Abstract

Speech recognition is a challenging problem. Therefore, considering the visual information besides the acoustic stream, is essential for improving the recognition accuracy in real-life un-constraint situations. MKPLS is shown to be successful framework for solving visual speech recognition. However, MKPLS lacks in-depth analysis to leverage its power. This work is intended to study two core building blocks in the pipeline of MKPLS: manifold parameterization and manifold kernel. For manifold parameterization, we study the effect of changing the number of centers and the regularization factor in order to find the best parameters of this step. Quantifying the similarity between visual units is challenging step for computing the kernel. Therefore, we explore three kernel categories: matrix-based, curve-based and subspace-based kernels. Intuition behind each kernel choice is provided and quantitative comparison among them is conducted. In other words, this study is intended to reason about the kernel choice for VSR. We compare different kernels using MKPLS framework. We use two public datasets: OuluVs and AVLetters databases.

1. Introduction

Audio visual speech recognition (AVSR) has been investigated intensively in the last few decades [1]. Specially after bimodal fusion of audio and visual stimuli in perceiving speech has been demonstrated by the McGurk effect [2]. For example, when the spoken sound /gæ/ is seen as /bæ/, most people perceive the sound as /dɑ/ [2]. More specifically, with the advances in computer vision, visual speech recognition (VSR), also called lipreading, have attracted research attention [3]. VSR systems gain importance with the need for controlling machines verbally. Lipreading, have attracted research attention [3]. VSR systems are essential for improving the recognition accuracy in real-life un-constraint situations. MKPLS is shown to be successful framework for solving visual speech recognition. However, MKPLS lacks in-depth analysis to leverage its power. This work is intended to study two core building blocks in the pipeline of MKPLS: manifold parameterization and manifold kernel. For manifold parameterization, we study the effect of changing the number of centers and the regularization factor in order to find the best parameters of this step. Quantifying the similarity between visual units is challenging step for computing the kernel. Therefore, we explore three kernel categories: matrix-based, curve-based and subspace-based kernels. Intuition behind each kernel choice is provided and quantitative comparison among them is conducted. In other words, this study is intended to reason about the kernel choice for VSR. We compare different kernels using MKPLS framework. We use two public datasets: OuluVs and AVLetters databases.

1 Viseme is the visual phoneme. It is defined as the smallest discriminative unit for visual speech

2. Manifold-KPLS

2.1 Problem Definition

Given set of visual units, we need to recognize new test unit, and infer the identity of the speaker. The visual unit could be viseme, word or a complete sentence. For each training visual unit is assigned to specific speech class and specific speaker.
Both the training and testing visual units are represented by sequence of images (frames) extracted from the speech video. Each frame exposes only the mouth area of the speaker.

### 2.2. Framework description

Manifold Kernel Partial Least Squares (MKPLS) framework is proposed in [13]. For convenience, we briefly describe the mathematical model and the framework pipelines here.

Let us denote the $k$-th sequence by $S_k = \{x_{ki} \in \mathbb{R}^D, i = 1 \ldots n_k\}$, where the image $x_{ki} \in \mathbb{R}^D$. Let $y_k$ represents the class labels for the $k$-th sequence. For the particular case of speech recognition and speaker identification, $y_k \in \{c_1, \ldots, c_N\} \times \{p_1, \ldots, p_l\}$. Here $c_i$ is the speech label, and $p_j$ is the performer identity. Let $M_k \subset \mathbb{R}^D$ is a low-dimensional manifold connects the images of sequence $S_k$. The basic assumption is that all these manifolds ($M_k \forall k$) are topologically equivalent, however, each of them has different geometry in $\mathbb{R}^D$. This assumption is stated clearly in [13].

In principal, MKPLS pipeline has three phases: individual manifold parameterization, latent space embedding and finally inference/classification. Consider the manifold $M_k$ connects the $n_k$ frames of specific Visual Unit (VU). Assuming, that we have a unified manifold $U$. The result of individual manifold parameterization is a single representation for VU which holds the topological deformation of $M_k$ with respect to $U$. Any further processing is done based on these parameterizations.

