UNIVERSITY OF SOUTHERN CALIFORNIA

Homework 4

Peehoo Dewan
Collaberated with Coco Chengliangdong, Pooja Voladoddi, Satakshi Rana
November 19, 2014

1 BOOSTING

Given,

$$\mathcal{H} = \{h_j, j = 1,2,3 \ldots M\}$$
$$y_i \in -1, 1$$
$$x_i \in \mathbb{R}^d, \text{ for i = 1,2, } \ldots \text{n}$$
$$\text{Loss function, } \; L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

1.1 GRADIENT CALCULATION

Gradient g_i is given as,

$$g_i = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}$$

Therefore differentiating the loss function w.r.t $\hat{y}_i$ gives,

$$g_i = 2(y_i - \hat{y}_i)(-1)$$

$$g_i = 2(\hat{y}_i - y_i) \tag{1.1}$$

1.2 Weak Learner Selection

$$h^* = \arg\min_{h \in \mathcal{H}} \left(\min_{\gamma \in \mathbb{R}} \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2 \right)$$

Looking at the inner bracket and differentiating w.r.t γ and setting it to zero gives,

$$\frac{\partial \left(\sum_{i=1}^{n} (-g_i - \gamma h(x_i)) \right)}{\partial \gamma} = \sum_{i=1}^{n} (2)(-g_i - \gamma h(x_i))(-h(x_i)) := 0$$

$$\sum_{i=1}^{n} g_i h(x_i) = \sum_{i=1}^{n} -\gamma h(x_i)^2$$

$$\gamma^* = \frac{\sum_{i=1}^{n} g_i h(x_i)}{\sum_{i=1}^{n} -h(x_i)^2} \tag{1.2}$$

This proves that the value of γ can be derived in closed form and therefore h^* can be derived independent of γ

1.3 Step Size Selection

$$\begin{aligned} \alpha^* &= \arg\min_{\alpha \in \mathbb{R}} \sum_{i=1}^{n} (L(y_i, \hat{y}_i + \alpha h^*(x_i)) \\ &= \arg\min_{\alpha \in \mathbb{R}} \sum_{i=1}^{n} (y_i - (\hat{y}_i + \alpha h^*(x_i)))^2 \end{aligned}$$

Differentiating w.r.t α and setting to zero gives,

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

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

$$\alpha^* = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)(h^*(x_i))}{\sum_{i=1}^{n} (h^*(x_i))^2} \tag{1.3}$$

Substituting we get the update equation as,

$$\hat{y}_i \leftarrow \hat{y}_i + \left(\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)(h^*(x_i))}{\sum_{i=1}^{n} (h^*(x_i))^2} \right) h^*(x_i) \tag{1.4}$$

2 Neural Network

2.1 Linear activation functions in the hidden layers

Figure 2.1 shows a neural network with linear activation functions in multiple hidden layers and single logistic output. The figure might be in the end.

Description : x1, x2, x3 are inputs to the neural network. This neural network consisits of 2 hidden layers having linear activation function. Common terminology is as below:

z_i^j - represents the i^{th} intermediate term in layer j

a_i^j - represents the i^{th} activation term in layer j

$a_i^j = h(z_i^j)$,where h is the activation function

w_{id}^j - represents the weight connecting d^{th} input of layer j to i^{th} neuron of layer $j+1$

Without loss of generality,

$$z_i^j = \sum_{d=1}^{D} w_{id}^{j-1} x_d$$

Similarly we can calculate intermediate terms for each i where i varies from 1 to the number of units in the hidden layer. In our case we have i=4 and j=2 for the above equation. Since we have linear activation function for the hidden layer we get that,

$$a_i^j = h(z_i^j) = z_i^j = \sum_{d=1}^{D} w_{id}^{j-1} x_d$$

Moving to the next hidden layer, k=4(where k is the number of units in this hidden layer) and j=3. The activation functions that we derived above now serve as input to this layer. We can again write the new intermediate terms and the activations as, Without loss of generality,

$$z_k^j = \sum_{i=1}^{4} w_{ki}^{j-1} a_i^{j-1}$$

Since we have linear activation function for the hidden layer we get that,

$$a_k^j = h(z_k^j) = z_k^j = \sum_{i=1}^{4} w_{ki}^{j-1} a_i^{j-1}$$

where j=3, k=4 We can keep repeating this again and again and we would see that after doing linear transformation, it still remains a linear combination. Since we finally have a logistic output unit so we can write that,

$$y = g(\sum_{k=1}^{4} w_{1k}^{j-1} a_k^{j-1})$$

for j=4 and g = sigmoid function. This way we have showed that this neural network is equal to the logistic regression.

