A Multi-class Classification Strategy for Fisher Scores: Application to Signer Independent Sign Language Recognition

Oya Aran [a,*,1] Lale Akarun [b]

[a] *Idiap Research Institute, Martigny, Switzerland*

[b] *Department of Computer Engineering, Bogazici University, Istanbul, Turkey*

Abstract

Fisher kernels combine the powers of discriminative and generative classifiers by mapping the variable-length sequences to a new fixed length feature space, called the Fisher score space. The mapping is based on a single generative model and the classifier is intrinsically binary. We propose a strategy that applies a multi-class classification on each Fisher score space and combines the decisions of multi-class classifiers. We experimentally show that the Fisher scores of one class provide discriminative information for the other classes as well. We compare several multi-class classification strategies for Fisher scores generated from the Hidden Markov Models (HMMs) of sign sequences. The proposed multi-class classification strategy increases the classification accuracy in comparison with the state of the art strategies based on combining binary classifiers. To reduce the computational complexity of the Fisher score extraction and the training phases, we also propose a score space selection method and show that, similar or even higher accuracies can be obtained by using only a subset of the score spaces. Based on the proposed score space selection method, a signer adaptation technique is also presented that does not require any re-training.

Key words: Fisher scores, multi-class classification, hidden Markov models, generative and discriminative classifiers, sign language recognition

[*] Corresponding Author, Tel: +41 27 7217758, Fax: +41 27 7217712
Email addresses: `oya.aran@idiap.ch` (Oya Aran), `akarun@boun.edu.tr` (Lale Akarun).

[1] Most of this work is done when the author was with Bogazici University

1 Introduction

Sign language recognition relies on spatiotemporal modeling of the hand. This can be achieved by using models that can handle variable length sequences and the dynamic nature of the data. Several methods are proposed and applied in the literature for modeling the dynamics of signs or hand gestures. These include Hidden Markov Models (HMM) [1] and its variants, dynamic time warping, time delay neural networks, and temporal templates [2,3], with HMMs being the most extensively used method.

Isolated sign language recognition can be defined as a sequence classification problem where each sign is represented as a variable length sequence. With their power in modeling sequential data and processing variable length sequences, HMMs offer a natural solution for modeling signs. However, HMMs as generative models, are not as successful as discriminative models on classification problems. Discriminative methods (such as support vector machines and neural networks) have flexible decision boundaries and better classification performance. Although most of the discriminative methods in the literature are only suitable for fixed length data, several approaches are proposed to learn variable length sequential data via discriminative models [4],[5].

Fisher kernels have been proposed as a method to map a variable length sequence to a new fixed dimension feature vector space [6]. The mapping is obtained by the derivatives of the parameters of an underlying generative model. This new feature space is called the Fisher score space [7] on which, any discriminative classifier can be used to perform discriminative training. The main idea of Fisher kernels is to combine generative models with discriminative classifiers to obtain a robust classifier which has the strengths of each approach. Since each Fisher score space is based on a single generative model, the new feature space is assumed to be suitable for binary classification problems in nature. In a multi-class classification problem where each class is represented by a different generative model, one would have as many Fisher score spaces as classes.

In the literature, Fisher kernels have been applied to binary classification problems such as bio-sequence analysis [6], protein homology detection [8], and also to multi-class classification problems such as audio classification [9], speech recognition [7], object recognition [10], texture classification [11], and face recognition [12]. To solve these multi-class classification problems, in most of these works, the researchers apply either a one-versus-one (1vs1) or a one-versus-all (1vsAll) scheme. In 1vs1 and 1vsAll, binary classification is applied and then the decisions of the binary classifiers are combined to give a multi-class decision [9][7][10][11]. In [12], the authors concatenate all the Fisher scores generated from the models of each class into a single feature vector.

Then, they apply a multi-class classification to this combined feature vector and achieve higher recognition performance when compared to binary classification schemes.

This study aims to use Fisher kernels to map the original variable length sign sequences based on HMMs to the fixed dimension Fisher score space, and to apply a discriminative multi-class classification on this new feature space. Preliminary versions of this work can be found in [13],[14]. In this work, we propose a new multi-class classification scheme that applies a multi-class classification on the Fisher score space of each generative model. In this approach, we use the discriminative power of the Fisher scores of one class to classify other classes. We compare this scheme with state of the art techniques applied for multi-class classification and show that the method is both more accurate in comparison with binary classification schemes and computationally more effective than concatenating all the score spaces into one feature vector, especially in terms of the memory requirements for high dimensional problems. Our results show that if the multi-class scheme is not properly determined, the recognition performance of the combined classifier may decrease even with respect to the underlying generative model.