The manifold $M_k$ is represented by a parameterization $C_k$ with respect to a set of basis $\{\psi_1, \psi_2, \ldots \psi_n\}$. These basis are a nonlinear function of points on $U$. We use Gaussian Radial Basis Function (Gaussian-RBF): $\psi_i(z) = \exp(\sigma \|z - w_i\|)$ where $w_i, i = 1 \ldots n$ are fixed points on $U$. The goal is to find a regression function $\gamma(t) = C_k^T \Psi(t)$ which minimize the objective function

$$
\sum_{i} \left\| x_i^k - \gamma_i (z_i^k) \right\|^2 + \lambda \Omega[\gamma_i],
$$

where $\|\cdot\|$ is the Euclidean norm, $\Psi(t) = [\psi_1(z_i), \psi_2(z_i), \ldots, \psi_n(z_i)]^T$, $\Omega$ is a regularization function that enforces the smoothness in the learned function, and $\lambda$ is the regularizer. Representer theory helps to find closed form for $C$ as

$$
C_k^T = (A_k A_k + \lambda G)^{-1} A_k X_k^T,
$$

where $A_k$ is an $n_k \times n$ matrix with $A_{ij} = \exp(\sigma \|z_i - w_j\|)$ and $G$ is an $n \times n$ matrix with $G_{ij} = \exp(\sigma \|w_i - w_j\|)$. $X_k$ is the $n_k \times D$ data matrix for UV$k$. The details of learning the manifold parameterization can be found in [13].

The choice of $\lambda$ and $n$ is crucial for better performance. Figure 1 shows the trade-off between value of $\lambda$ and $n$. This choice depends upon the application. In this work, we need to capture the smooth dynamics in the visual units. Therefore, we choose $\lambda = 50$. It is clear that $n = 16$ expose more variations than with $n = 8$. More information can be useful in some cases and can be more confusing in others. Therefore, in Section 4, we show both configurations.

In Latent space embedding, kernel partial least squares (KPLS) [14] is adopted for embedding the parameterizations $\{C_k, k = 1 \ldots N\}$ into a low-dimensional latent space $\mathbb{R}^r$, as $\{t_k \in \mathbb{R}^r, k = 1 \ldots N\}$. KPLS is supervised method, so it uses the set of labels $\{y_k, k = 1 \ldots N\}$ for the embedding. Supervised embedding guarantees to achieve the most concise and informative low-dimensional latent space embedding. For any manifold $M_s$, represented by its parameterization $C_s$, the corresponding embedded point can be computed by

$$
t_s = v_s R.
$$

Where the projection matrix $R$ is learned from KPLS algorithm, and $v_s = K(C_s, \cdot) \in \mathbb{R}^N$. $K$ measures the similarity between $C_s$ and all training manifold parameterizations $\{C_k, k = 1 \ldots N\}$. The choice of kernel $K$ and its computation is discussed in detailed in Section 4. Because we solve two problems, speech recognition and speaker identification, we learn one $R$ for each task. We learn $R_s$ based on speech labels, and we learn $R_p$ for speaker identification with subject labels. As a result, for each $v_s$, we get two embedding in two latent spaces: $t_s^c = v_s R_c$ in the speech latent space and $t_s^p = v_s R_p$ in the speaker latent space.

Finally in manifold classification, given a latent space embedding $t_s$, MKPLS uses several techniques to classify it such as Regression for classification (RIC) [13]. Support vector machines (SVM) and K-nearest neighbor (KNN). In the latent space of speech, we want to infer the speech label ($c$) while in the speaker space, we need to infer the subject label ($p$).

### 3. Manifold-to-manifold Kernels

The parameterization, extracted out of the first phase of MKPLS, holds the dynamics in each video which encodes speech-related information along with speaker-related information. Because MKPLS uses kernel-based approach for dimensionality reduction, the kernel choice is critical for achieving the best performance. In this section, we investigate several types of kernels.

MKPLS claims that, to define manifold-to-manifold kernel, it suffices to define it in the parameterization space, i.e., $K_{manifold}(M_i, M_j) = K(C_i, C_j)$. Therefore, we need to define kernels over the space of parameterizations, which consequently, measure the similarity between manifolds in terms of their geometric deformation from the common manifold representation. MKPLS gives us the ability to plugin any valid kernel. In this section, we investigate several choice of kernels: matrix-based kernels, curve-based kernels and subspace-based kernels.

#### 3.1. Matrix-based kernels

Since the dimensionality of all parameterizations is unique, we can measure the similarity between them by measuring the similarity between the corresponding column. This is the idea behind the matrix-based kernels.