2.2 BACKPROPAGATION

From the model which is given to us, we see that,

$$z_k = tanh(\sum_{i=1}^{3} w_{ki} x_i) \text{ for k=1,2,3,4}$$

$$a_k = \sum_{i=1}^{3} w_{ki} x_i$$

$$y_j = \sum_{k=1}^{4} v_{jk} z_k \text{ for j=1,2}$$

$$b_j = \sum_{k=1}^{4} v_{jk} z_k$$

$$L(y, \hat{y}) = \frac{1}{2}((y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2)$$

$$x_i \xrightarrow{w} a_i \xrightarrow{g} z_i \xrightarrow{v} bi \xrightarrow{h} \hat{y}_i$$

Let us consider θ = (w,v) where w and v are first and second layer weights

$$L(\theta) = \frac{1}{2}\sum_{n}((y_{n1} - \hat{y}_{n1}(\theta))^2 + (y_{n2} - \hat{y}_{n2}(\theta))^2)$$

In order to find the backpropagation updates, we have to take the derivative of $L(\theta)$ w.r.t θ We will derive the gradient for each input n (without loss of generality) and then sum over all inputs to obtain the overall gradient.

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{2}((y_{ij} - \hat{y}_{ij}(\theta))^2$$

$$\nabla_{v_j} L_i = \frac{\partial L_i}{\partial b_{ij}} . \frac{\partial b_{ij}}{\partial v_j}$$
$$= \frac{\partial L_i}{\partial b_{ij}} z_i$$

We get the above because,

$$b_{ij} = \mathbf{v}_j^T \mathbf{z}_i$$

$$\frac{\partial L_i}{\partial b_{ij}} = \delta_{ij}^v = y_{ij} - \hat{y}_{ij} = \text{ error in the output layer}$$

Therefore,

$$\nabla_{v_j} L_i = \delta_{ij}^v z_i$$

We propagate this error to the input layers. Therefore we see,

$$\begin{aligned}\nabla_{w_k} L_i &= \frac{\partial L_i}{\partial a_{ik}} . \frac{\partial a_{ik}}{\partial w_k} \\ &= \frac{\partial L_i}{\partial a_{ik}} x_i\end{aligned}$$

We get the above because,

$$a_{ik} = \mathbf{w}_k^T \mathbf{x}_i$$

$$\begin{aligned}\frac{\partial L_i}{\partial a_{ik}} &= \delta_{ik}^w \\ &= \sum_{j=1}^{2} \frac{\partial L_i}{\partial b_{ij}} . \frac{\partial b_{ij}}{\partial a_{ik}} \\ &= \sum_{j=1}^{2} \delta_{ij}^v \frac{\partial b_{ij}}{\partial a_{ik}}\end{aligned}$$

Since,

$$b_{ij} = \sum_{k=1}^{4} v_{jk} g(a_{ik})$$

Therefore,

$$\frac{\partial b_{ij}}{\partial a_{ik}} = v_{jk} g'(a_{ik})$$

As,

$$\begin{aligned} g(a) &= tanh(a) \\ g'(a) = 1 - tanh^2(a) &= sech^2(a)\end{aligned}$$

$$\begin{aligned}\delta_{ik}^w &= \sum_{j=1}^{2} \delta_{ij}^v v_{jk} g'(a_{ik}) \\ &= \sum_{j=1}^{2} \delta_{ij}^v v_{jk} sech^2 a_{ik}\end{aligned}$$

Let η_1 and η_2 be the step sizes, therefore we can write the update equation for weights as,

$$\begin{aligned} v_{jk}^{t+1} &= v_{jk}^t - \eta_1 (\hat{y}_j - y_j) z_k \\ w_{ki}^{t+1} = w_{ki}^t - \eta_2 x_i sech^2 (\sum_{i=1}^{3} w_{ki} x_i) &\sum_{j=1}^{2} (\hat{y}_j - y_j) v_{jk}\end{aligned}$$

