521285S Affective Computing (Laboratory Exercises)

Exercise – III Multi-modal Emotion Recognition

Objective

Your task is to use the feature-level method to combine the facial expression features and audio features. A multi-modal emotion recognition system is constructed to recognize happy versus sadness facial expressions (binary-class problem) by using a classifier training and testing structure.

The original data is based on lab1 and lab2, from ten actors acting happy and sadness behaviors.

- Task 1: Subspace-based feature fusion method: In this case, z-score normalization is utilized. Please read “Fusing Gabor and LBP feature sets for kernel-based face recognition” and learn how to use subspace-based feature fusion method for multi-modal system.
- Task 2: Based on Task1, use Canonical Correlation Analysis to calculate the correlation coefficient of facial expression and audio features. Finally, use CCA to build a multi-modal emotion recognition system. The method is described in one conference paper “Feature fusion method based on canonical correlation analysis and handwritten character recognition”
- Optional task: Use feature-level method (Task 2) on 10-fold cross-validation estimate of the emotion recognition system performance

To produce emotion recognition case, Support Vector Machine (SVM) classifiers are trained. 50 videos from 5 participants are used to train the emotion recognition, use spatiotemporal features. The rest of the data (50 videos) is used to evaluate the performances of the trained recognition systems.

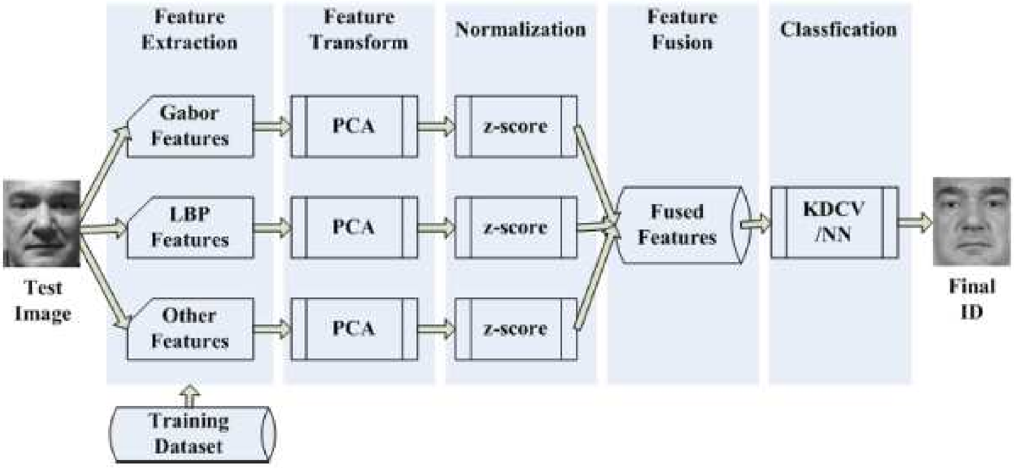
Implementation

The data and toolbox files used in this exercise can be found in the Affective Computing course webpage (see the Noppa system).

Download the data file ‘lab3_data.mat’ containing the facial expression features and audio features. Download also ‘PCA.m’, ‘EuDist2.m’, ‘mySVD.m’, ‘svmclassify_vr.m’ for the required MATLAB functions.

Study the PCA functions (‘PCA.m’) and generic MATLAB functions ‘svmtrain’, ‘svmclassify’, ‘confusionmat’, ‘canoncorr’, as they are needed in the exercise.

1. Subspace-based method



Please read “Fusing Gabor and LBP feature sets for kernel-based face recognition” and apply their framework for the exercise. We use Support Vector Machine (SVM) with linear kernel for classification.

Steps:

- Extract the subspace for facial expression feature and audio features using principal component analysis (PCA.m). Example to use PCA: [eigvector, eigvalue] = PCA(data, options), where data is a data matrix, in which each row vector of data is a sample. options.ReducedDim is the dimensionality of the reduced subspace.

Example:
```
            fea = rand(7, 10);
            options = [];
            options.ReducedDim = 4;
            [eigvector, eigvalue] = PCA(fea, options);% eigvector is the
subspace
            Y = fea * eigvector; % this code is to map ‘fea’ to the subspace
‘eigvector’
```

    - Set ReducedDim to 20 and 15 for facial expression feature and audio feature, respectively.
    - Construct two subspaces U1 and U2 for facial expression features (‘training_data’) and audio features (‘training_data_proso’), respectively.
- Extract low-dimensional features for training and testing features:
    - Map ‘training_data’ and ‘testing_data’ into the subspace U1, named as ‘training_data1’, ‘testing_data1’, respectively, and then use Z-score normalization for facial expression.
    - Map ‘training_data_proso’ and ‘testing_data_proso’ into the subspace U2, named as ‘training_data2’ and ‘testing_data2’, respectively, and then use Z-score normalization for audio.
        - Z-score normalization is interpreted in the following link:
            - https://docs.tibco.com/pub/spotfire/6.5.1/doc/html/norm/norm_z_score.htm

