CMPUT 366—Intelligent Systems

Assignment 3

Fall 2004
Department of Computing Science
University of Alberta

Due: *Thursday, November 4 at 23:59:59 local time*
Worth: 10% of final grade
(7 questions worth 1%, 2%, 1%, 2%, 1%, 1% and 2%.)
Instructor: Dale Schuurmans, Ath409, x2-4806, dale@cs.ualberta.ca

---

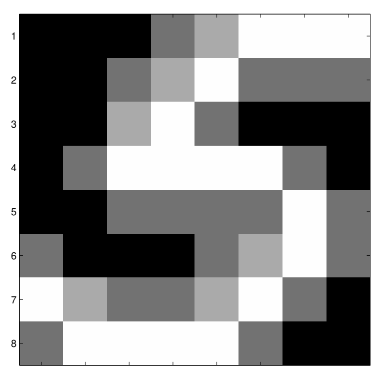
Note: Submit six Matlab functions in files called gen1.m, rec1.m, gen2.m, rec2.m, gen3.m, rec3.m, and one short derivation in a file called note3.pdf.

Handwritten digit recognition

In this assignment you will implement simple handwritten digit recognizers using simple probability models. The first step is to go to the course webpage and get the files model.mat and show.m. In Matlab, type "`load model.mat`." This will load the arrays `digits` ($8 \times 8 \times 10$), `noise` (1×7), `htrans` (1×3), and `vtrans` (1×3). The array `digits` contains image templates for the digits 1,...,9,0. To see them, type `show(digits(:,:,`d`))` for $d = 1, ..., 10$. There is also a collection of real handwritten digits in the file data.mat which contains 1100 examples each digit in an $8 \times 8 \times 1100 \times 10$ array called `data`.

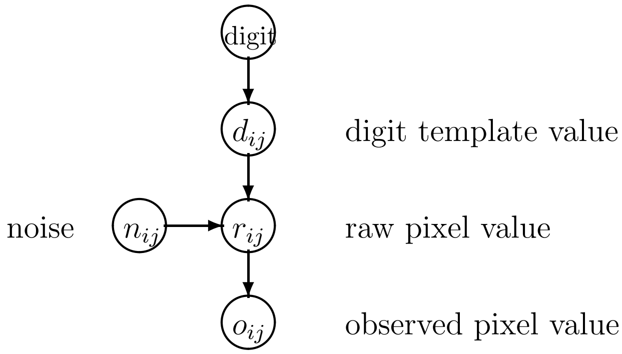
Background: In this assignment digit images will be represented by 8×8 arrays of greyscale numbers, where 1 = black, 2 = dark grey, 3 = light grey, and 4 = white. For example, the following matrix represents the corresponding image.

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 4 & 4 & 4 \\ 1 & 1 & 2 & 3 & 4 & 2 & 2 & 2 \\ 1 & 1 & 3 & 4 & 2 & 1 & 1 & 1 \\ 1 & 2 & 4 & 4 & 4 & 4 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 2 & 4 & 2 \\ 2 & 1 & 1 & 1 & 2 & 3 & 4 & 2 \\ 4 & 3 & 2 & 2 & 3 & 4 & 2 & 1 \\ 2 & 4 & 4 & 4 & 4 & 2 & 1 & 1 \end{bmatrix}$$



The goal is to implement three probability models that can be used to generate and recognize images of handwritten digits. The assignment is split up into three main parts: in each part you will implement a generator and recognizer for a different probability model (where each model extends the previous).