3 CLUSTERING

3.1 DERIVATION OF MEAN WITH L2 NORM

Distortion measure,

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||_2^2$$

Without loss of generality, differentiating w.r.t cluster k and solving for μ_k, we get

$$\frac{\partial D}{\partial \mu_k} = \sum_{n=1}^{N} (-2) r_{nk} (x_n - \mu_k) := 0$$
$$\sum_{n=1}^{N} r_{nk} x_n = \sum_{n=1}^{N} r_{nk} \mu_k$$

Therefore we get μ_k as,

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}$$

3.2 DISTORTION MEASURE WITH L1 NORM

Distortion measure,

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

Without loss of generality, differentiating w.r.t cluster k and solving for μ_k, we get

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

We know that absolute value function or mod function is not differentiable when it is zero, so we replace it by the sign function,

$$\frac{\partial D}{\partial \mu_k} = \sum_{n=1}^{N} sign(x_n - \mu_k) := 0$$
$$\sum_{n=1}^{N} r_{nk} (I_{\mu_k > x_n}) = \sum_{n=1}^{N} r_{nk} (I_{\mu_k < x_n})$$

Therefore we see that the number of points whose value is greater than μ_k is equal to the number of points whose value is less than μ_k. Therefore μ_k is the median of the k-th cluster.

4 MIXTURE MODELS

4.1 LIKELIHOOD

Given,

$$f_X(x,\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & otherwise \end{cases}$$

Likelihood can be written as,

$$\begin{aligned} L(\lambda|x_1, x_2, \dots x_n) &= \prod_{n=1}^{n} P(X_i) \\ &= \lambda^n e^{-\lambda(\sum_{i=1}^{n} x_i)} \end{aligned}$$

Taking log, we get log likelihood of complete set as,

$$logL(\lambda|x_1, x_2, \dots x_n) = nlog\lambda - \lambda \sum_{i=1}^{n} x_i$$

4.2 EXPECTATION

In this step we write the auxillary function Q as,

$$Q = E[logL(\lambda|x_1, x_2, \dots x_n)]$$

Substituting the value of likelihood we get,

$$Q = E[nlog\lambda - \lambda \sum_{i=1}^{n} x_i]$$

Here we find that the first term is constant and so its expected value is the term itself. Next we break up the n examples into observed and unobserved

$$Q(\lambda, \lambda^{old}) = nlog\lambda - \lambda \sum_{i=1}^{r} x_i - E[\lambda \sum_{i=r+1}^{n} x_i] \tag{4.1}$$

Now we compute the conditional expectation $E[\lambda \sum_{i=r+1}^{n} x_i]$

$$E[\lambda \sum_{i=r+1}^{n} x_i] = E[x_i|x_i > c_i] = \int_{c_i}^{\infty} x_i f_X(x_i|x_i > c_i) dx$$

Applying Bayes rule,

$$f_X(x_i|x_i > c_i) = \frac{f_X(x_i, x_i > c_i)}{f_X(x_i > c_i)} \tag{4.2}$$

$$f_X(x_i > c_i) = 1 - f_X(x_i < c_i) \tag{4.3}$$

We know that,

$$\begin{aligned} f_X(x_i < c_i) = F_X(c_i) &= \int_0^{c_i} f_X(x_i)\,dx_i \\ &= \frac{\lambda e^{-\lambda x_i}|_0^{c_i}}{(-\lambda)} \\ &= 1 - e^{-\lambda c_i} \end{aligned}$$

Substituting in equation 4.3 we get,

$$\begin{aligned} f_X(x_i > c_i) &= 1 - (1 - e^{-\lambda c_i}) \\ &= e^{-\lambda c_i} \end{aligned} \tag{4.4}$$

Substituting in equation 4.2 we get,

$$\begin{aligned} f_X(x_i | x_i > c_i) &= \frac{f_X(x_i, x_i > c_i)}{e^{-\lambda c_i}} \\ &= \int_{c_i}^{\infty} \frac{x_i \lambda e^{-\lambda x_i}}{e^{-\lambda c_i}} dx_i \end{aligned} \tag{4.5}$$