One disadvantage of the Fisher scores is the high dimensionality of the generated feature vectors. The dimension of the new feature vector is directly related to the number of parameters of the underlying generative model. Although the generative models for gesture and sign sequences are as simple as a left-to-right HMM with a few states, the dimensionality of the Fisher scores gets higher with the feature dimensionality of the sequences and the number of classes. So, both the computation of the Fisher scores and the discriminative training on the new feature space become costly. We analyze the effect of the parameters of the generative model and each score space, on the recognition performance. To reduce this computational complexity, we propose to perform score space selection and compare several methods such as parameter selection, and dimensionality reduction with PCA or LDA.

Our contributions in this paper can be summarized as follows: We propose a new multi-class classification scheme that applies a multi-class classification on the Fisher score space of each generative model. We compare state-of-the-art techniques to reduce the computational complexity of Fisher score extraction, training and testing phases and we show that the complexity can be further reduced, without compromising the accuracy, by an intelligent score space selection strategy. Lastly, we present a signer adaptation strategy that does not require re-training of the system. During the score space selection phase, examples from the current signer is used to determine the score space subset. To show the validity of the proposed techniques, we conduct experiments on a sign language dataset as well as on a hand gesture and a head gesture/facial expression dataset. We compare the performance of several state-of-the-art multi-class classification schemes with the proposed scheme and show that the proposed scheme provides the highest accuracy and enhances the performance of the base classifier the most. The organization of the paper is as follows: In Section 2, we introduce the Fisher kernel methodology in detail. The multi-class classification strategies are discussed in Section 3. Section 4 presents our proposed multi-class classification strategy. In Section 5, we discuss several strategies for reducing the computational cost of Fisher score calculation and classification. The results of the experiments are reported in Section 6.

2 Fisher Kernels and Score Spaces

A mapping function, ϕ, that is capable of mapping variable length sequences to fixed length vectors enables the use of discriminative classifiers for variable length examples. Fisher kernel [6] defines such a mapping function and is designed to handle variable length sequences by deriving the kernel from a generative probability model. The gradient space of the generative model is used for this purpose. The gradient of the log likelihood with respect to a parameter of the model describes how that parameter contributes to the process of generating a particular example. All the structural assumptions encoded in the model about the generation process are naturally preserved in this gradient space [6]. Higher order Fisher kernels can also be constructed by taking the second or third order derivatives.

Fisher score, U_X, is defined as the gradient of the log likelihood with respect to the parameters of the model:

$$U_X = \nabla_\theta logP(X|\theta) \tag{1}$$

U_X defines a mapping to a feature vector, which is a point in the gradient space of the manifold of the probability model class. The direction of steepest ascent in $logP(X|\theta)$ along the manifold can be calculated by the Fisher score, U_X. By normalizing via the diagonal of the covariance matrix, Σ_S, of the score space estimated from the training set, the normalized Fisher kernel can be defined as follows:

$$K(X_i, X_j) = U_{X_i}^T \Sigma_S^{-1} U_{X_j}^T \tag{2}$$

In practice, Fisher scores are used to extract fixed size feature vectors from variable length sequences modeled with any generative model. This new feature space can be used with any discriminative classifier. However, the dimensionality of this new feature space can be high when the underlying generative model has many parameters and the original feature space is multivariate.

Table 1
Fisher, likelihood and likelihood ratio score spaces

Score Space	Feature Vector
FSS	$\nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1)$
LSS	$\begin{bmatrix} log\ p_1(O\|\hat{\Theta}_1) \\ \nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1) \end{bmatrix}$
LRSS	$\begin{bmatrix} log\ p_1(O\|\hat{\Theta}_1) - log\ p_2(O\|\hat{\Theta}_2) \\ \nabla_{\hat{\Theta}_1} log\ p_1(O\|\hat{\Theta}_1) \\ -\nabla_{\hat{\Theta}_2} log\ p_2(O\|\hat{\Theta}_2) \end{bmatrix}$

Thus, Support Vector Machines (SVMs) become a good choice of a classifier since they do not suffer from the curse of dimensionality [15].

