Intelligent Visual Computing [Spring 2018]

Assignment 1 (warm-up): Shape Classification

Overview

In this assignment you will learn to implement a simple shape classification scheme based on logistic regression and a shape descriptor based on histograms of surface point locations. This assignment counts for **10 points** towards your final grade.

Instructions

The starter code for this assignment is written in Matlab. Download the code here:
starter_code.zip
The code has three matlab scripts that you have to complete with a few lines of code each: computeShapeHistogram.m, LogisticRegressionTrain.m, and LogisticRegressionTest.m.

The zip file also contains a dataset of 42 3D meshes in obj format. The dataset is split into a training set of 20 meshes (folder 'train') where you will train your classifier, and a test of 22 meshes (folder 'test') where you will test your classifier. The meshes represent 3D objects of chairs and tables. All meshes are approximately oriented such that gravity ('upright' axis) is the y-axis. The folders 'train' and 'test' contain a txt file (ground_truth_labels.txt) that contains the category id for each mesh (0 for chairs, 1 for tables). The zip file includes a small library of functions (mesh_toolbox*.m) that load obj/off files and display meshes. Each loaded mesh contains the 3D locations of mesh vertices (3xV matrix, where V is the number of its vertices), and mesh triangles (3xF matrix, where F is the nuimber of faces, each face contains indices to three vertices). You do not need to edit the code of these mesh toolbox functions. You will only insert code in computeShapeHistogram.m, LogisticRegressionTrain.m, and LogisticRegressionTest.m.

What You Need To Do (10 points in total)

Task A [2 points]: Complete the script computeShapeHistogram.m so that it computes a histogram capturing the distribution

of surface point locations along the upright axis (y-axis) in a given range [y_min, y_max] for an input mesh. The range [y_min, y_max] should be divided into K uniform bins (K is input parameter, called 'number_of_bins' in the code). Each histogram bin counts the percentage of 3D points with y-coordinate (second coordinate) falling within the bin interval. The histogram should be normalized i.e., represent a discrete probability distribution.

Task B [4 points]: Complete the logisticRegressionTrain.m script so that it trains a logistic regression classifier that classifies shapes into chairs or tables based on your implemented histogram descriptor. The code reads the consistently scales and centers the input meshes, reads their ground-truth labels, computes the range [y_min, y_max], and calls the computeShapeHistogram.m script per mesh for you. Use 10 bins for your histogram descriptor. Train your logistic regression classifier such that the cross-entropy cross function is minimized with gradient descent (step rate 0.1), 10000 iterations, including L1-norm regularization (set the regularization parameter to 0.1). Include in your report the learned parameters (vector w) returned by your completed script. The script should be executed as follows:
[w, min_y, max_y] = logisticRegressionTrain('dataset/train', 10);

Task C [2 points]: Complete the logisticRegressionTest.m script so that it applies your logistic regression classifier with learned parameters 'w' in the test dataset. The code consistently scales and centers the input meshes and reads the ground-truth labels, which are necessary here to evaluate the classifier. It also calls the computeShapeHistogram.m script per mesh. Apply the logistic regression classifier to compute the probability of table per mesh based on its input descriptor. Compute also the classification accuracy in your test dataset. Include in your report the output probabilities (vector t), and classification accuracy (variable 'test_err') returned by your completed script. The script should be executed as follows:
[t, test_err] = logisticRegressionTest('dataset/test', w, min_y, max_y)

Task D [1 point]: Include in your report the learned parameters, output probabilities and classification accuracy in your test dataset when you train your classifier without regularization (but when you submit your code, please -include- the regularization).

Task E [1 point]: Include in your report your **derivation** of gradient you used for training (gradient of cross-entropy including L1-norm regularization).

Submission

Please follow the **Submission** instructions in the course policy to upload your zip file to Moodle. The zip file should contain all the Matlab code and a short PDF report. Make sure your program runs in Matlab.

- starter_code.zip

Grading summary

Participants	60

Moodle at UMass Amherst

Get Help

Submitted	0
Needs grading	0
Due date	Friday, February 2, 2018, 11:55 PM
Time remaining	6 days 4 hours

View/grade all submissions

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Friday, February 2, 2018, 11:55 PM
Time remaining	6 days 4 hours
Last modified	-
Submission comments	Comments (0)

Add submission

Make changes to your submission