Results for Assignment 1 - Group xx

Markus Ruplitsch, Julia Tschuden

April 14, 2020

1 Task 1

Information which was deduced from the text:
$P(D = 0) = P(H) = 0.95$,
$P(D = 1) = P(A) = 0.04$,
$P(D = 2) = P(C) = 0.01$,
$P(T = 1|C) = 0.98; P(T = 0|C) = 1 - P(T = 1|C) = 0.02$,
$P(T = 1|H) = 0.01; P(T = 0|H) = 1 - P(T = 1|H) = 0.99$,
$P(T = 1|A) = 0.20; P(T = 0|A) = 1 - P(T = 1|A) = 0.80$.

Table 1: Probability table.

T/D	H (D = 0)	A (D = 1)	C (D = 2)	p(T_i)
T = 1	0.0259	0.0011	0.0003	0.0273
T = 0	0.9241	0.0389	0.0097	0.9727
p(D_i)	0.95	0.04	0.01	1

At first calculate the marginal probability.
$P(T = 1) = 0.01 * 0.95 + 0.2 * 0.04 + 0.98 * 0.01 = 0.0273$
$P(T = 0) = 0.99 * 0.95 + 0.8 * 0.04 + 0.02 * 0.01 = 0.9727$

Then calculate all joint probabilities.
$P(H, T = 1) = P(H) * P(T = 1) = 0.95 * 0.0273 = 0.0259$
$P(A, T = 1) = P(A) * P(T = 1) = 0.04 * 0.0273 = 0.0011$
$P(C, T = 1) = P(C) * P(T = 1) = 0.01 * 0.0273 = 0.0003$

$P(H, T = 0) = P(H) * P(T = 0) = 0.95 * 0.9727 = 0.9241$
$P(A, T = 0) = P(A) * P(T = 0) = 0.04 * 0.9727 = 0.0389$
$P(C, T = 0) = P(C) * P(T = 0) = 0.01 * 0.9727 = 0.0097$

2 Task 2

Blabla explanation.
Blabla.

3 Task 3

Blabla explanation.
Blabla.

4 Task 4

4.1 Verify conditional mean.

$E_Y[Y|X=x] = \sum_{n=1}^{N} y_n * p(Y=y_n|X=x)$
$= \sum_{n=1}^{N} y_n * \frac{p(X=x|Y=y_n)*p(Y=y)}{p(X=x)}$,

where $p(X=x)$ is a constant and can be neglected and $p(Y=y_n)=1$ because of the uniform distribution. Which brings the following equation:

$= \sum_{n=1}^{N} y_n * p(X=x|Y=y)$,

however, the equation needs to be normalized. Which is why the formular needs to be divided by $\sum_{n=1}^{N} p(X=x|Y=y)$, which brings following equation:

$E_Y[Y|X=x] = \frac{\sum_{n=1}^{N} y_n * p(X=x|Y=y)}{\sum_{n=1}^{N} p(X=x|Y=y)}$

4.2 Verify MAP.

$P(Y|X=x) = \frac{P(X=x|Y)*p(Y)}{p(X=x)}$
We seek value $y_n \in Y$ that maximises the posterior.

$\hat{y}_{MAX} = argmaxp(Y=y_n|X=x)$
$= argmax\frac{p(X=x|Y=y_n)*p(Y=y_n)}{p(X=x)}$,

where $p(X=x)$ is a constant and can be neglected and $p(Y=y_n)=1$ because of the uniform distribution. Which brings the following equation:

$\hat{y}_{MAX} = argmaxp(X=x|Y=y_n)$

4.3 Plots and discussion.

In our experiments we used three different values for $\sigma : 0.25, 0.5, 1$. The smaller the sigma, the smaller the amount of noise added to the testing images. The bigger the sigma, the bigger the amount of added noise. Generally speaking: the more noise an image has, the harder it is for an algorithm to remove the noise.

4.3.1 Sigma with 0.25

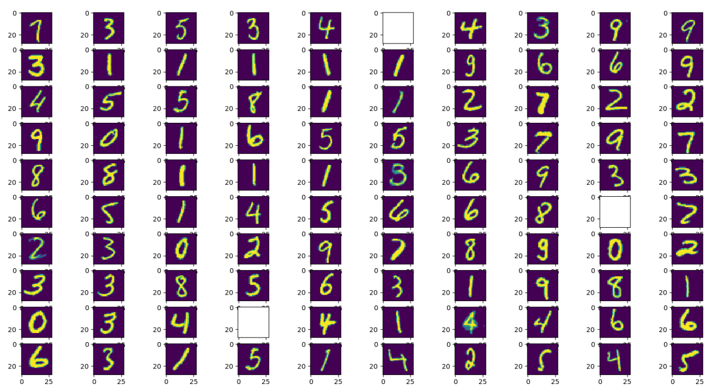

Figure 1: Result of CM algo with sigma 0.25.

As can be seen in figure 1 the CM algorithm did a pretty good job at removing the noise. There are only a few images, where the noise was not removed completely (e.g. the 3 at position x=6,y=5, where the algorithm almost made an 8 out of the 3) The MAP algorithm, however, did an excellent job, as can be seen in figure 2.

