TSBK35 Lab1

Johnny Elmér - jonel107@student.liu.se

February 2022

1 Method

Here the method used to solve the lab are presented in two different sections, mainly the the process used for audio and images respectively.

1.1 Audio

The audio files were loaded into MATLAB using audioread and then scaled to be between -128 and 128. The probability for each sample was calculated using a for loop and was then put into the entropy function for $H(X_i)$ to calculate the entropy. To calculate $H(X_i, X_{i+1})$ a similar approach was used, but this time the calculating the pair probability before using the corresponding entropy equation. To calculate the conditional entropy $H(X_{i+1}|X_i)$ a simple subtraction was performed ($H(X_i, X_{i+1}) - H(X_i)$).

To calculate the data rate for memoryless Huffman coding the given Huffman function was used with the original probability function as input. After this Huffman coding of differences was performed by first creating a vector containing the difference between the predicted values and the real values using the different predictors given (using an audio signal as the base). Then the probability of each sample was calculated for each sample level and put into a probability matrix. Subsequently, the data rate was calculated using the given Huffman code with the probability matrix as the input as well as the entropy ($H(X_i)$) to allow for comparisons.

1.2 Images

Similar to how the process was done in 1.1 the images were loaded into MATLAB using the imread function and then converted into the data type double for ease of computation. Then the probability was calculated, in this case using two for loops as the images are two-dimensional and both height and width need to be taken into account. $H(X_{i,j})$ was then calculated using the corresponding function. Like in 1.1 the calculation of $H(X_{i,j}, X_{i+1,j})$ and $H(X_{i,j}, X_{i,j+1})$ was done in a similar fashion to the first entropy but with slight adjustments to how the probability was calculated. (For the horizontal and vertical entropy the

probability matrices were saved separately for later use in the data rate calculations.) Similarly there needed to be a step introduced in between calculating the probability and calculating the entropy where the probability matrix was formatted into a vector which did not contain any non-zero elements. To calculate the conditional entropies a similar approach as before was used. $H(X_{i+1,j}|X_{i,j})$ was calculated by subtracting $H(X_{i,j})$ from $H(X_{i,j}, X_{i+1,j})$ and $H(X_{i,j+1}|X_{i,j})$ was calculated by subtracting $H(X_{i,j})$ from $H(X_{i,j}, X_{i,j+1})$.

To calculate the data rate for memoryless Huffman coding the given Huffman function was again used but this time for both the probability of the horizontal and vertical probability matrices. Huffman coding of the difference was done in the exact same fashion as previously with the main change being adjustments of the prediction equations and modifications to work with the two-dimensionality of the images. The entropy was also calculated to allow for easier comparison.

2 Results

Here the results of the lab are presented in two different sections, audio (2.1 - 2.2) and images (2.3 - 2.4).

2.1 Entropy calculations for audio

This part presents the results of the different entropy calculations on the audio clips. Entropy calculations done with predictions in connection to Huffman coding are presented in 2.2.

Entropy type	**hey04**	**nuit04**	**speech**
$H(X_i)$	7.1638	6.8060	5.4340
$H(X_i, X_{i+1})$	11.4363	9.5281	8.7231
$H(X_{i+1}\|X_i)$	4.2725	2.7221	3.2891

Table 1: Results of the entropy calculations on the different audio files.

2.2 Huffman coding for audio

Table 2 presents the data rates calculated using memoryless Huffman coding, while Table 3 presents the data rate as well as the entropy calculated when using predictions.

2.3 Entropy calculations for images

This part presents the results of the different entropy calculations on the images.

hey04	**nuit04**	**speech**
7.1968	6.8273	5.4512

Table 2: Data rate results from the memoryless Huffman coding of the audio clips.

$P_i =$	**hey04 R/E**	**nuit04 R/E**	**speech R/E**
X_{i-1}	4.3405 / 4.3115	2.7894 / 2.7390	3.9366 / 3.8883
$2X_{i-1} - X_{i-2}$	4.3191 / 4.2810	1.9876 / 1.8864	3.3527 / 3.2823

Table 3: Data rate results from Huffman coding using predictions.R stands for data rate and E stands for Entropy.

Entropy type	**baboon**	**boat**	**woodgrain**
$H(X_{i,j})$	7.4745	7.1238	6.3562
$H(X_{i,j}, X_{i+1,j})$	13.7566	12.0492	12.1658
$H(X_{i,j}, X_{i,j+1})$	14.0210	11.5126	10.5594
$H(X_{i+1,j}\|X_{i,j})$	6.2821	4.9254	5.8096
$H(X_{i,j+1}\|X_{i,j})$	6.5465	4.6889	4.2032

Table 4: Results of the entropy calculations on the different images.

2.4 Huffman coding for images

Table 5 presents the data rates calculated using memoryless Huffman coding both vertically and horizontally in each image, while Table 6 presents the data rate as well as the entropy calculated when using predictions.

baboon	**boat**	**woodgrain**
7.5171	7.1468	6.3914

Table 5: Data rate results from the memoryless Huffman coding of the images.

$P_{i,j} =$	**baboon R/E**	**boat R/E**	**woodgrain R/E**
$X_{i,j-1}$	6.5902 / 6.5608	5.2492 / 5.2134	6.4006 / 6.3771
$X_{i-1,j}$	6.9298 / 6.8954	5.0144 / 4.9782	4.4621 / 4.4231
$X_{i-1,j} + X_{i,j-1} - X_{i-1,j-1}$	6.8639 / 6.8303	4.8934 / 4.8600	4.6684 / 4.6375

Table 6: Data rate results from Huffman coding using predictions.R stands for data rate and E stands for Entropy.

3 Discussion

Listening to the audio clips it is fairly easy to speculate how the characteristics of the different sounds affect the data rate and entropy results. For example, hey04 is rather irregular in its sound and is therefore harder to predict, which means that the entropy and data rate values are going to be higher than that of the other sound. Meanwhile nuit04 is a lot more consistent in what notes are being played in the background and in speech the speaker is rather monotone and there aren't a lot of overlapping sounds. Because of the irregularity of nuit04 the best predictor was the one that looked the furthest into the past.

When looking at the images there isn't as clear of a difference between the data rates and entropies as there are with the audio clips, but there are still some clear patterns. For example, the woodgrain image is nearly always somewhat "better" than the other two images, which makes sense seeing as the image doesn't contain a lot of complex structures or irregular shapes. Similarly, the boat image is "better" than the baboon image, which most likely is because of there being larger areas in the image of similar colour values with minimal changes (like the sky) in comparison to the very noisy parts in the baboon image (like the hair). When looking at the predictors the largest difference in data rate and entropy results can be seen for the woodgrain. The first predictor is of a horizontal nature and will therefore more frequently encounter darker areas in the woodgrain, meaning that it is harder to predict correctly. Meanwhile, the second and third predictor, vertical and diagonal respectively, have a higher chance of traversing larger areas of the image before encountering some sort of shift (from dark to light and vice versa), meaning that it has an easier time predicting what will come next.

To see my code, visit the following link: `https://github.com/JohnnyElmer/TSBK35/tree/main/Lab1`