Probability Model 1: Generation



Pick a digit n uniformly at random from among 1,...,9,0 and retrieve its corresponding image template D (an 8×8 array). Given the digit template, each image pixel o_{ij} is generated independently by first picking a random noise value $n_{ij} \in \{-3, -2, -1, 0, 1, 2, 3\}$ (according to the distribution defined in `noise`), adding the noise to the corresponding digit template pixel d_{ij} to obtain the raw pixel value, $r_{ij} = d_{ij} + n_{ij}$, and then thresholding the raw value to obtain the observed image pixel:

$$o_{ij} = \begin{cases} r_{ij} & \text{if} \quad 1 \le r_{ij} \le 4 \\ 1 & \text{if} \quad r_{ij} < 1 \\ 4 & \text{if} \quad r_{ij} > 4 \end{cases}$$

The `noise` vector gives the probability of obtaining each noise value: `noise(1)`$=$ P$(n_{ij} = -3)$, `noise(2)`$=$ P$(n_{ij} = -2)$, ..., `noise(7)`$=$ P$(n_{ij} = 3)$.

Part 1 (Generating from Model 1—1%)

Write a Matlab function "`gen1`" that takes two arguments, `digits` and `noise`, and returns a random image of a digit generated by the model outlined above.

Probability Model 1: Recognition

Given an observed image O, we wish to compute the most likely digit given the image. This amounts to computing the most likely digit image template D given O (since digits and their image templates are in a one-to-one correspondence). Thus, we wish to find

$$\begin{aligned} D^* &= \arg\max_D \; \mathrm{P}(D|O) \\ &= \arg\max_D \; \mathrm{P}(DO) \\ &= \arg\max_D \; \sum_N \mathrm{P}(DNO) \end{aligned}$$

$$
\begin{aligned}
&= \arg\max_D \sum_N \mathrm{P}(D)\mathrm{P}(N)\mathrm{P}(O|DN) \\
&= \arg\max_D \; \mathrm{P}(D) \sum_{n_{11}=-3}^{3} \cdots \sum_{n_{88}=-3}^{3} \left[\prod_{i=1}^{8}\prod_{j=1}^{8} \mathrm{P}(n_{ij})\right] \left[\prod_{i=1}^{8}\prod_{j=1}^{8} \mathrm{P}(o_{ij}|d_{ij}, n_{ij})\right] \\
&= \arg\max_D \; \mathrm{P}(D) \prod_{i=1}^{8}\prod_{j=1}^{8} \sum_{n_{ij}=-3}^{3} \mathrm{P}(n_{ij})\mathrm{P}(o_{ij}|d_{ij}, n_{ij})
\end{aligned}
$$

where

$$
\mathrm{P}(o_{ij}|d_{ij}, n_{ij}) = \begin{cases} 1 & \text{if } d_{ij} + n_{ij} = o_{ij} \\ & \text{or } o_{ij} = 1 \text{ and } d_{ij} + n_{ij} < 1 \\ & \text{or } o_{ij} = 4 \text{ and } d_{ij} + n_{ij} > 4 \\ & \\ 0 & \text{otherwise} \end{cases}
$$

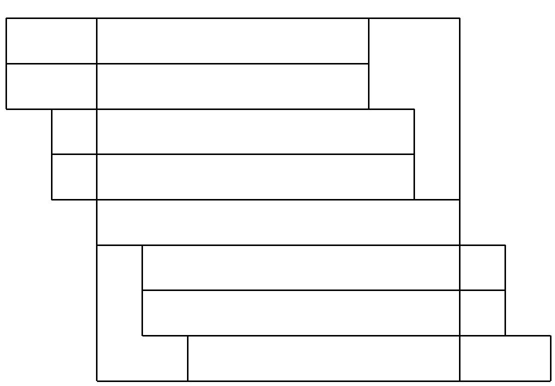
and $P(D)$ is uniform. For each digit image template, $D^{(1)}, ..., D^{(9)}, D^{(0)}$, we can then compute the conditional probability of $D^{(d)}$ given O by $\mathrm{P}(D^{(d)}|O) = \mathrm{P}(D^{(d)}O)/\sum_{d=1,...9,0} \mathrm{P}(D^{(d)}O)$.

