EECS 445 F14
HW # 5

WU Tongshuang 40782356

December 1, 2014

1 Gaussian mixtures 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. Also, the overall image is somehow darkened even compare to the k-means compressed version, probably due to the small cluster number.

d

To represent one of the 5 colors, it requires $log_2 5 \approx 3$ bits per pixel. Since The original image use 3 bytes = 24 bits per pixel,

Since we can expressed the line as

$$\text{compression factor} = \frac{\text{space used to store original image}}{\text{space used to store compressed image}} = \frac{24}{3} = 8$$

. This is a higher compression rate comparing to the k-means version. However, if use the same cluster number, then the compression rate should be the same.

e

Table 1: μ, σ and likelihood of Gaussian mixture output

k	μ_k	Σ_k
1	$\begin{bmatrix} 143.0288 \\ 191.8520 \\ 227.1027 \end{bmatrix}$	$\begin{bmatrix} 0.5624 & 0.1032 & -0.1274 \\ 0.1032 & 0.0486 & 0.0033 \\ -0.1274 & 0.0033 & 0.0656 \end{bmatrix}$
2	$\begin{bmatrix} 107.6263 \\ 117.2739 \\ 93.2895 \end{bmatrix}$	$\begin{bmatrix} 1.0065 & 1.0022 & 0.6306 \\ 1.0022 & 1.1431 & 0.8470 \\ 0.6306 & 0.8470 & 0.9878 \end{bmatrix}$
3	$\begin{bmatrix} 233.8846 \\ 86.9096 \\ 67.9841 \end{bmatrix}$	$\begin{bmatrix} 0.1282 & -0.1146 & -0.2562 \\ -0.1146 & 0.2683 & 0.4820 \\ -0.2562 & 0.4820 & 1.0843 \end{bmatrix}$
4	$\begin{bmatrix} 176.7728 \\ 148.8993 \\ 76.5845 \end{bmatrix}$	$\begin{bmatrix} 0.8923 & 0.6287 & -0.0757 \\ 0.6287 & 0.7927 & 0.4004 \\ -0.0757 & 0.4004 & 0.7669 \end{bmatrix}$
5	$\begin{bmatrix} 147.3821 \\ 147.4918 \\ 158.4752 \end{bmatrix}$	$\begin{bmatrix} 1.3819 & 0.1393 & -0.4975 \\ 0.1393 & 0.9266 & 1.0582 \\ -0.4975 & 1.0582 & 1.7573 \end{bmatrix}$

And the likelihood is -2.2778×10^5

Gaussian Processes

a

2 Distributed ray tracing

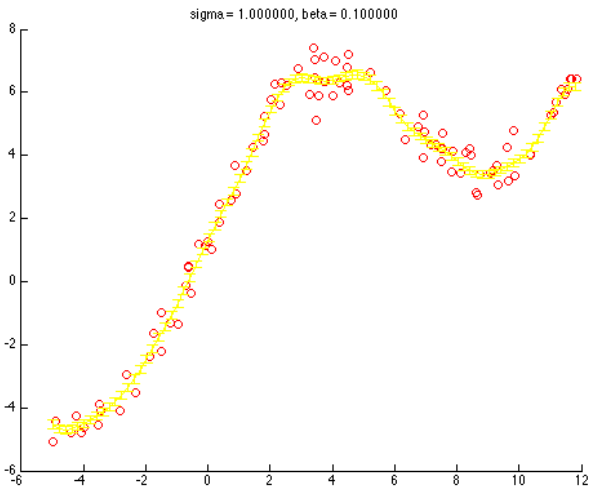
Please refer to Figure 3 for details.

b

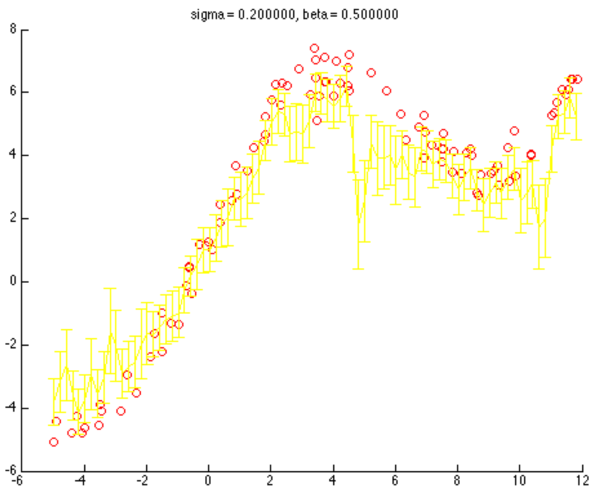
σ models the standard derivation (or equivalently, variance) of gaussian kernel, and the flatness of the predicted regression curve. When it is too large, the line may be over-smoothed, and when it is too small, the line varies too quickly.

