Partial Least Squares Regression on Grassmannian Manifold for Emotion Recognition

Mengyi Liu, Ruiping Wang, Zhiwu Huang, Shiguang Shan, Xilin Chen

Institute of Computing Technology, Chinese Academy of Sciences
Outline

- Problem
- Related work
- Our Method
- Experiments
- Conclusion
Outline

• Problem
• Related work
• Our Method
• Experiments
• Conclusion
Emotion recognition in the wild

• Challenges
  – Large data variations
    • head pose, illumination, partial occlusion, etc.
  – Lack of labeled data
    • Manual annotation is hard as spontaneous expression is ambiguous in the real world.
Outline

- Problem
- Related work
- Our Method
- Experiments
- Conclusion
Video-based emotion recognition

- **Acoustic information based**
  - Time domain and frequency domain
    - e.g. pitch, intensity, pitch contour, Low Short-time Energy Ratio (LSTER), maximum bandwidth, …

- **Vision information based**
  - Spatial space and temporal space
    - e.g. Optical flow, 3D descriptor (LBP-TOP, HOG 3D), tracking based (AAM, CLM), probabilistic graph model (HMM, CRF), …
Outline

• Problem
• Related work
• Our Method
• Experiments
• Conclusion
Our method

• Key issue
  – How to model the emotion video clip?

• Motivation
  – Alleviate the effect of mis-alignment of facial images
  – Encode the data variations among video frames

• Basic idea
  – Inspired by recent progress of image set-based face recognition [1]
  – Treat the video clip as an image set, i.e., a collection of frames
  – Linear subspace for video (image set) modeling

Our method

- An overview

Preprocessing
- Original aligned face images
- Purified face images
  - Filtering out non-face in PCA subspace

Feature Designing
- Mid-level image features
- Video/Image set features
  - Subspace learning on Grassmannian manifold

Classification
- One-to-Rest PLS classification
- Video-Audio Fusion
- One-to-Rest PLS classification

Original audio data
- Clip-wise audio features extracted using openSMILE toolkit *[2]

Our method

• Preprocessing
  – Original face alignment using MoPS [3] *(provided by organizer)*
  – Purification of face images
    • Original aligned face images set: \( X = \{x_1, x_2, \ldots, x_n\}, x_i \in \mathbb{R}^D \).
    • PCA projection learned on \( X \) by preserving low energy: \( W \).
    • Mean reconstruction error of each image:
      \[
      \text{MeanErr}_t = \frac{1}{D} \| x_t - W^T W x_t \|^2
      \]
    • Non-face/Badly-aligned face images tend to have large \( \text{MeanErr}_t \).

Our method

• Preprocessing
  – The distribution of $\text{MeanErr}_t$ on training set in EmotiW2013.

* Threshold is for filtering out non-face in PCA space.
Our method

- Preprocessing
  - An example of 100 samples with largest mean reconstruction error. Most are non-face images or mis-alignment results.
Our method

• An overview

Preprocessing
- Original aligned face images
- Purified face images
  - Filtering out non-face images in PCA subspace

Feature Designing
- Mid-level image features
- Video/Image set features
  - Subspace learning on Grassmannian manifold

Classification
- One-to-Rest PLS classification
- Video-Audio Fusion
  - One-to-Rest PLS classification

Video
Audio

*Clip-wise audio features extracted using openSMILE toolkit* [2]
Our method

• Feature designing
  – Image feature [4]

Convolution Filters
6x6x100

Face Image
32x32

Filter Maps
27x27x100

Max-Pooling
3x3

Mid-level Feature
9x9x100

Our method

• Feature designing
  – Video feature
    • Each video clip is a set of images, denoted as $S_i \in R^{f \times n_i}$, where $f$ is the dimension of image feature, and $n_i$ is the number of frames.
    • The video $S_i$ can be represented as a linear subspace $P_i$, s.t.
      $$S_iS_i^T = P_i\Lambda_iP_i^T$$
    • Thus all the video clips can be modeled as a collection of subspaces, which are also the points on Grassmannian manifold.
Our method

• Feature designing
  – Video feature
    • An illustration of subspaces on Grassmannian manifold
Our method

- **Feature designing**
  - Video feature
    - The similarity between two points \( P_i \) and \( P_j \) on manifold \( M \) can be measured by a linear combination of Grassmannian kernels.
      - Projection kernel [5]: \( k_{ij}^{[\text{proj}]} = ||P_i^T P_j||_F^2 \).
      - Canonical correlation kernel [6]: \( k_{ij}^{[\text{cc}]} = \max_{a_p \in \text{span}(P_i)} \max_{b_q \in \text{span}(P_j)} a_p^T b_q \).
      - Linear combination: \( k_{ij}^{[\text{com}]} = k_{ij}^{[\text{proj}]} + \alpha k_{ij}^{[\text{cc}]} \).
  - The kernels of each point (i.e., each video) to all training points serve as its **final feature representation** for classification.