        - Functions ‘Sum’, ‘Repmat’, ‘Mean’ and ‘Std’ are used for z-score normalization.
        - Mean and standardization are computed on training set.
- Concatenate ‘training_data1’ and ‘training_data2’ into a new feature ‘combined_trainingData’
- Concatenate ‘testing_data1’ and ‘testing_data2’ into a new feature ‘combined_testingData’.
- Use the ‘svmtrain’ function to train Support Vector Machine (SVM) classifiers. Construct an SVM using the ‘combined_trainingData’ and linear kernel. The ‘training_class’ group vector contains the class of samples: 1 = happy, 2 = sadness, corresponding to the rows of the training data matrices.
- Use the ‘svmclassify’ function (and your trained SVM structures) to classify the ‘combined_trainingData’ and the ‘combined_testingData’ matrices. Then, calculate average classification performances for both training and testing data. The correct class labels corresponding with the rows of the training and testing data matrices are in the variables ‘training_class’ and ‘testing_class’, respectively.

Task 1 Show and check the results:

- ➢ Calculate recognition rate (accuracy) and confusion matrix
    - Please understand how to calculate accuracy and confusion matrix. (See https://en.wikipedia.org/wiki/Confusion_matrix)

2. Correlation-based method: Based on Task 1, use Canonical Correlation Analysis to combine multiple features.

Please read “Feature fusion method based on canonical correlation analysis and handwritten character recognition” and apply their framework for the exercise. We use Support Vector Machine (SVM) with linear kernel for classification.

Steps:

- Extract the subspace for facial expression feature and audio features using principal component analysis (PCA.m). Example to use PCA: [eigvector, eigvalue] = PCA(data, options), where data is a data matrix, in which each row vector of data is a sample. options.ReducedDim is the dimensionality of the reduced subspace.
    - Set ReducedDim to 20 and 15 for facial expression feature and audio feature, respectively.
    - Construct two subspaces U1 and U2 for facial expression features (‘training_data’) and audio features (‘training_data_proso’), respectively.
- Extract low-dimensional features for training and testing features:
    - Map ‘training_data’ and ‘testing_data’ into the subspace U1, named as ‘training_data1’, ‘testing_data1’, respectively, and then use Z-score normalization for facial expression.

    - Map 'training_data_proso' and 'testing_data_proso' into the subspace U2, named as 'training_data2' and 'testing_data2', respectively, and then use Z-score normalization for audio.
        - Z-score normalization is interpreted in the following link:
            - https://docs.tibco.com/pub/spotfire/6.5.1/doc/html/norm/norm_z_score.htm
        - Functions 'Sum', 'Repmat', 'Mean' and 'Std' are used for z-score normalization.
        - Mean and standardization are computed on training set.
- Use 'canoncorr' to construct the Canonical Projective Vector (CPV) using 'training_data1' and 'training_data2'.
    - [A, B, r] = canoncorr (X, Y), where A and B are canonical project vectors for X and Y, respectively, r is the correlation coefficient of X, Y.
- Construct Canonical Correlation Discriminant Features (CCDF)
    - U = X * A, V = Y * B
- Use the 'svmtrain' function to train Support Vector Machine (SVM) classifiers. Construct an SVM using the 'combined_trainingData' and linear kernel. The 'training_class' group vector contains the class of samples: 1 = happy, 2 = sadness, corresponding to the rows of the training data matrices.
- Use the 'svmclassify' function (and your trained SVM structures) to classify the 'combined_trainingData' and the 'combined_testingData' matrices. Then, calculate average classification performances for both training and testing data. The correct class labels corresponding with the rows of the training and testing data matrices are in the variables 'training_class' and 'testing_class', respectively.

Task 2 Show and check the results:

- ➢ Show the correlation coefficient of two feature sets
- ➢ Calculate the average classification performances and confusion matrix for CCA-based multi-modal emotion recognition.

3. Optional task: Cross-validation estimation for recognition performance (NOT REQURIED TO PASS THE EXERCISE)

Generate a person independent 10-fold cross-validation (CV) estimate of the emotion recognition system performance.

- Join the training/testing data matrices and the class vectors. Combine also the 'training_data_personID' and 'testing_data_personID' vectors that are needed to make the CV folds.
- Construct the CV folds by training ten SVMs. For each SVM nine persons' data is used as the training set (i.e. 90 samples) and one persons' samples are kept as the test set (i.e. 10 samples) for the respective fold (i.e. each SVM has different persons' samples excluded from the training set). Test each ten trained SVMs by using the corresponding one held-out persons' samples and then calculate the average classification performances for each fold.

- Calculate the mean and SD of the ten CV fold performances to produce the final CV performance estimate of the emotion recognition system.

Reference:
[1] X. Tan and B. Triggs. Fusing Gabor and LBP feature sets for kernel-based face recognition. Analysis and Modelling of Face and Gestures, pp. 235-249, 2007.
[2] Q. Sun, S. Zeng, P. Heng, and D. Xia. Feature fusion method based on canonical correlation analysis and handwritten character recognition. International Conference on Control, Automation, Robotics and Vision, pp. 1547-1552, 2004