EECS 445 F14
HW # 4

WU Tongshuang 40782356

November 11, 2014

1 K-means for image compression

a

Figure 1 shows the original image.

Figure 1: The original image of maindrill

b and c

Figure 2 shows the compressed image.

Figure 2: The compressed image of maindrill

Comparing the two image, the nose and fur where contrast is significant are preserved, while fur with low contrast rate (e.g. bottom left region of the region) are not preserved well, probably due to more pixels clustering into one group because of their high similarity.

d

To represent one of the 16 colors, it requires $log_2 16 = 4$ bits per pixel. Since The original image use 3 bytes = 24 bits per pixel,

Since we can expressed the line as

$$\text{compression factor} = \frac{\text{space used to store original image}}{\text{space used to store compressed image}} = \frac{24}{4} = 6$$

.

2 Cross-Validation on Hyper-Parameters of SVM

a

As discussed in homework 3,

$$\nabla E^{(i)}(w,b) = \frac{1}{N}w - C \times I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}x^{(i)}$$

$$\frac{\partial}{\partial b}E^{(i)}(w,b) = -C \times I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}$$

Therefore, combining all $E^{(i)}(w,b)$ and $\frac{\partial}{\partial b}E^{(i)}(w,b)$ together:

$$\nabla E(w,b) = \sum_{i=1}^{N}(\frac{1}{N}w - C \times I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}x^{(i)})$$

$$\frac{\partial}{\partial b}E(w,b) = \sum_{i=1}^{N} -C \times I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}$$

The update rule is described in 1, where

$$\alpha(j) = \frac{\eta_0}{1 + j\eta_0}$$

Algorithm 1 SVM Stochastic Gradient Descent

```
w* ← 0
b* ← 0
for j = 1 to NumIterations do
  for i = 1 to N do
    w_grad ← ∇_w E^(i)(w*, b*);
    w_grad ← ∂/∂b E^(i)(w*, b*);

    w* ← w* − α(j) w_grad
    b* ← w* − 0.01 α(j) b*
  end for
end for
return w*
```

b

Please refer to the code for implementation details.

The resulted hyper-parameters are

$$C = 0.1, \eta_0 = 1.0, \text{ with validation error } e = 13.3\%$$

Train the whole set with these hyper parameter,

$$\mathbf{w} = \begin{bmatrix} -0.2860 \\ 0.0410 \\ -0.4307 \end{bmatrix}, b = 0.1475, \text{with test error } e = 6.50\%$$

c

Please refer to the code for implementation details.

The resulted hyper-parameters are

$$C = 0.1, \eta_0 = 0.01, \text{with validation error } e = 14.83\%$$

Train the whole set with these hyper parameter,

$$\mathbf{w} = \begin{bmatrix} -0.2965 \\ 0.0421 \\ -0.4068 \end{bmatrix}, b = 0.0356, \text{with test error } e = 6.00\%$$