Integrating by parts we get,

$$\begin{aligned} &\int_{c_i}^{\infty} \frac{x_i \lambda e^{-\lambda x_i}}{e^{-\lambda c_i}} dx_i \\ &\qquad = \frac{\lambda}{e^{-\lambda c_i}} \left[\frac{-1 x_i e^{-\lambda x_i}}{\lambda} \right]_{c_i}^{\infty} \\ &\qquad = c_i + \frac{1}{\lambda} \end{aligned} \tag{4.6}$$

Substituting in the equation 4.1,

$$\begin{aligned} Q(\lambda, \lambda^{old}) &= n log\lambda - \lambda \sum_{i=1}^{r} x_i - \lambda \Big(\sum_{i=r+1}^{n} (c_i + \frac{1}{\lambda^{old}}) \Big) \\ &= n log\lambda - \lambda \sum_{i=1}^{r} y_i - \lambda \Big(\sum_{i=r+1}^{n} (c_i + \frac{1}{\lambda^{old}}) \Big) \end{aligned} \tag{4.7}$$

4.3 Maximization

$$\frac{\partial Q(\lambda, \lambda^{old})}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{r} y_i - \sum_{i=r+1}^{n} c_i + \frac{n-r}{\lambda^{old}} := 0$$

Solving for λ we get,

$$\lambda = \frac{n}{\sum_{i=1}^{n} y_i + \frac{n-r}{\lambda^{old}}}$$

5 PROGRAMMING

5.1 CLUSTERS

Figure 5.1, 5.2 and 5.3 show clusters for different K values.

5.2 OBJECTIVE FUNCTION

Figure 5.4 shows objective function for different initializations.

5.3 CONVERGENCE

Yes, it is gauranteed to converge. However, the clustering might not be optimal as it might converge to local optima as compared to global optima. Correct initialization is very important if you want to make it coverge to global optimum.

5.4 VECTOR QUANTIZATION

Figure 5.5, 5.6 and 5.7 shows image reconstruction for different K values.

