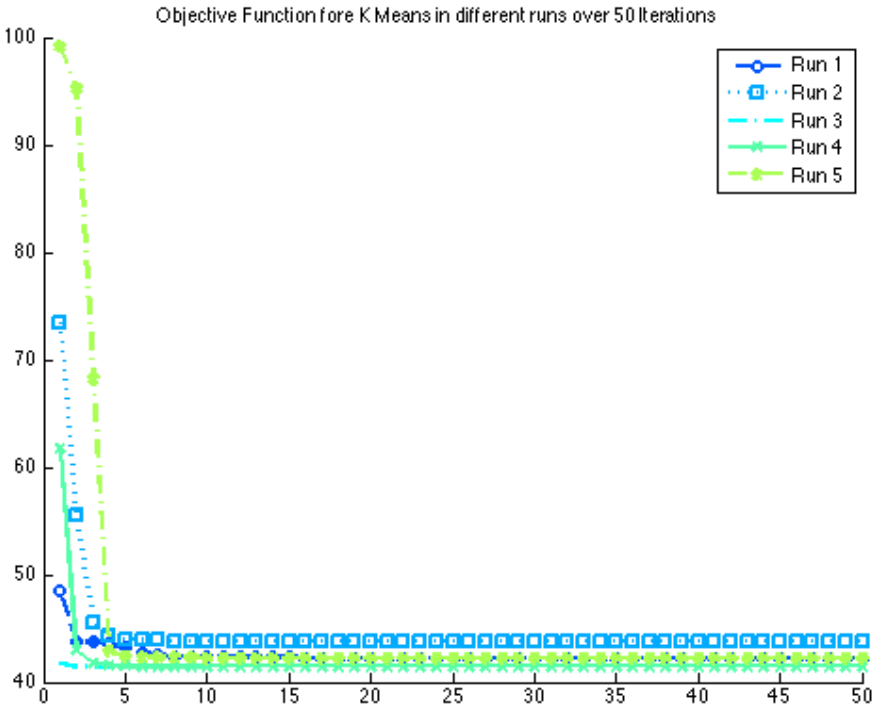

Figure 5.4: Cost Function for KMeans for random initializations and K=4

5.2.3

K means always converge because its objective function is the distance between points in the space. As there are only finite number of assignments for μ which is calculated from the given data points.

Though it may happen that, it may converge to global optimum rather than local optimum and in few cases may take exponential amount of time to converge. But the convergence is gauranteed.

In our data it can be seen that most of them converge within 5 iterations itself.