3 Cross-Validation on Hyper-Parameters of SVM

a

The figure is shown as Figure 3. *accuracy* = 87.00%

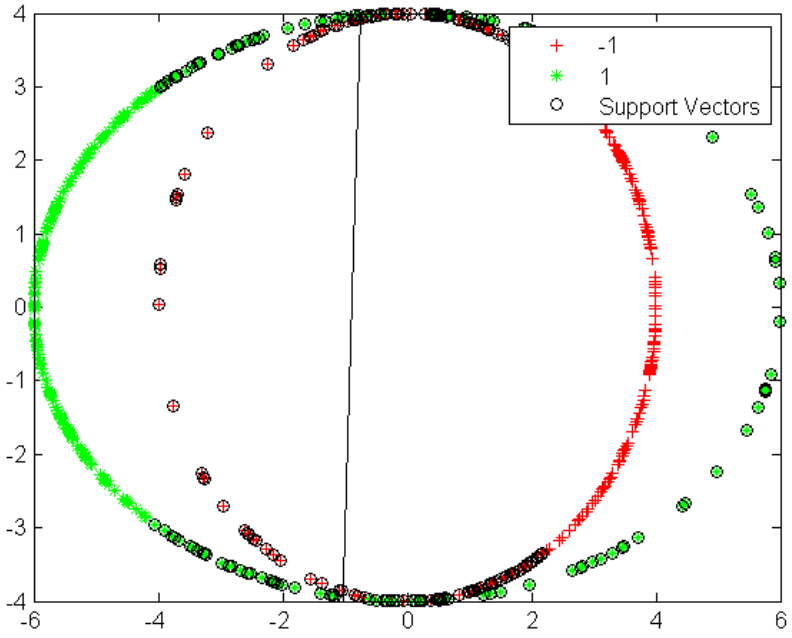


Figure 3: the training data and the separating hyperplane for Linear Kernel

b

The figure is shown as Figure 4. $accuracy = 93.50\%$

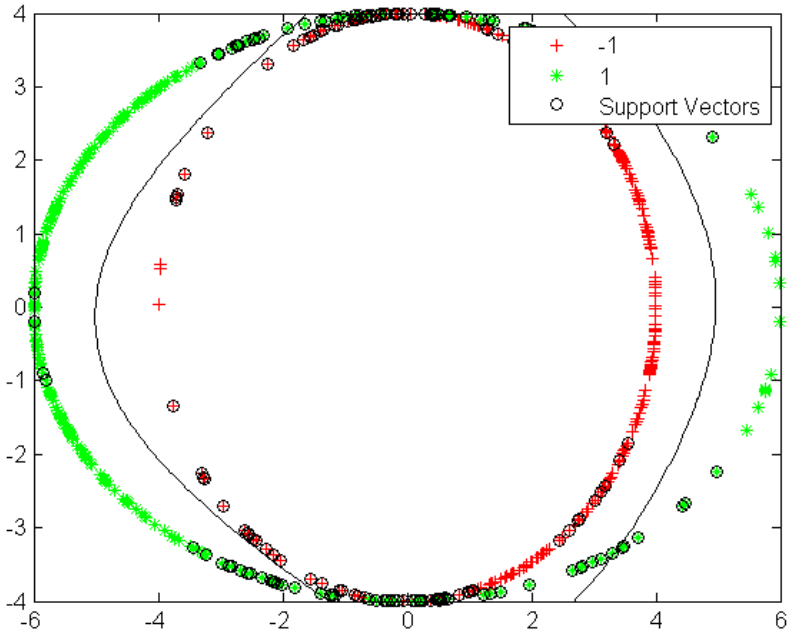


Figure 4: the training data and the separating hyperplane for RBF Kernel

c

The *RBF* Kernel performs better than *Linear* Kernel since its resulted accuracy is higher. This is because, from the figure, it apparent that the data is not linear-separatable, so using the *Linear* Kernel (or no kernel) cannot work perfectly since it's trying to find a straight line to distinguish between groups. Meanwhile, *RBF* Kernel, a non-linear kernel, maps the features to a higher dimensional space to make it linearly separable, which helps SVM to run more accurately.

d

Table 1: Tested hyper parameters and their corresponding frequancy

σ	ave. validation acc	test acc
0.2	91.83%	91.70%
0.5	92.00%	93.50%
1.0	86.83%	93.40%
1.5	86.67%	93.10%
2.0	84.33%	89.40%
2.5	81.83%	88.00%
3.0	81.50%	86.60%

From the table, we can conclude $\sigma = 0.5$ is the best hyper-meter since it reaches the highest average validation accuracy.

e

Linear Kernel: $accuracy = 87.00\%$. The figure is shown as Figure 5.