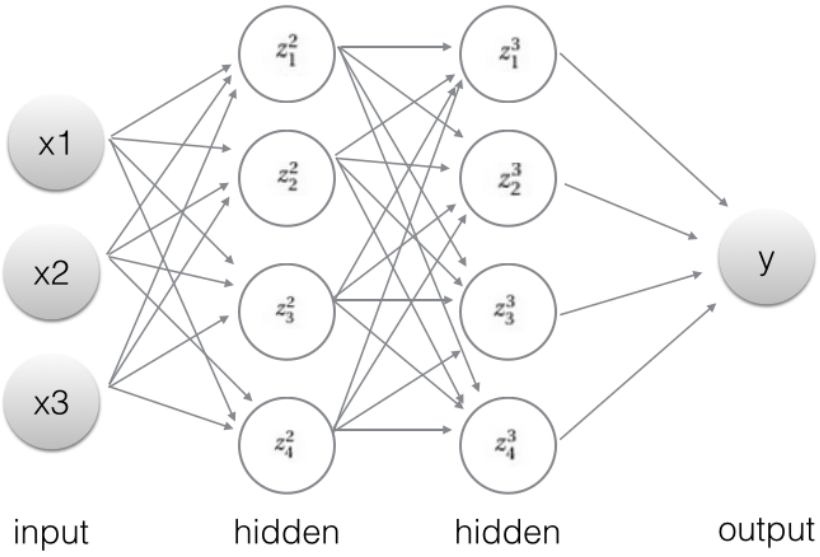

Figure 2.1: neural network with multiple hidden layers

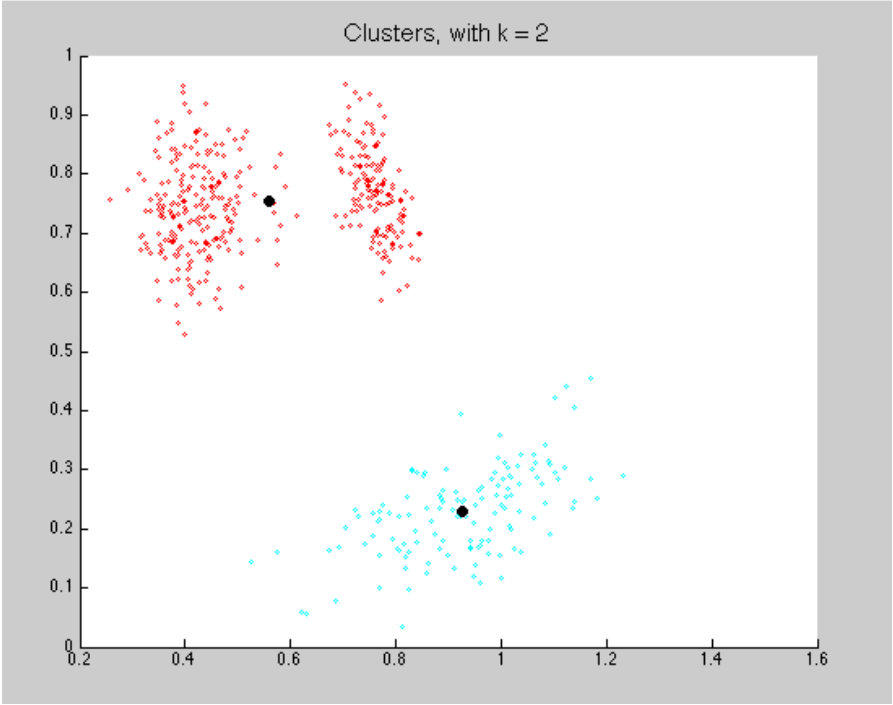

Figure 5.1: With 2 clusters, black dots represent centroids