##### 3.1.1. Cosine-similarity kernel (Cosine)

We can measure the similarity between columns using cosine the angle between them. As a result, the overall similarity between two parameterizations is the sum over all column-wise similarities. Therefore, the cosine similarity kernel can be defined as

$$
K_{cos}(C_i, C_j) = \frac{tr(C_i C_j^T)^2}{\|C_i\|_F \|C_j\|_F},
$$

where $\|\cdot\|_F$ is matrix Frobenius norm.

##### 3.1.2. Euclidean-distance kernel (Eculid)

In this kernel, we measure the Euclidean distance between the $i$-th column in parameterization $C_i (u_{i,1})$ and its corresponding
points are matched between the two curves. At the end, not all other curve, and the distance Fréchet distance, also known as coupling distance, in which we consider the columns of the parameterization kernel is computed using Eqn 5.

\[ K(C_1, C_2) = \exp(-\omega \delta) \]

where \( \omega \) is a normalization factor. For \( K(\ldots) \) to be valid kernel, it needs to be symmetric positive definite matrix (SPD). The exponential function takes care of the positive definiteness part. For the symmetry, the used distance measure should be metric, which is satisfied for Euclidean distance case.

### 3.2. Curve-based Kernels

In this category, we consider the columns of the parameterization matrix as points in \( \mathbb{R}^D \), and the matrix define a curve connecting those points. The matching between columns should obey the ordering. This means that if two columns \( i, j \) from the first matrix match the columns \( u, v \) from other matrix respectively, the \( u \leq v \) if \( i < j \). For each of the following distances, the parameterization kernel is computed using Eqn 5.

#### 3.2.1. Fréchet-distance Kernel (Frechet)

Fréchet distance is a known metric to measure the distance between two curves, that takes into account the location and ordering of the points along the curves. Here, we use discrete Fréchet distance, also known as coupling distance, in which we assume that the curves are piece-wise linear. The basic idea that, each point in one curve is matched with its closest point on the other curve, and the distance \( d \) will be the maximum Euclidean distance between each two matched points. At the end, not all points are matched between the two curves.

#### 3.2.2. Edit-distance kernel (EditDist)

The idea of this metric is similar to minimum edit distance between strings. Two main difference between EditDist and Frechet algorithms: in EditDist the overall distance is the sum of distance between all matches while Frechet takes the maximum of all matches, and EditDist considers unmatched points as being matched with the origin while Frechet ignores the unmatched points. Empirically, we found that the best column-wise distance, in both EditDist and Frechet, is the Euclidean distance.

#### 3.3. Subspace-based Kernel

Each parameterization \( C \) represents \( n \)-dimensional subspace in \( \mathbb{R}^D \). Therefore, we can use subspace-to-space metric to measure the similarity in parameterization space. This gives the most general comparison between matrices. Because it considers the subspace spanned by the columns of each parameterization without encoding any ordering.

#### 3.3.1. Grassmannian kernel

Every coefficient matrix \( C \) is \( D \times n \). Since \( D \gg n \), hence \( C \) represent \( d \) dimensional subspace in \( \mathbb{R}^D \). Therefore, the matrix \( C \) belongs to Grassmannian manifold \( G_{D,d} \). For more details about Grassmannian manifolds, the reader is referred to [15].

There are several approaches for measuring the similarity on Grassmannian manifold, we use the one defined in [16].

\[ K_{ij} = a_1 K_{ij}^{cc} + a_2 K_{ij}^{proj} \]

Where \( K_{ij}^{proj} \) and \( K_{ij}^{cc} \) are the projection kernel and the canon-
We apply single feature configuration. The projection kernel is defined by $K_{ij}^{(p)} = \| \Pi_i \Pi_j \|_F^2$, where $\Pi_i$ is the orthogonal version of the coefficient matrix $C_i$, computed by Gram-Schmidt orthogonalization algorithm. The canonical correlation kernel is defined by

$$K_{ij}^{cc} = \max_{a_p \in \text{span}(\Pi_i)} \max_{b_q \in \text{span}(\Pi_j)} a_p^T b_q$$

Subject to $a_p^T a_p = b_q^T b_q = 1$ if $p = q$, and 0 otherwise. We use two Grassmannian kernels: Grassm defined by Eq 6 and GrassmCC defined by Eq 7.

Since Grassmannian distance does not consider the ordering of the parameterization columns, we can encode some temporal information by using the parameterization of difference of the input features. We denote this experiment by GrassmDiff.

