MTAT.03.227 Machine Learning (Spring 2012)
Excercise session: Learning a Bernoulli mixture model with EM

Meelis Kull
Exercise session: April 17, 2012
Homework deadline: April 24, 2012 at 12:15 EET

Important notes:
1) For homework provide a report in the PDF-format. The report should include R code and explanatory figures, formulas and explanatory text.
2) The goal of this homework is to build a classifier of handwritten digits.
3) The code of every programming task must be accompanied with a small test code showing that the code works correctly at least on one data set.
4) The points for tasks sum up to 11, but homework results above 10 will be awarded 10 points.

Task 1 (0.5 points): Download data files http://biit.cs.ut.ee/~kull/600digits.tsv and http://biit.cs.ut.ee/~kull/3000digits.tsv which contain respectively 600 and 3000 first digits 2, 3 and 4 in the MNIST dataset of handwritten digits (original version: http://yann.lecun.com/exdb/mnist/). Each row of these files corresponds to a single handwritten digit and contains a list of integers separated by tabulators. The integer in the first column specifies whether the represented handwritten digit is 2, 3 or 4. The remaining 784 integers in a row represent a $28 \times 28$ bitmap image of the digit. The integers range from 0 to 255 and encode the greyscale colors. The background of the digit has color 0, and the handwritten digit is written with color 255 and rendered with anti-aliasing.

Download the R code http://biit.cs.ut.ee/~kull/digits.R. This code helps you to input the handwritten digits to R, convert the greyscale codes to the range from 0 to 1 and draw the bitmap images. Read the code, understand what it does, and run it.

Using the function draw_digits create two figures (one with interpolate=T and one with interpolate=F) with images of the 9 first digits of the data. Report these figures in the homework.

Task 2 (0.5 points): Sharpen all the digit images into black-and-white images with no grey. For this create sharp600 and sharp3000 calculated from shades600 and shades3000 by fixing a threshold on 0.5 and setting all values above or below this value to 1 or 0, respectively. Include a figure of 9 first sharpened digits in the homework.

Task 3 (0.5 points): The goal of tasks 3–10 is to build a Bernoulli mixture model of the handwritten digits. The mixture will have three components, and in the case of successful learning these will be generating bitmaps resembling the digits 2, 3 and 4.

Bernoulli distribution is a probability distribution over two possible outcomes: 0 or 1. It is characterized by a single parameter $\mu$, which is the probability of the outcome 1. In the context of handwritten digits we refer to each pixel in the image with $28 \times 28$ pixels as an outcome of some random variable with Bernoulli distribution. This is why we created the sharpened variant of the picture with pixel values only 0 and 1.

We assume the Bernoulli random variables corresponding to different pixels to be independent. Although the image is two-dimensional, we can create a one-dimensional vector with $28 \cdot 28 = 784$
For example, at positions 1...28 we can hold pixels of the topmost row, at positions 29...56 the next row, etc. Thus, we can treat each image as an outcome of a random vector with 784 random variables, each of which is Bernoulli distributed and independent of the others. This random vector has 784 parameters, which can be gathered together as a parameter vector \( \mu = (\mu_1, \mu_2, \ldots, \mu_{784}) \).

Write a function to generate an outcome of a Bernoulli random vector with given parameter vector \( \mu \):

```r
generate_from_bernoulli = function(mu)
```

Test the function by creating a pixel vector from a model with parameter vector \( \mu = (0.5, 0.5, \ldots, 0.5) \) of length 784. Visualize four pixel vectors obtained this way with `draw_digits`.

**Task 4 (0.5 points):** Define a function which produces randomly a parameter vector \( \mu \) of length 784, where each of the elements is obtained uniformly randomly from the interval \([a, b]\).

```r
random_mu = function(a,b)
```

Create three parameter vectors: `mu1` and `mu2` with `random_mu(0.25, 0.75)` and `mu3` with `random_mu(0, 1)`. Draw the three obtained parameter vectors with `draw_digits` (can be done because `draw_digits` accepts any vectors of 784 values from the range \([0, 1]\)) and report in the homework. Use `generate_from_bernoulli` to obtain two outcomes for each of these parameter vectors (six in total). Draw these also and report. NB! Use `draw_digits` with `interpolate=F` to see clear borders of pixels. Please save the models and outcomes in variables for use in the next task.

**Task 5 (1 point):** Define a function to calculate log-probability (natural logarithm of the probability) that a Bernoulli model with given parameter vector \( \mu = (\mu_1, \mu_2, \ldots, \mu_{784}) \) is generating a given outcome \( x = (x_1, x_2, \ldots, x_{784}) \), where each \( x_i \) is either 0 or 1.

```r
logprob = function(x,mu)
```

As the elements of the vector are independent, the probability can be obtained by multiplying the individual probabilities. And thus, the log-probability can be obtained by summing up the individual log-probabilities:

$$
\ln p(x|\mu) = \sum_{i=1}^{784} \ln p(x_i|\mu_i)
$$

where \( p(1|\mu_i) = \mu_i \) and \( p(0|\mu_i) = 1 - \mu_i \).

Using data from the previous task, calculate the log-probabilities of the six outcomes with respect to each of the three Bernoulli models. Is the result as expected? Why?