5.3

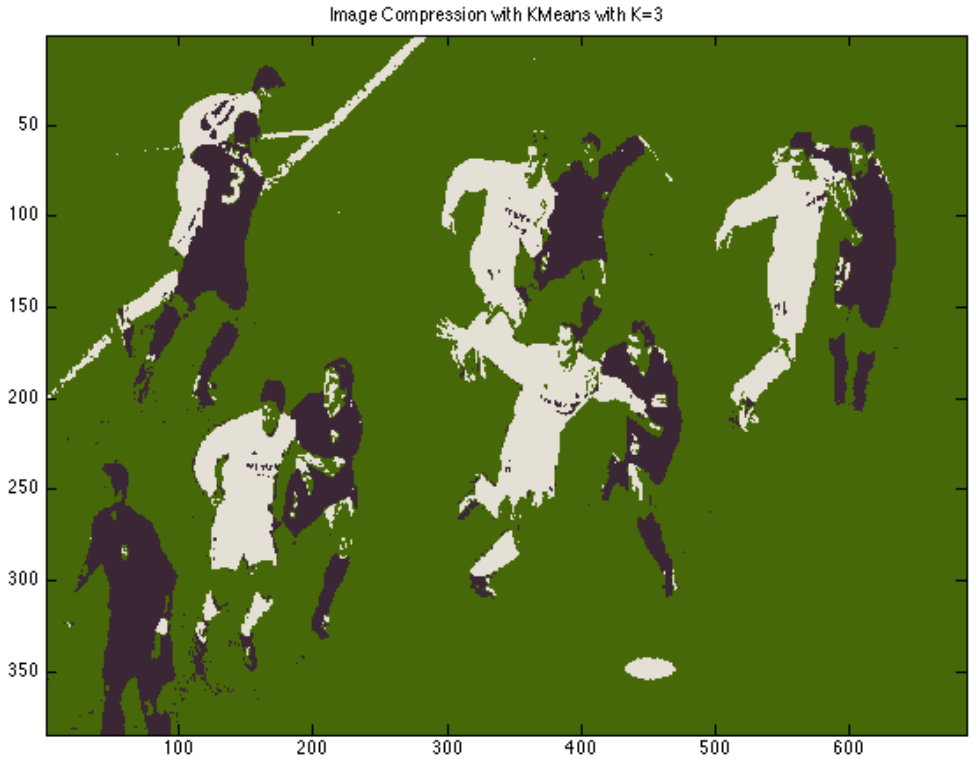

Figure 5.5: Reconstructed Image for K=3

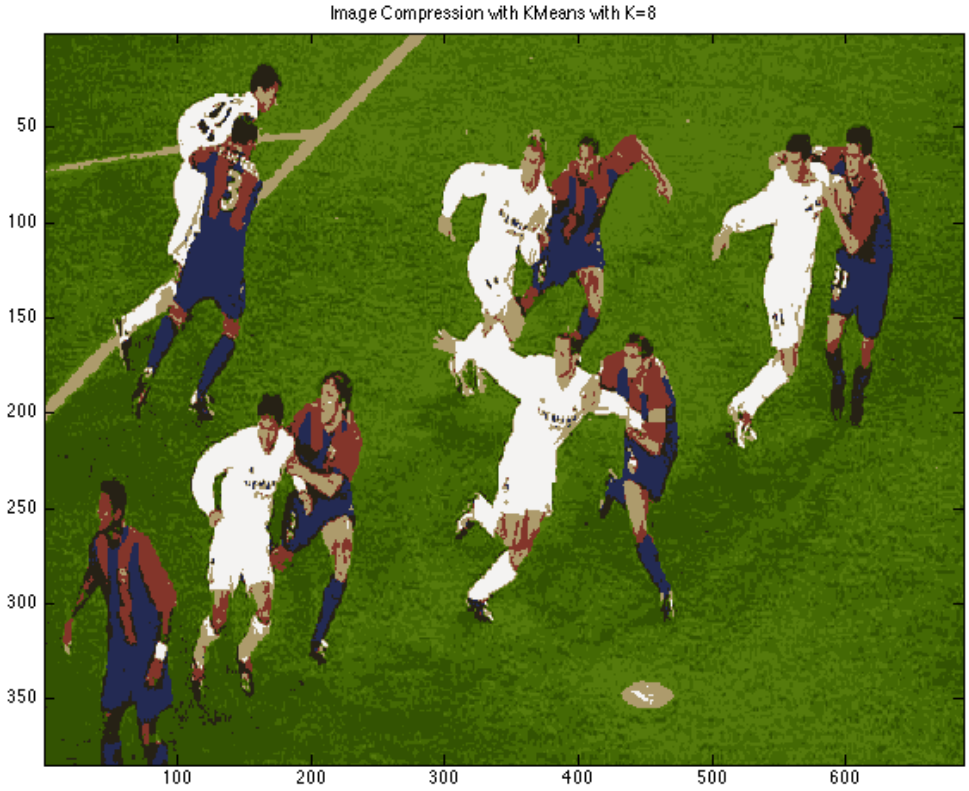

Figure 5.6: Reconstructed Image for K=8

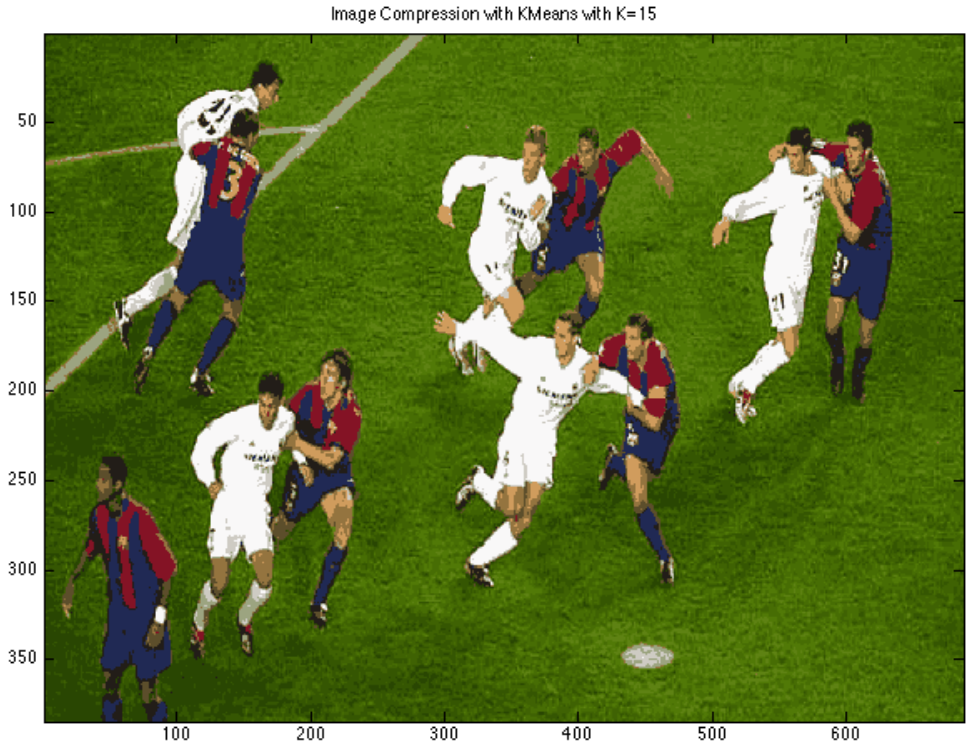

Figure 5.7: Reconstructed Image for K=15