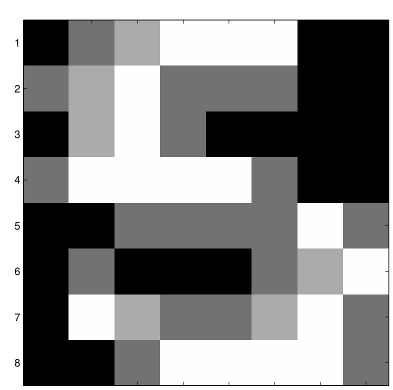
Part 2 (Recognition with Model 1—2%)

Write a Matlab function "`rec1`" that takes three arguments, `obs`, `digits`, and `noise`, where `obs` is an 8×8 image of an observed digit, and returns `class` and `probs`, where `class` is the digit 1,...,9,0 that is most likely given `obs`, and `probs` is a 1×10 array of the conditional probabilities of each digit given `obs`. The classification and conditional probabilities are to be calculated using the procedures outlined above. (Note that the conditional probabilities may be near zero or one.)

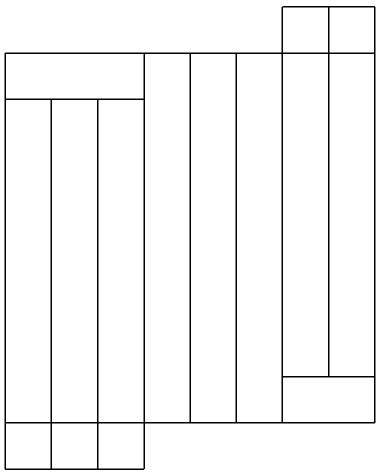
Probability Model 2: Generation

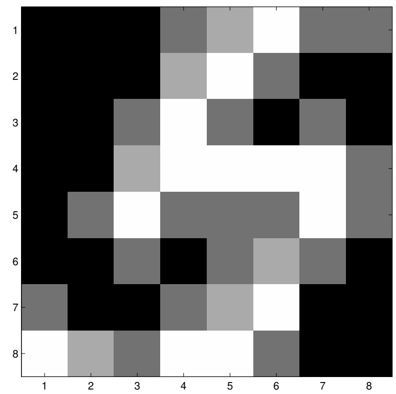
The second probability model extends the first by adding a possible transformation to the digit templates before generating the final image. We will consider a small set of simple horizontal and vertical shear transformations. In particular, horizontal shear transformations will slide the top part of the image template to the left and the bottom part to the right. For example, a horizontal shear value of $h = 2$ will shift the template as follows:



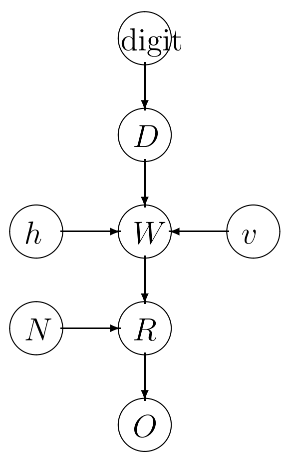


Similarly, a vertical shear transformation will slide the left part of the image in an opposite direction to the right part. For example, a vertical shear value of $v = -1$ will shift the template as follows:





Generation from this model is similar to before, except that we introduce an intermediate transformation step which takes into account both the horizontal and vertical shears.



To generate an image, pick a digit n uniformly at random and retrieve its corresponding image template D. Then independently pick a random horizontal shear number $h \in \{0, 1, 2\}$ (according to the distribution defined in `htrans`), and a random vertical shear number $v \in \{-1, 0, 1\}$ (according to the distribution defined in `vtrans`). Then, given D, h and v, a warped template W is created according to

$$w_{ij} = \begin{cases} d_{\,t(j,i,v)\; t(i,j,h)} & \text{if both } 1 \leq t(j,i,v) \leq 8 \text{ and } 1 \leq t(i,j,h) \leq 8 \\ 1 & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} t(j,i,v) &= i + round(v(5-j)/4) \\ t(i,j,h) &= j + round(h(5-i)/4) \end{aligned}$$

Note that the `htrans` vector gives the probability of obtaining each horizontal shear value: `htrans(1)`$=$ $\mathrm{P}(h=0)$, `htrans(2)`$=$ $\mathrm{P}(h=1)$, `htrans(3)`$=$ $\mathrm{P}(h=2)$; and the `vtrans` vector gives the probability of obtaining each vertical shear value: `vtrans(1)`$=\mathrm{P}(v=-1)$, `vtrans(2)`$=\mathrm{P}(v=0)$, `vtrans(3)`$=\mathrm{P}(v=1)$.