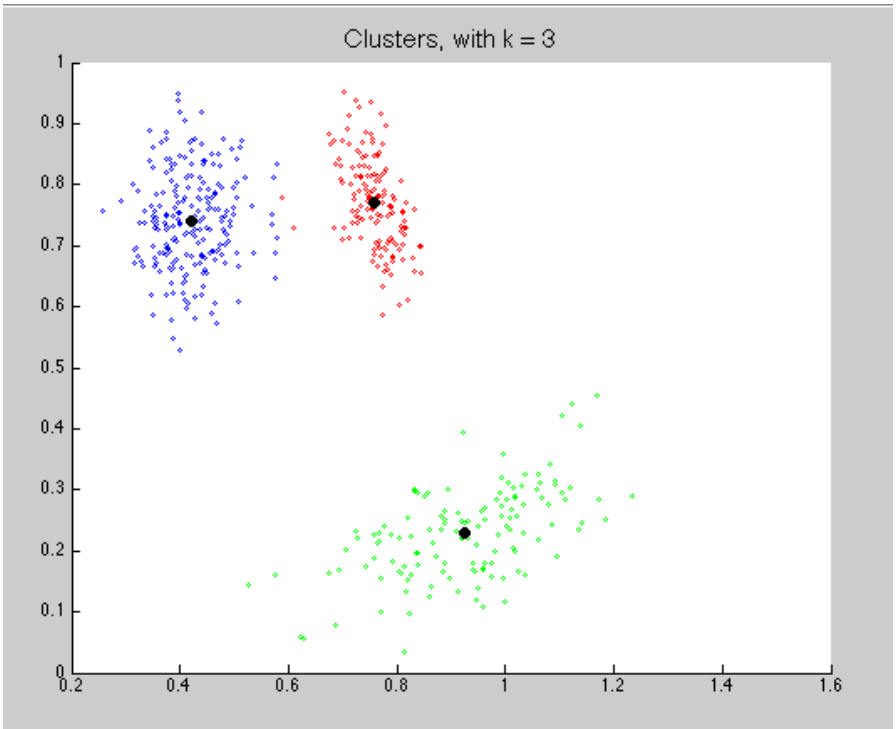

Figure 5.2: With 3 clusters, black dots represent centroids

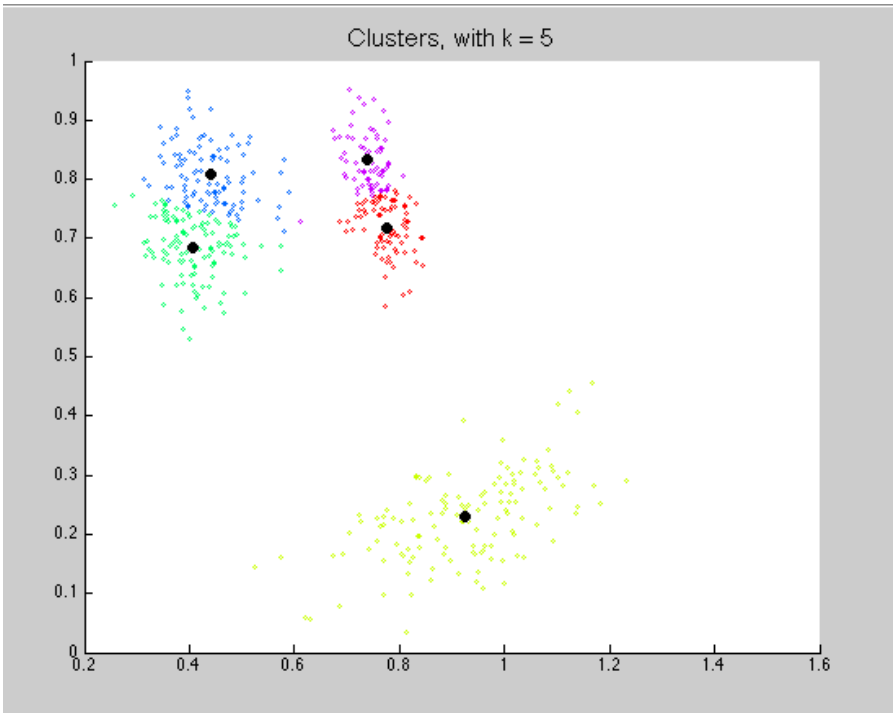

Figure 5.3: With 5 clusters, black dots represent centroids