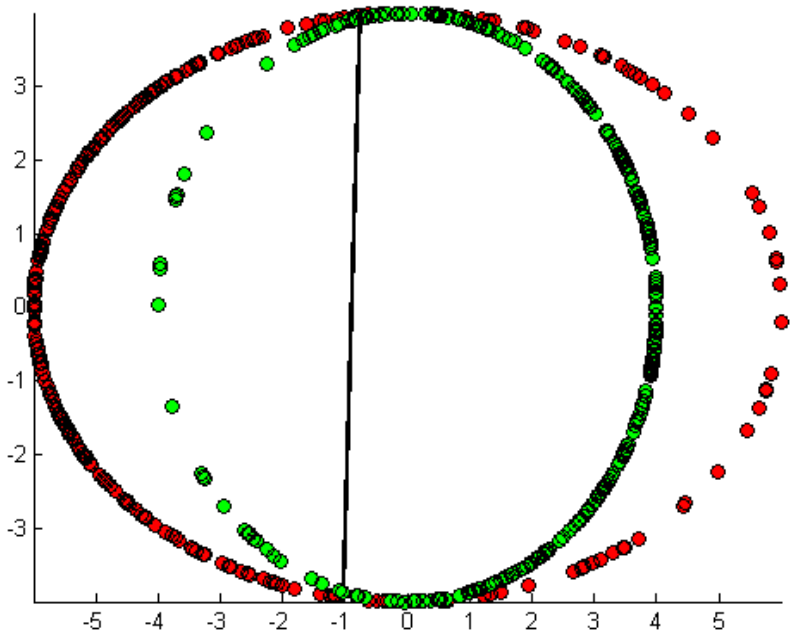


Figure 5: the training data and the separating hyperplane for Linear Kernel using libsvm

RBF Kernel: $accuracy = 92.50\%$. The figure is shown as Figure 6.

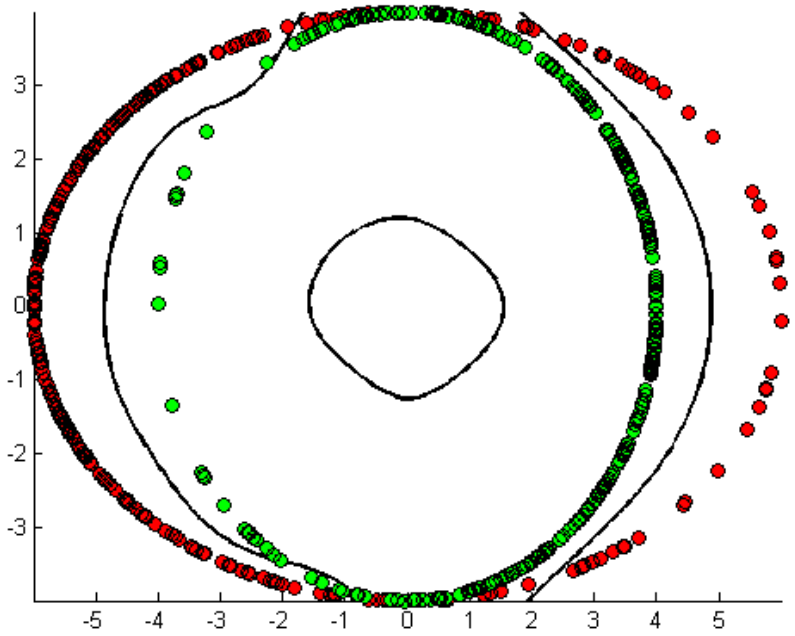


Figure 6: the training data and the separating hyperplane for RBF Kernel using libsvm

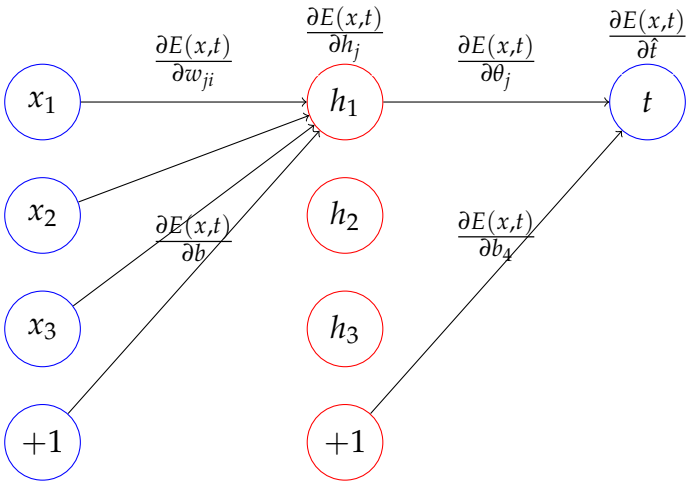
Table 2: Tested hyper parameters and their corresponding frequancy

σ	ave. validation acc	test acc
0.2	90.83%	93.00%
0.5	91.83%	91.50%
1.0	91.67%	89.90%
1.5	91.50%	88.90%
2.0	91.50%	86.50%
2.5	91.33%	84.90%
3.0	91.50%	83.30%

From the table, we can conclude $\sigma = 0.5$ is the best hyper-meter since it reaches the highest average validation accuracy.

