Coursera:Univ of Washington:Predictive Analytics

bdanalytics: Jan 15, 2016

Kaggle Competition: Facial Keypoints Detection

Problem Description:

The task is to automatically locate specific key-points (e.g. left eye center, nose tip, mouth right corner, etc.) on facial images. Training dataset contains 7,049 grayscale facial images (96 x 96 pixels) with xy co-ordinates of 15 key-points. Test dataset contains 1,783 images without any key-points location(s). Results are evaluated based on distance of predictions from pre-computed locations (not available in the data). Lower score indicates closer predictions. The competition description does not disclose methodology of the evaluation metric (or maybe I missed it)

Analysis Approach:

I followed the suggestion provided by the competition organizers of extracting a patch (21 x 21) of pixels around each keypoint and computing their mean pixel pattern. The mean pixel pattern is then compared against a query patch. Comparison metrics include correlation and Minkowski similarities (Minkowski L1 distance is identical to Manhattan distance, Minkowski L2 distance is identical to Euclidan distance, etc.).

Would like to investigate the option of using external pre-trained classifiers (e.g. opencv haartraining).

Initial Solution:

My implementation was in R utilizing ggplot2 (image plotting), proxy (matrix similarity computations) & caret (cross-validation & algorithms). Initial solution once again was plagiarized from the competition notes which involved computing the mean location of all keypoints across the training images. This was used as the prediction for all test images which resulted in a score of 3.9654.

Challenges included figuring out how to create ggplots for image matrices (geom_raster on the reversed image vector) & overlaying the actual locations & query locations (used different shapes & colors for actual vs. query)

Revised Solution(s):

Initially I just focused on left_eye_center detection due to time & computing throughput constraints. The predictions for all other key-points defaulted to the training mean locations.

1. Extract mean pixel patch (21 x 21) around the keypoint location in the training data set
2. For each training image:
    - Extract pixel patches for each center in a 5 x 5 grid around the keypoint location
    - Compute similarity metrics for these 25 patches with the keypoint mean patch
        - correlation after transforming the patch matrix into a single vector
        - Minkowski distance (L1, L2 & L3) for matrices
3. Output patch center & associated metrics from all training images with a label as “training” input to a classifier
        - label is “left_eye_center” for keypoint location
        - label is “none” for all other patch centers
4. For each test image:

    - Extract pixel patches for each center in a 5 x 5 grid around the keypoint mean location
    - Compute similarity metrics

5. Output patch center & associated metrics from all test images as “test” input to a classifier
6. Train a GLMnet classifier on training patches
7. Predict labels for test patches
        - Utilizing correlation resulted in a score of 3.93942 (0.66 %) improvement
            - not worth all that work !!!
            - but learnt a lot