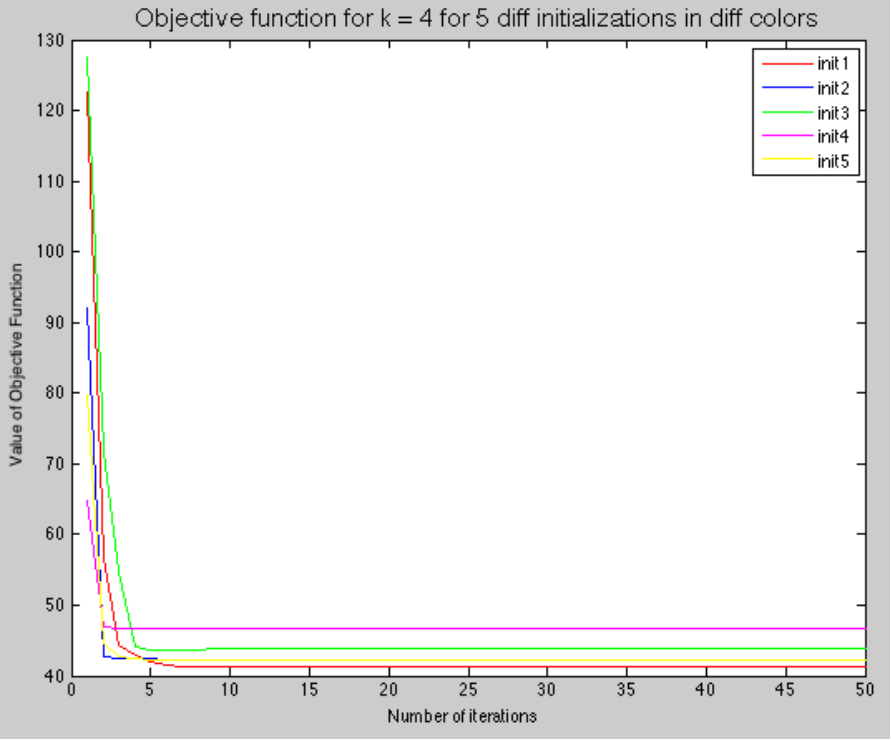

Figure 5.4: Objective function for 5 different initializations

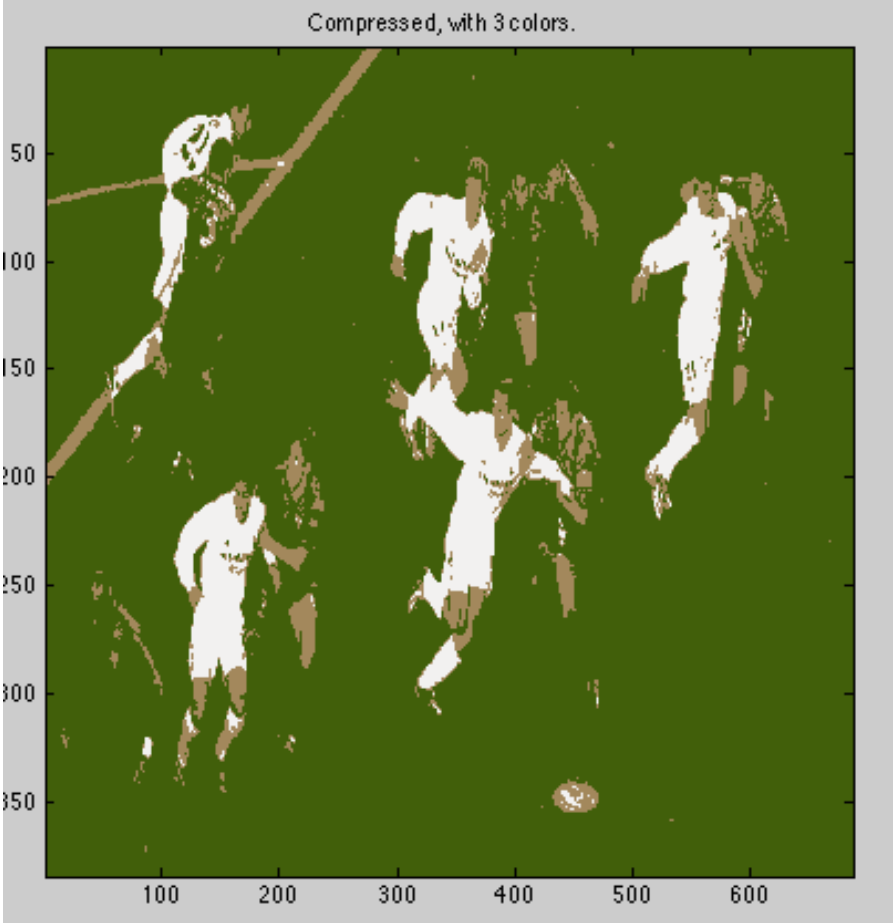