4 Update rules for a 2-layer Neural Network

4.1 a



Given $E(x,t) = -(tlog(\hat{t}) + (1-t)log(1-\hat{t}))$, with the backward propagation

shown above, we can compute:

$$\begin{aligned}
\frac{\partial E(x,t)}{\partial \hat{t}} &= \frac{\partial}{\partial \hat{t}}(-(tlog(\hat{t}) + (1-t)log(1-\hat{t}))) \\
&= -\frac{t}{\hat{t}} + \frac{1-t}{1-\hat{t}} \\
&= \frac{\hat{t}-t}{(1-\hat{t})\hat{t}} \\
&= \frac{1+e^{-(b_4+\theta^T\mathbf{h})} - t(1+e^{-(b_4+\theta^T\mathbf{h})})^2}{e^{-(b_4+\theta^T\mathbf{h})}} \\
\frac{\partial \hat{t}}{\partial \theta_j} &= \frac{\partial}{\partial \theta}\sigma(b_4+\theta_j h_j) \\
&= \frac{\partial}{\partial \theta}\frac{1}{1+e^{-(b_4+\theta_j h_j)}} \\
&= h_j \cdot \frac{e^{-(b_4+\theta_j h_j)}}{(1+e^{-(b_4+\theta_j h_j)})^2} \\
\frac{\partial \hat{t}}{\partial \theta} &= \frac{\partial}{\partial \theta}\sigma(b_4+\theta^T\mathbf{h}) \\
&= \frac{\partial}{\partial \theta}\frac{1}{1+e^{-(b_4+\theta^T\mathbf{h})}} \\
&= -\mathbf{h} \cdot \frac{e^{-(b_4+\theta^T\mathbf{h})}}{(1+e^{-(b_4+\theta^T\mathbf{h})})^2} \\
\frac{\partial E(x,t)}{\partial \theta_j} &= \frac{\partial \hat{t}}{\partial \theta_j}\frac{\partial E(x,t)}{\partial \hat{t}} \\
\frac{\partial \hat{t}}{\partial b_4} &= \frac{\partial}{\partial b_4}\sigma(b_4+\theta^T\mathbf{h}) \\
&= \frac{\partial}{\partial b_4}\frac{1}{1+e^{-(b_4+\theta^T\mathbf{h})}} \\
&= \frac{e^{-(b_4+\theta^T\mathbf{h})}}{(1+e^{-(b_4+\theta^T\mathbf{h})})^2}
\end{aligned}$$

Therefore,

$$
\begin{aligned}
\nabla_\theta E(x,t) &= \frac{\partial \hat{t}}{\partial \theta}\frac{\partial E(x,t)}{\partial \hat{t}} \\
&= \frac{1+e^{-(b_4+\theta^T\mathbf{h})}-t(1+e^{-(b_4+\theta^T\mathbf{h})})^2}{e^{-(b_4+\theta^T\mathbf{h})}} \cdot (\mathbf{h}\cdot\frac{e^{-(b_4+\theta^T\mathbf{h})}}{(1+e^{-(b_4+\theta^T\mathbf{h})})^2}) \\
&= \mathbf{h}(\frac{1}{1+e^{-(b_4+\theta^T\mathbf{h})}}-t) \\
&= (\mathbf{W}\mathbf{x}+\mathbf{b})(\frac{1}{1+e^{-(b_4+\theta^T(\mathbf{W}\mathbf{x}+\mathbf{b}))}}-t) \\
&= \mathbf{h}\cdot(\sigma(b_4+\theta^T\mathbf{h})-t) \\
\frac{\partial E(x,t)}{\partial b_4} &= \frac{\partial \hat{t}}{\partial b_4}\frac{\partial E(x,t)}{\partial \hat{t}} \\
&= \frac{1+e^{-(b_4+\theta^T\mathbf{h})}-t(1+e^{-(b_4+\theta^T\mathbf{h})})^2}{e^{-(b_4+\theta^T\mathbf{h})}} \cdot (\frac{e^{-(b_4+\theta^T\mathbf{h})}}{(1+e^{-(b_4+\theta^T\mathbf{h})})^2}) \\
&= (\frac{1}{1+e^{-(b_4+\theta^T\mathbf{h})}}-t) \\
&= (\frac{1}{1+e^{-(b_4+\theta^T(\mathbf{W}\mathbf{x}+\mathbf{b}))}}-t) \\
&= \sigma(b_4+\theta^T\mathbf{h})-t
\end{aligned}
$$

