1. Naive Bayes classifier

Create a Naive Bayes classifier for each handwritten digit that support discrete and continuous features

INPUT:

1. Training image data from MNIST (http://yann.lecun.com/exdb/mnist/)

Pleaes read the description in the link to understand the format.

Basically, each image is represented by 28X28X8bits (the header is in big endian format; you need to deal with it), you can use a char arrary to store an image.

There are some headers you need to deal with as well, please read the link for more details.

TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

[offset]	[type]	[value]	[description]
0000	32 bit integer	0x00000803(2051)	magic number
0004	32 bit integer	60000	number of images
0008	32 bit integer	28	number of rows
0012	32 bit integer	28	number of columns
0016	unsigned byte	??	pixel
0017	unsigned byte	??	pixel
....			
xxxx	unsigned byte	??	pixel

TRAINING SET LABEL FILE (train-labels-idx1-ubyte):

[offset]	[type]	[value]	[description]
0000	32 bit integer	0x00000801(2049)	magic number (MSB first)
0004	32 bit integer	60000	number of items
0008	unsigned byte	??	label
0009	unsigned byte	??	label
....			
xxxx	unsigned byte	??	label

The labels values are 0 to 9.

2. Training lable data from MNIST.
3. Testing image from MNIST
4. Testing label from MNIST
5. Toggle option: 0 for discrete mode, 1 for continuous mode.

OUTPUT:

Print out the the posterior (in log scale to avoid underflow) of the ten categories (0-9) for each row in INPUT 3 (your prediction is the category having the highest posterior), and tally the number of correct prediction by comparing with INPUT4. Calculate and report the error rate in the end.

FUNCTION:

In discrete mode:

Tally the frequency of the values of each pixel into 32 bins. Perform Naive Bayes classifer.

Note that to avoid empty bin, you can use a peudocount (such as the minimum value in other bins) for instead.

In Continuous mode: Use MLE to fit a Gaussian distribution for the value of each pixel. Perform Naive Bayes classifer.

2. Online learning

Use online learning to learn the beta distribution of the parameter p (chance to see 1) of the coin tossing trails in batch.

INPUT:

1. A file contains many lines of binary outcomes:
   0101010111011011010101
   0110101
   010110101101
   ...
2. parameter a for the initial beta prior
3. parameter b for the initial beta prior

OUTPUT:

Print out the Binomial likelihood by MLE, Beta prior and posterior probability for each line.

FUNCTION:

Use Beta-Binomial conjugation to perform online learning.