c

β is a hyperparameter representing the precision of the noise. When it is too big, the confidence for the predicted regression will be much lower and the error bar becomes

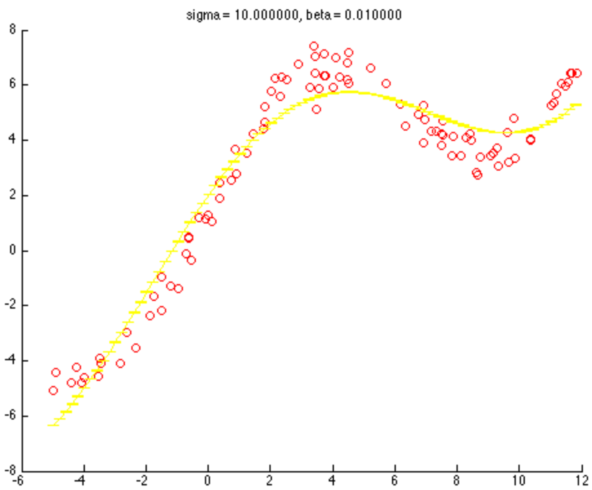


(a) $\sigma = 1.0, \beta = 0.1$



,

(b) $\sigma = 0.2, \beta = 0.5$



(c) $\sigma = 10.0, \beta = 0.01$

Figure 3: Images with different distributed ray tracing effects

much larger. When it is too small, we will be very confident about the noise, and the resulted regression curve will not have a valid errorbar.

d

the best parameter among the ones I tried is $\sigma = 2, \beta = 0.3$. the predicted regression curve with error bars is shown in Figure 4.

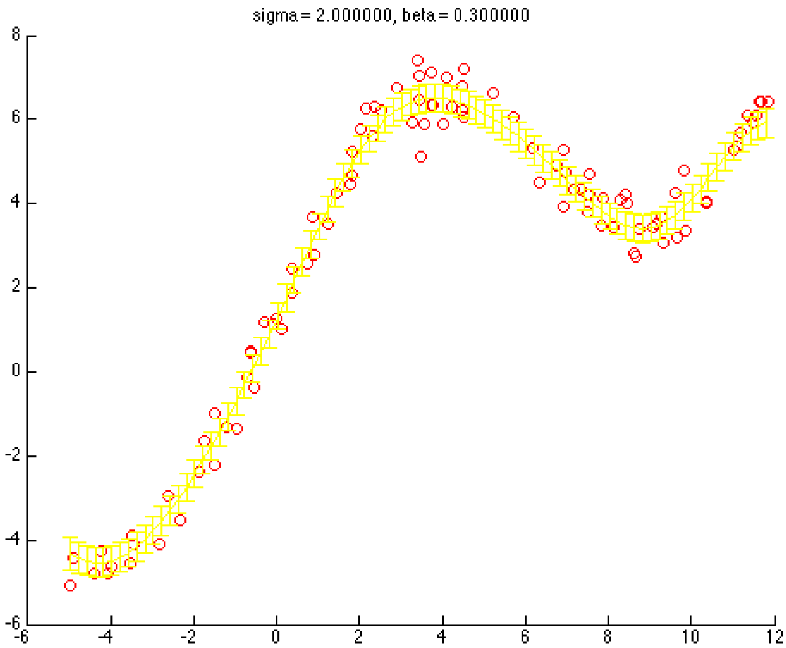


Figure 4: $\sigma = 2.0, \beta = 0.3$

3 Training Sparse Autoencoders

a

Figure 5 shows the original image.

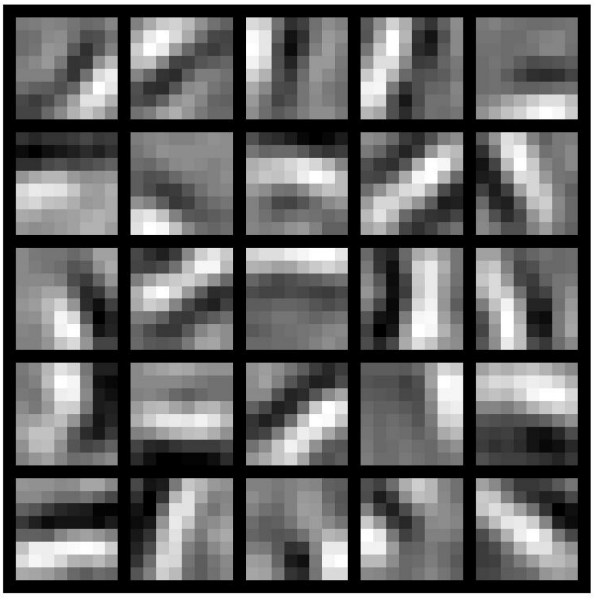


Figure 5: Visualization obtained

b

I used Liblinear SVM with L2 regularization. The test accuracy is 98.37%.