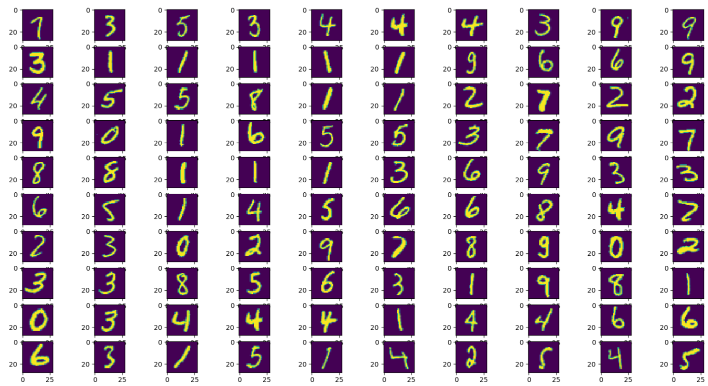

Figure 2: Result of MAP algo with sigma 0.25.

4.4 Sigma with 0.50

The labels for the choosen images can be seen in figure 3.

As can be seen in figure 5 for a sigma of 0.5 the MAP algo again did a very good job. All noisy images were correctly assigned. And while the mc algo performed good as well, the results are a bit different, as can be seen in figure 4. The algorithm connected lines which were disconnected in the original image (e.g. the sloppy 8 at position x=6;y=6 or the sloppy 9 at position x=9;x=8) and thickened lines, which were thin (because of the pen pressure) in the origiinal image (e.g. the 5 at position x=3;y=9 or the 8 at position x=3;y=7)

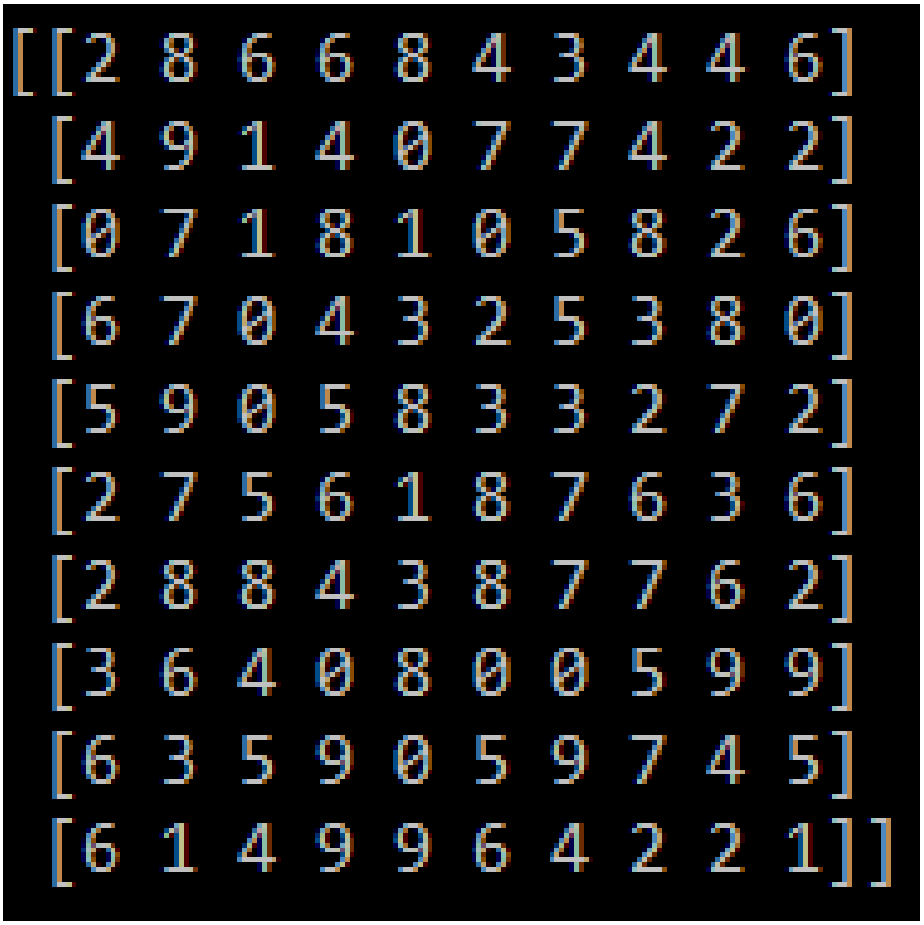

Figure 3: Labels.

4.5 Sigma with 1.00

The labels for the choosen images can be seen in figure 3.

With a larger sigma, the MAP algo did multiple misassignments, as can be seen in figure 8, if one compares the images to the ground truth labels. More than 15% of the images are misassigned, in addition, none of the numbers which were misassigned resemble each other (e.g. 0 instead of 2 at position x=8;y=10) The results for the CM algo, which can be seen in figure 7 are as worse as we expected for a sigma of 1.0. Most of the images have a lot of noise in it, some even have traces of other numbers mixed in. (e.g. the 3, where the noise resembles a 9 at position x=2;y=3)

Which brings following conclusion: Both algorithms don't work too well with a lot of noise, however, if the noise is small both work pretty well!