Finally, given the warped template, each image pixel o_{ij} is generated independently as before by first picking a random noise value n_{ij} and adding it to w_{ij} to obtain a raw pixel value $r_{ij} = w_{ij} + n_{ij}$, and then thresholding between 1 and 4 to obtain the final observed value.

Part 3 (Generating from Model 2—1%)

Write a Matlab function "`gen2`" that takes four arguments, `digits`, `noise`, `htrans` and `vtrans`, and returns a random image of a digit generated by the model outlined above.

Probability Model 2: Recognition

$$
\begin{aligned}
D^* &= \arg\max_D \ \mathrm{P}(DO) \\
&= \arg\max_D \ \sum_h \sum_v \sum_N \mathrm{P}(D)\mathrm{P}(h)\mathrm{P}(v)\mathrm{P}(N)\mathrm{P}(O|D,h,v,N) \\
&= \arg\max_D \ \mathrm{P}(D) \sum_h \mathrm{P}(h) \sum_v \mathrm{P}(v) \sum_{n_{11}} \cdots \sum_{n_{88}} \left[\prod_i \prod_j \mathrm{P}(n_{ij})\right] \left[\prod_i \prod_j \mathrm{P}(o_{ij}|d_{\ t(j,i,v)\ t(i,j,h)}\ , n_{ij})\right] \\
&= \arg\max_D \ \mathrm{P}(D) \sum_h \mathrm{P}(h) \sum_v \mathrm{P}(v) \prod_i \prod_j \sum_{n_{ij}} \mathrm{P}(n_{ij})\mathrm{P}(o_{ij}|d_{\ t(j,i,v)\ t(i,j,h)}\ , n_{ij})
\end{aligned}
$$

where $\mathrm{P}(o_{ij}|d_{\ t(j,i,v)\ t(i,j,h)}\ , n_{ij})$ is determined as before.

Part 4 (Recognition with Model 2—2%)

Write a Matlab function "`rec2`" that takes five arguments, `obs`, `digits`, `noise`, `htrans` and `vtrans`, and returns `class` and `probs`, where `class` is the digit 1,...,9,0 that is most likely given `obs`, and `probs` is a 1×10 array of the conditional probabilities of each digit given `obs`. The classification and conditional probabilities are to be calculated using the procedures outlined above.

Probability Model 3: Your own improved model

Design a probability model that extends either Model 1 or Model 2, and achieves better recognition performance (than both models) on the test examples in `data`. Note that the 1100 images in `data` forms a very large set and it would probably take too long to test your model on every image. Instead, you can (repeatedly) test the model on 25 randomly chosen images for each digit. (The TA will apply a similar test to verify that your model is indeed an improvement.)

Part 5 (Derivation of Model 3—1%)

Describe your extended probability model. Explain how the data can be generated from your model, and derive the formulas needed for recognition and for calculating the conditional probabilities of digits given images in your model. Please typeset your answer neatly (in whatever typesetting tool you prefer), and convert it to PDF format for submission. Submit the brief write-up in a file called note3.pdf.

Part 6 (Generation with Model 3—1%)

Write a Matlab function "`gen3`" that takes `digits`, `noise` and some other parameters, and returns a random image generated by your extended model.

Part 7 (Recognition with Model 3—2%)

Write a Matlab function "`rec3`" that takes `obs`, `digits`, `noise` and some other parameters, and returns `class` and `probs`, where `class` is the digit 1,...,9,0 that is most likely given `obs`, and `probs` is a 1×10 array of the conditional probabilities of each digit given `obs`.