4. Experiments

4.1. Databases

We apply MKPLS for OuluVs database [11]. OuluVs has ten different everyday phrases. Each phrase is uttered by 20 subjects up to five times. We use the same test protocol used in [12]. The frame rate was set to 25 fps. The dataset contains sequence of images for mouth area with average resolution of $120 \times 60$ pixels. LBP [17] visual features is extracted from images. Two feature configurations have been used: the first configuration is $LBP_{1:8 \times 8}$ with $n = 16$ (16 basis for Gaussian-RBF) and the second one is $1 \times 4 LBP_{3 \times 8 \times 8}$ with $n = 8$. We also apply MKPLS for AVLetters database [22] which has ten subjects. Each speaker repeats every English letter ($A \cdots Z$) exactly three times, with a total of 780 video sequences. The speaker was requested to start and end utterance of every letter in a neutral state (mouth closed). We apply single feature configuration: $3 \times 4$ cell-grid with four-resolutions LBP features ($3 \times 4 LBP_{3 \times 8 \times 8}$) with $n = 8$.

In all experiments, the recognition rate is measured as the ratio between the correctly recognized clips and the total number of clips.

4.2. Experimental Results

In this section, we present the empirical results of MKPLS when plugged with each one of the kernels described in Section 3. To give real comparison between the proposed kernels, we need to explore different parameters of MKPLS pipeline: The number ($n$) of Gaussian-RBF basis $\psi$ that we use to learn the individual manifold parameterization, we show results for $n = 8, 16$. The dimensionality of the manifold latent space, we use $m = 10, 30, 50, 80, 100, 130, 200$, which cover wide range of the possible values.

Especially for Grassmann-based kernels [3] we show the affect of using the parameterization of the LBP of the images itself vs LBP of the images concatenated with parameterization of the discrete difference between those images.

Table 2: Speaker Indepenent - speech recognition Accuracy on OuluVs database

| $m$ | 10 | 30 | 50 | 80 | 100 | 130 | 200 |
|-----|----|----|----|----|-----|-----|-----|
| Cosine | 50.08 | 50.00 | 49.84 | 50.00 | 49.84 | 50.00 |
| Euclid | 48.50 | 55.75 | 55.00 | 55.47 | 55.47 | 55.00 |
| EditDist | 47.97 | 50.94 | 41.25 | 26.09 | 21.72 | 18.12 | 16.41 |
| Frechet | 11.25 | 13.28 | 12.50 | 12.34 | 13.75 | 13.44 | 13.75 |
| Grassm | 29.84 | 30.31 | 32.34 | 31.09 | 28.59 | 24.84 | 31.56 |
| GrassmCC | 29.84 | 30.31 | 32.34 | 31.09 | 28.59 | 24.84 | 31.56 |

Table 2 shows the SI speech recognition accuracy for OuluVs database with the two feature configurations. Table 4 shows the result of matrix-based kernels and curve-based kernels applied avletters.

Table 4: SSD speech recognition on AvLetters

| $m$ | 10 | 30 | 50 | 80 | 100 | 130 | 200 |
|-----|----|----|----|----|-----|-----|-----|
| Cosine | 50.08 | 50.00 | 49.84 | 50.00 | 49.84 | 50.00 |
| Euclid | 48.50 | 55.75 | 55.00 | 55.47 | 55.47 | 55.00 |
| EditDist | 47.97 | 50.94 | 41.25 | 26.09 | 21.72 | 18.12 | 16.41 |
| Frechet | 11.25 | 13.28 | 12.50 | 12.34 | 13.75 | 13.44 | 13.75 |
| Grassm | 29.84 | 30.31 | 32.34 | 31.09 | 28.59 | 24.84 | 31.56 |
| GrassmCC | 29.84 | 30.31 | 32.34 | 31.09 | 28.59 | 24.84 | 31.56 |

4.3. Discussion

From the numbers, we can clearly see the superiority of both techniques of Matrix-based kernels (Cosine and Euclid) in all test and features configurations. For $m = 10$, EditDist gives the best results for SSD-speech recognitoin and Speaker Semi-Dependent (SSD) as defined in [13].

Speaker Semi-Dependent VSR (SSD): Here we test on one repeat of the available videos and train based on the remaining repeats for the same subjects. In this configuration all subjects and phrases are presented in the training set. Table 1 shows the SSD speech recognition accuracy for OuluVs database with the two feature configurations. Table 4 shows the result of matrix-based kernels and curve-based kernels applied avletters.