**Task 6 (1 point):** Suppose now that an outcome \( x \) is obtained from the mixture model of three Bernoulli models \( \mu_1, \mu_2, \mu_3 \) with mixing proportions \( \pi_1, \pi_2, \pi_3 \). Write a function to calculate the responsibilities of \( x \) with respect to the three components of the mixture model.

```r
outcome.responsibilities = function(x,mu1,mu2,mu3,p1,p2,p3)
```

For this you first need to calculate the log-probabilities of \( x \) given \( \mu_1 \), \( x \) given \( \mu_2 \) and \( x \) given \( \mu_3 \). The responsibility of \( x \) with respect to \( \mu^{(k)} \) can be calculated for \( k = 1, 2, 3 \) using the following
formula:

\[ \frac{\pi_k p(x|\mu^{(k)})}{\pi_1 p(x|\mu^{(1)}) + \pi_2 p(x|\mu^{(2)}) + \pi_3 p(x|\mu^{(3)})} \]

Note that the formula involves probabilities, not log-probabilities, and thus you need to exponentiate the log-probabilities. A problem that can arise in practice is that with exponentiation all the three log-probabilities can turn into zeros, causing the denominator to become zero. To avoid this, you should add the same positive constant to all three log-probabilities before exponentiation.

Calculate the responsibilities of each of the six outcomes with respect to the Bernoulli mixture model of the three Bernoulli models of Task 4 with equal proportions \((1/3)\). What can you say about these numbers?

**Task 7 (0.5 points):** Make a function to calculate responsibilities for each of the handwritten digits in a given list (sharp600 or sharp3000), given values of \(\mu\) and \(\pi\).

```r
responsibilities = function(data,mu1,mu2,mu3,pi1,pi2,pi3)
```

Report the results for the first 3 digits with respect to the Bernoulli mixture model of the three Bernoulli models of Task 4 with equal proportions \((1/3)\).

**Task 8 (0.5 points):** Make a function to find the index (1, 2 or 3) of the most likely mixture component for a given outcome.

```r
mlcomponent = function(x,mu1,mu2,mu3,pi1,pi2,pi3)
```

The most likely mixture component is the one with highest responsibility.

Find the most likely mixture components for the first 3 digits in the Bernoulli mixture model of the three Bernoulli models of Task 4 with equal proportions \((1/3)\).

**Task 9 (1 point):** The following update rules have been derived from E- and M-step formulas of the EM-algorithm:

\[
\begin{align*}
N_k &= \sum_{n=1}^{N} \gamma(z_{nk}) \\
\pi_k &= \frac{N_k}{N} \\
\gamma(z_{nk}) &= \frac{\pi_k p(x_n|\mu^{(k)})}{\pi_1 p(x_n|\mu^{(1)}) + \pi_2 p(x_n|\mu^{(2)}) + \pi_3 p(x_n|\mu^{(3)})} \\
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
\end{align*}
\]

Which of these formulas need to be used in the E-step and which in the M-step? Hint: the formulas are analogous to the Gaussian mixture model case.

**Task 10 (1 point):** Implement the EM algorithm to find the most likely three-component 784-dimensional Bernoulli mixture model given the data.

```r
em.bmm <- function(data,epsilon=1e-10,max.iter=100) {
```

iter = 0
repeat {
    iter = iter + 1
    if (iter>max.iter) break
} 
result = list(iter,mu1,mu2,mu3,pi1,pi2,pi3)
names(result) = c("iter","mu1","mu2","mu3","pi1","pi2","pi3")
result

The algorithm needs to iterate the update rules given in the previous task in the appropriate order until convergence, e.g. until the parameter vectors \( \mu \) change less than a given \( \epsilon \), or until the maximum number of allowed iterations is exceeded. You should initialize \( \pi = (1/3,1/3,1/3) \) and \( \mu_1, \mu_2, \mu_3 \) each with \( \text{random}_\mu(0.25,0.75) \).

Apply \( \text{em.bmm} \) on the data \( \text{sharp600} \) and draw the resulting parameter vectors \( \mu \). Generate an outcome from each of the component distributions of the mixture using the function \( \text{generate_from_bernoulli} \) and draw the results. How would you comment the results? Please save the results for the next tasks.

**Task 11 (1 point):** Suppose we have decided to attach labels 2,3,4 (possibly in some other order) to the components of the BMM. Given a digit it is now possible to find its most likely generating component with \( \text{mlcomponent} \) and use its label to classify the digit. Define a function returning labels for all given handwritten digits (data):

\[
\text{labels} = \text{function}(\text{data}, \text{labelling}, \mu_1, \mu_2, \mu_3, \pi_1, \pi_2, \pi_3)
\]

Create the labelling for \( \text{sharp600} \) based on the results of \( \text{em.bmm} \) and the labels obtained by visual inspection of the results of \( \text{em.bmm} \). Report the number of digits for which this classification was erroneous.

**Task 12 (1 point):** Define a function which tests all possible labellings and finds the one with the least number of classification errors among the first 10 handwritten digits. The function should report the resulting labels for all data.

\[
\text{classify} = \text{function}(\text{data}, \text{first10labels}, \mu_1, \mu_2, \mu_3, \pi_1, \pi_2, \pi_3)
\]

**Task 13 (1 point):** Define a function to evaluate the quality of classification. The function should count the number of correctly and incorrectly classified digits and also produce a confusion matrix, which tells how many digits 2 or 3 or 4 were classified into class 2 or class 3 or class 4.

\[
\text{evaluate} = \text{function}(\text{true_digits}, \text{predicted_digits})
\]

Classify the \( \text{sharp600} \) dataset with \( \text{classify} \). Report the quality of classification.

**Task 14 (1 point):** Compare the quality of classification for different runs of \( \text{classify} \) on the two dataset \( \text{sharp600} \) and \( \text{sharp3000} \). What is the average quality of classification over several runs? Is the variance of quality high? Are the results on \( \text{sharp3000} \) better than on \( \text{sharp600} \)?