Our method

- **An overview**

  **Preprocessing**
  - Original aligned face images
  - Purified face images
  - Filtering out non-face in PCA subspace

  **Feature Designing**
  - Mid-level image features
  - Video/Image set features
  - Subspace learning on Grassmannian manifold

  **Classification**
  - One-to-Rest PLS classification
  - Video-Audio Fusion
  - One-to-Rest PLS classification

- Original audio data
- Clip-wise audio features extracted using openSMILE toolkit*[2]*
Our method

- Classification
  - Partial Least Squares (PLS) for classification [1]
    - Maximize the covariance between observations and class labels

\[ X = T \times P' \]
\[ Y = T \times B \times C' = X \times B_{pls} \]

Our method

- Classification
  - One-to-Rest PLS
    - Suppose there are $c$ categories and $N$ training samples, we train $c$ One-to-Rest PLS classifiers to predict each class simultaneously.
    - Effectively to handle the hard classes, e.g. “Sad” vs. “Disgust”

Original training label vector $Y \in R^{N \times 1}$

Binary training label matrix $Y \in R^{N \times c}$

One-to-Rest training label vectors, $y_1, y_2, \ldots, y_c \in R^{N \times 1}$
Our method

- **Classification**
  - One-to-Rest PLS

  **Training and test process**

  Training data $X$

  One-to-Rest training label vectors $y_1, y_2, ..., y_c$

  Test sample

  One-to-Rest PLS(1)
  One-to-Rest PLS(2)
  One-to-Rest PLS(3)
  ... (c-1)
  One-to-Rest PLS(c)

  Test result: $Ft \in R^{c \times 1}$
Our method

• Classification
  – Video-Audio fusion for final test output
    • For a given test video, using the \( c \) PLS classifiers for video and audio respectively, we obtain two prediction vectors \( F_{t_{\text{video}}} \), \( F_{t_{\text{audio}}} \in \mathbb{R}^{c \times 1} \).
    • We conduct a linear fusion at decision level using weighted parameter \( \lambda \)
      \[
      F_{t_{\text{fusion}}} = (1 - \lambda) F_{t_{\text{video}}} + \lambda F_{t_{\text{audio}}}.
      \]
    • The category corresponding to the maximum value in \( F_{t_{\text{fusion}}} \) is determined to be the recognition result.
Outline

• Problem
• Related work
• Our Method
• Experiments
• Conclusion
Experiments

• Discussion of Parameters
  – The fusion weights of Grassmannian kernels

\[ k_{ij}^{[\text{com}]} = k_{ij}^{[\text{proj}]} + \alpha k_{ij}^{[\text{CC}]} \]

Train-Val: @\( \alpha = 2^{-6}, 2^{-5} \)
Val-Train: @\( \alpha = 2^{-10} \)
Experiments

• Discussion of Parameters
  – The dimension of One-to-Rest PLS (video)

Train-Val: \(@dim = 10\)
Val-Train: \(@dim = 5\)
Experiments

• Discussion of Parameters
  – The dimension of One-to-Rest PLS (audio)

Train-Val: \( @dim = 5 \)
Val-Train: \( @dim = 5 \)
Experiments

• Discussion of Parameters
  – The fusion weights of video and audio modalities

\[
\text{Train-Val: } \lambda = 0.25
\]
\[
\text{Val-Train: } \lambda = 0.85
\]
## Experiments

### Results comparison

<table>
<thead>
<tr>
<th>Performance Comparison</th>
<th>Audio only</th>
<th>Video only</th>
<th>Audio + Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-to-Rest PLS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grassmannian Discriminant Analysis [6]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grassmannian Kernels + One-to-Rest PLS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision-level fusion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purified data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature-level fusion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-class LR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision-level fusion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One-to-Rest PLS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Val</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ours</strong></td>
<td>Val</td>
<td>24.49%</td>
<td>30.81%</td>
<td>32.07%</td>
<td>22.48%</td>
<td>24.24%</td>
<td>34.34%</td>
<td>35.86%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test*</td>
<td></td>
<td>24.04%</td>
<td></td>
<td></td>
<td></td>
<td>26.28%</td>
<td>33.01%</td>
<td>34.61%</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>Val</td>
<td>19.95%</td>
<td>27.27%</td>
<td></td>
<td></td>
<td></td>
<td>22.22%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>22.44%</td>
<td>22.75%</td>
<td></td>
<td></td>
<td></td>
<td>27.56%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outline

• Problem
• Related work
• Our Method
• Experiments
• Conclusion
Conclusion

• Key points of the current method
  – PCA-based **data purifying** to filter out mis-alignment faces
  – Linear subspace modeling of video data variations
  – Multiple video features fusion by **Grassmannian kernels combination**
  – **Multi-modality fusion** at decision level of video and audio

• Issues to further address
  – Exploration of **video temporal dynamics** information
  – More sophisticated **video modeling**
  – More effective fusion at **feature level**
  – …
Thank you.

Question?