Programming Homework 6

CS540

July 19, 2019

1 Instruction

Please submit your output files and code on Canvas → Assignments → P6. Please do not put code into zip files and do not submit data files. The homework can be submitted within 2 weeks after the due date on Canvas without penalty (50 percent penalty after that).
Please add a file named ”comments.txt”, and in the file, you must include the instructions on how to generate the output, for example:

- Data files required: train.csv, test.csv. Run: main.jar.
- Data folder required: data/train1.png ... data/train100.png . Compile and Run: main.java.

2 Details

All the requirements are listed on the course website. The following is only an example workflow to solve the problem.

1. Download either the ndjson or the bin file. The original data contains x and y values of each stroke, they should be concatenated into one single array. Either:

$$(x_{1,1}, x_{1,2}, ..., x_{1,m_1}, x_{2,1}, x_{2,2}, ..., x_{2,m_2}, ..., x_{M,1}, x_{M,2}, ..., x_{M,m_M})$$
$$(y_{1,1}, y_{1,2}, ..., y_{1,m_1}, y_{2,1}, y_{2,2}, ..., y_{2,m_2}, ..., y_{M,1}, y_{M,2}, ..., y_{M,m_M})$$

OR (the previous version is preferred for step 2):

$$(x_{1,1}, y_{1,1}, ..., x_{1,m_1}, y_{1,m_1}, ..., x_{M,1}, y_{M,2}, ..., x_{M,m_M}, y_{M,m_M})$$

Here, there are M strokes and stroke j has m_j points sampled from it.

2. Fix m as the feature length. For example, start with $m = 50$ and try different values based on the final clusters. Sample m pairs of (x, y) values from each image. Given an array $(x_1, x_2, ..., x_t)$, to downsample (in the case $t > m$):

$$x'_i = x_{\text{round}\left(i\frac{t-1}{m-1}\right)}, i = 0, 1, ..., m-1$$

Given an array $(x_1, x_2, ..., x_t)$, to upsample (in the case $t < m$), there are many ways, the simplest way is to use linear interpolation:

$$x'_i = \left(1 - \frac{i \text{ mod } q}{q}\right) x_{\text{floor}\left(\frac{i}{q}\right)} + \frac{i \text{ mod } q}{q} x_{\text{floor}\left(\frac{i}{q}\right)+1}, i = 0, 1, ..., m-2$$

$$x'_{m-1} = x_{t-1}$$

Here, $q = \dfrac{m-1}{t-1}$ should be an integer, if it is not an integer, upsample to $\text{ ceiling }(q)(t-1)+1$ and then downsample to m.
Add: important: you should do the above interpolation for x and y separately.
For $(y_1, y_2, ..., y_t)$,

$$y'_i = y_{\text{round}\left(i\frac{t-1}{m-1}\right)}, i = 0, 1, ..., m-1, \text{ if } t > m$$

$$y'_i = \left(1 - \frac{i \text{ mod } q}{q}\right) y_{\text{floor}\left(\frac{i}{q}\right)} + \frac{i \text{ mod } q}{q} y_{\text{floor}\left(\frac{i}{q}\right)+1}, i = 0, 1, ..., m-2, \text{ with } y'_{m-1} = x_{t-1}, \text{ if } t < m$$

3. Randomly generate K vectors of length m with numbers between the minimum and maximum x and y values. The minimum and maximum x and y values should be 0 and 255, but please check to make sure.

4. For each of the n images, say image i, compute the Euclidean distance from the image to each of the K cluster centers and find the cluster center that is the closest to the image, say cluster $k^\star$. Label image i as in cluster $k^\star$, that is, $\hat{y}_i = k^\star$.

$$\hat{y}_i = \arg\min_k \|x_i - c_k\|_2^2$$

5. For each cluster, say k, recompute the center by averaging all images that are labelled k. Here, n_k is the number of instances that are labelled k.

$$c_k = \frac{1}{n_k} \sum_{i=1}^{n} x_i \mathbb{1}_{\{\hat{y}_i = k\}}$$

$$n_k = \sum_{i=1}^{n} \mathbb{1}_{\{\hat{y}_i = k\}}$$

6. Repeat until c_k is not changing.