4.2.4. Speaker Identification (SPlit):

The goal in this experiment is to find the speaker within the register set of users. The challenge is to find the speaker from the limited available information in the mouth area. We take one repetition out for testing, and train over all other repetitions. Table 3 shows the speaker identification accuracy when applied to OuluVs for the two test configurations.

Speaker Independent VSR (SI): the challenge here is to show the scalability of the model, mean how far the model can recognize the spoken phrase based on the dynamics even if the speaker is not seen before in the training set. In this experiment, we use one-speaker-out. Table 3 shows the SI speech recognition accuracy for OuluVs for the two configurations.

Test two protocols has been adopted for visual speech recognition: Speaker Independent (SI) and Speaker Semi-Dependent (SSD) as defined in [13].

Speaker Independent VSR (SI): Here we test on one repeat of the available videos and train based on the remaining repeats for the same subjects. In this configuration all subjects and phrases are presented in the training set. Table 1 shows the SSD speech recognition accuracy for OuluVs database with the two feature configurations. Table 4 shows the result of matrix-based kernels and curve-based kernels applied avletters.
5. Conclusion

We investigated the kernel choice for the middle phase of MKPLS framework. We explored three kinds of manifold kernels: matrix-based kernel, curve-based kernel and subspace-based kernel. We compare the kernels based on different parameter configuration for MKPLS. The experiments shows that using parameterization-to-parameterization kernel can delegate manifold-to-manifold kernel. The results shows the superiority of the matrix-based kernel for visual speech recognition. For speaker identification tasks, all kernels gives perfect results.

6. References

[1] G. Potamianos and C. Neti, “Audio-visual automatic speech recognition: An overview,” Issues in Visual and Audio-Visual Speech Processing.

[2] H. McGurk and J. MacDonald, “Hearing lips and seeing voices,” Nature, vol. 264, no. 23 December, pp. 746–748, 1976.

[3] D. Shiell and L. Terry, “Audio-Visual and Visual-Only Speech and Speaker Recognition: Issues about Theory, System Design, and Implementation,” Visual speech recognition: lip segmentation and mapping.

[4] J. Luettin, N. Thacker, and S. Beet, “Speaker identification by lipreading,” International Conference on Spoken Language Processing, pp. 1–4.

[5] C. Sanderson and K. Paliwal, “Identity verification using speech and face information,” Digital Signal Processing, vol. 5, pp. 449–480, Sep.

[6] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, no. 2, pp. 257–286.

[7] I. Matthews and T. Cootes, “Extraction of visual features for lipreading,” PAMI, vol. 24, no. 2, pp. 198–213, 2002.

[8] K. Saenko and K. Livescu, “Visual speech recognition with loosely synchronized feature streams,” IEEE International Conference on Computer Vision.

[9] Z. Zhou and G. Zhao, “Lipreading: A Graph Embedding Approach,” IJCV, pp. 523–526, 2010.

[10] Z. Zhou, G. Zhao, and M. Pietikainen, “Towards a practical lipreading system,” Computer Vision and Pattern Recognition, 2011.

[11] G. Zhao, “Lipreading with local spatiotemporal descriptors,” IEEE Transactions on Multimedia, pp. 1–11, 2009.

[12] A. Shaikh, D. Kumar, and W. Yau, “Lip Reading using Optical Flow and Support Vector Machines,” IEEE International Congress on Image and Signal Processing, vol. 1, pp. 327–330, Oct. 2010.

[13] A. Bakry and A. Elgammal, “MKPLS: Manifold Kernel Partial Least Squares for Lipreading and Speaker Identification,” Computer Vision and Pattern Recognition, 2013.

[14] R. Rosipal and L. Trejo, “Kernel partial least squares regression in reproducing kernel hilbert space,” The Journal of Machine Learning Research, pp. 97–123.

[15] A. Edelman, T. a. Arias, and S. T. Smith, “The Geometry of Algorithms with Orthogonality Constraints,” SIAM Journal on Matrix Analysis and Applications, no. 2, pp. 303–353, Jan.

[16] M. Harandi and C. Sanderson, “Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 2705–2712, Jun.

[17] T. Ojala, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” PAMI, no. 7, pp. 971–987.