Figure 5.5: Image reconstruction with k=3

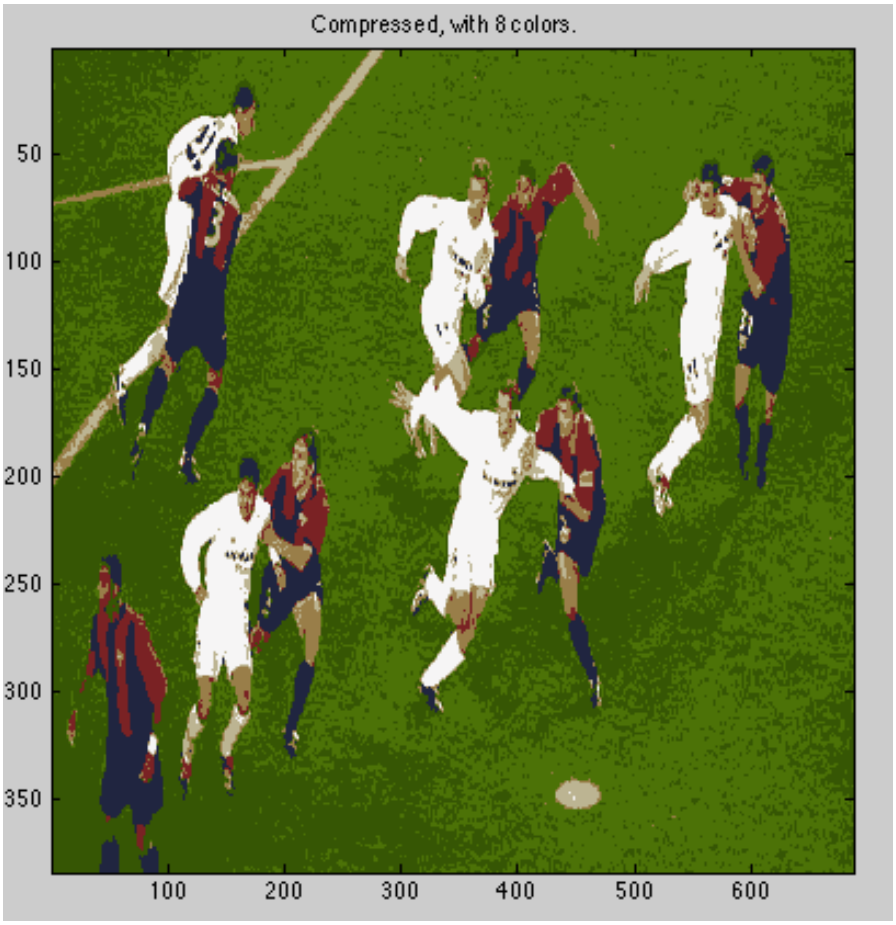

Figure 5.6: Image reconstruction with k=8

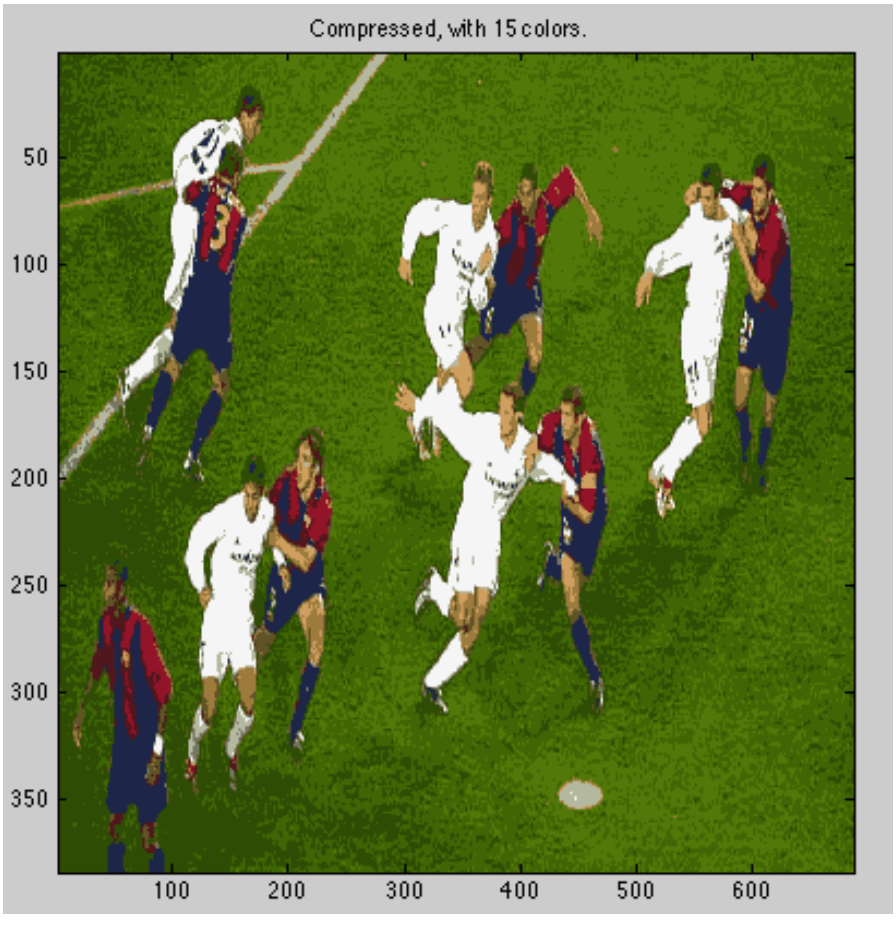

Figure 5.7: Image reconstruction with k=15