b

$$
\begin{aligned}
\frac{\partial \hat{t}}{\partial h_j} &= \frac{\partial}{\partial h_j}\sigma(b_4 + \theta_j h_j) \\
&= \frac{\partial}{\partial h_j}\frac{1}{1+e^{-(b_4+\theta_j h_j)}} \\
&= \theta_j \cdot \frac{e^{-(b_4+\theta_j h_j)}}{(1+e^{-(b_4+\theta_j h_j)})^2} \\
\frac{\partial \hat{t}}{\partial \mathbf{h}} &= \frac{\partial}{\partial \mathbf{h}}\sigma(b_4 + \theta^T \mathbf{h}) \\
&= \frac{\partial}{\partial \mathbf{h}}\frac{1}{1+e^{-(b_4+\theta^T \mathbf{h})}} \\
&= \theta^T \cdot \frac{e^{-(b_4+\theta^T \mathbf{h})}}{(1+e^{-(b_4+\theta^T \mathbf{h})})^2} \\
\frac{\partial E(x,t)}{\partial \mathbf{h}} &= \frac{\partial \hat{t}}{\partial \mathbf{h}}\frac{\partial E(x,t)}{\partial \hat{t}} \\
&= \frac{1+e^{-(b_4+\theta^T \mathbf{h})} - t(1+e^{-(b_4+\theta^T \mathbf{h})})^2}{e^{-(b_4+\theta^T \mathbf{h})}} \cdot (\theta^T \cdot \frac{e^{-(b_4+\theta^T \mathbf{h})}}{(1+e^{-(b_4+\theta^T \mathbf{h})})^2}) \\
&= \theta^T(\frac{1}{1+e^{-(b_4+\theta^T \mathbf{h})}} - t) \\
&= \theta^T(\frac{1}{1+e^{-(b_4+\theta^T (\mathbf{b}+\mathbf{W}\mathbf{x}))}} - t) \\
\frac{\partial \mathbf{h}}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}} tanh(\mathbf{b} + \mathbf{W}\mathbf{x}) \\
&= \mathbf{x}^T \cdot (1 - tanh^2(\mathbf{b} + \mathbf{W}\mathbf{x})) \\
&= \mathbf{x}^T \cdot sech^2(\mathbf{b} + \mathbf{W}\mathbf{x}) \\
\frac{\partial \mathbf{h}}{\partial \mathbf{b}} &= \frac{\partial}{\partial \mathbf{b}} tanh(\mathbf{b} + \mathbf{W}\mathbf{x}) \\
&= 1 - tanh^2(\mathbf{b} + \mathbf{W}\mathbf{x}) \\
&= sech^2(\mathbf{b} + \mathbf{W}\mathbf{x})
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\nabla_{\mathbf{W}} E(x,t) &= \frac{\partial \mathbf{h}}{\partial \mathbf{W}}\frac{\partial E(x,t)}{\partial \mathbf{h}} \\
&= \mathbf{x}^T \cdot sech^2(\mathbf{b} + \mathbf{W}\mathbf{x}) \cdot (\theta^T(\frac{1}{1+e^{-(b_4+\theta^T (\mathbf{b}+\mathbf{W}\mathbf{x}))}} - t)) \\
&= (\sigma(b_4 + \theta^T \mathbf{h}) - t)\theta^T sech^2(\mathbf{b} + \mathbf{W}\mathbf{x})\mathbf{x}^T \\
\nabla_{\mathbf{b}} E(x,t) &= \frac{\partial \mathbf{h}}{\partial \mathbf{b}}\frac{\partial E(x,t)}{\partial \mathbf{h}} \\
&= sech^2(\mathbf{b} + \mathbf{W}\mathbf{x}) \cdot (\theta^T(\frac{1}{1+e^{-(b_4+\theta^T (\mathbf{b}+\mathbf{W}\mathbf{x}))}} - t)) \\
&= (\sigma(b_4 + \theta^T \mathbf{h}) - t)\theta^T sech^2(\mathbf{b} + \mathbf{W}\mathbf{x})
\end{aligned}
$$