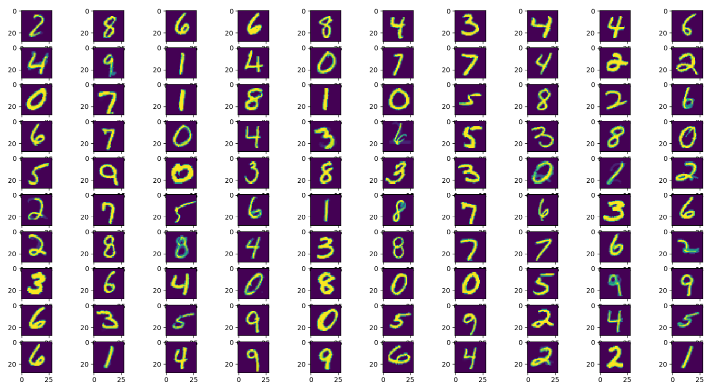

Figure 4: Result of CM algo with sigma 0.50.

4.6 Differences between CM and MAP

The MAP algorithm returns the image with the highest probability - the image the noisy image resembles the most. The CM algorithm however considers all the images which resemble the noisy image (= images with a high probability weight more than images with a low one) and reconstructs the image by taking the mean of all images.

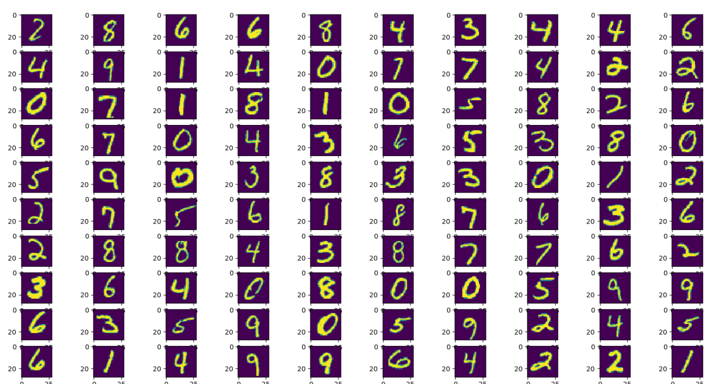

Figure 5: Result of MAP algo with sigma 0.50.

```
[[5 4 6 0 7 1 3 8 8 2]
 [0 4 7 8 9 0 7 7 3 3]
 [2 3 9 8 9 7 4 0 2 3]
 [2 5 4 7 3 2 1 9 2 5]
 [7 8 6 7 2 4 8 7 7 1]
 [7 9 7 3 1 1 8 1 9 2]
 [7 3 2 1 9 0 2 2 2 3]
 [9 7 6 6 7 6 2 1 7 9]
 [4 0 2 5 7 8 8 2 3 7]
 [8 3 4 3 1 6 8 2 0 8]]
```

Figure 6: Labels.

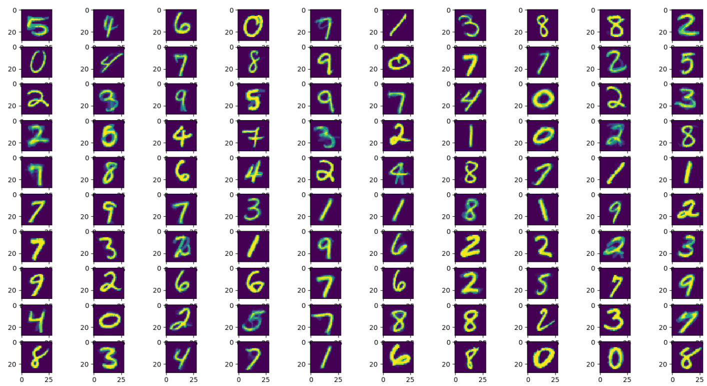

Figure 7: Result of CM algo with sigma 1.

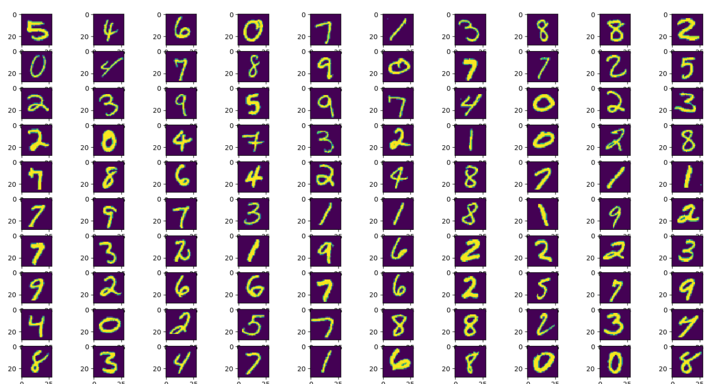

Figure 8: Result of MAP algo with sigma 1.0.