2.1 Fisher Score Spaces

Score spaces are generalizations of Fisher kernels and define the mapping space [16]. A score space is derived from the likelihood of a generative model. Score vectors are calculated by applying a score operator to the score argument. Score argument can be the log likelihood or posterior of the generative model, whereas the score operator can be the first or second derivative, or the argument itself. Table 1 shows the Fisher (FSS), Likelihood (LSS) and Likelihood Ratio (LRSS) Score Spaces. $p_1(O|\hat{\theta}_1)$ and $p_2(O|\hat{\theta}_2)$ are the likelihood estimates produced by the generative models of class 1 and class 2, respectively. Other score spaces and their derivations can be found in [7].

The difference between the FSS and the LSS is that the latter also uses the likelihood itself in the score vector. LRSS represents the two classes by putting the likelihood ratio instead of the likelihood in the score vector, together with the score operators for each of the classes.

2.2 Fisher Kernels for HMMs Using Continuous Density Mixture of Gaussians

In sign language recognition and hand gesture recognition problems, HMMs are extensively used and have proven successful in modeling hand gestures.

Among different HMM architectures, left-to-right models with no skips are shown to be superior to other HMM architectures [17].

In this work, we have used continuous observations in a left-to-right HMM with no skips. The parameters of such an architecture are, prior probabilities of states, π_i, transition probabilities, a_{ij} and observation probabilities, $b_i(O_t)$ which are modeled by mixture of M multivariate Gaussians:

$$b_i(O_t) = \sum_{m=1}^{M} w_{im} \; \mathcal{N}(O_t; \mu_{im}, \Sigma_{im}) \tag{3}$$

where O_t is the observation at time t and w_{im}, μ_{im}, and Σ_{im} are weight, mean and covariance of the Gaussian component m at state i, with a total of M Gaussian components.

For a left-to-right HMM, the prior probability matrix is constant since the system always starts with the first state with $\pi_1 = 1$. Moreover, using only self-transition parameters is enough since there are no state skips $(a_{ii}+a_{i(i+1)} = 1)$. Observation parameters in the continuous case are weight, w_{im}, mean, μ_{im} and covariance, Σ_{im} of each Gaussian component. The first order derivatives of the log-likelihood, $P(O|\theta)$ with respect to each parameter are given below:

$$\nabla_{a_{ii}} = \sum_{t=1}^{T} \frac{\gamma_i(t)}{a_{ii}} - \frac{1}{T \; a_{ii} \; (1 - a_{ii})} \tag{4}$$

$$\nabla_{w_{im}} = \sum_{t=1}^{T} [\frac{\gamma_{im}(t)}{w_{im}} - \frac{\gamma_{i1}(t)}{w_{i1}}] \tag{5}$$

$$\nabla_{\mu_{im}} = \sum_{t=1}^{T} \gamma_{im}(t) \; (O_t - \mu_{im})^T \; \Sigma_{ik}^{-1} \tag{6}$$

$$\nabla_{\Sigma_{im}} = \sum_{t=1}^{T} \gamma_{im}(t) \; [-\Sigma_{im}^{-1} + \Sigma_{im}^{-1} \; (O_t - \mu_{im}) \; (O_t - \mu_{im})^T \; \Sigma_{im}^{-1}] \tag{7}$$

where $\gamma_i(t)$ is the posterior probability of state i at time t and $\gamma_{im}(t)$ is the posterior probability of component m of state i at time t, and T is the total length of the data sequence. Since the component weights of a state sum to 1, one of the weight parameters at each state, i.e. w_{i1}, can be eliminated. The derivations of these gradients can be found in [18]. These gradients are concatenated to form the new feature vector which is the Fisher score. The log-likelihood score space where log-likelihood itself is also concatenated to the feature vector is given as:

$$\phi_{O_t} = diag(\Sigma_S)^{-\frac{1}{2}} \left[log \; p(O_t|\theta) \; \nabla_{a_{ii}} \; \nabla_{w_{im}} \; \nabla_{\mu_{im}} \; \nabla_{vec(\Sigma)_{im}} \right]^T \tag{8}$$

When the sequences are of variable length, it is important to normalize the scores by the length of the sequence. We have used *sequence length normalization* [16] for normalizing variable length sign trajectories by using normalized component posterior probabilities, $\hat{\gamma}_{im}(t)$, in the above gradients:

$$\hat{\gamma}_{im}(t) = \frac{\gamma_{im}(t)}{\sum_{t=1}^{T} \gamma_i(t)} \tag{9}$$

3 Methods for Multi-class Classification Using Fisher Scores

As Fisher kernels are extracted from generative models which are trained with the examples of a single class, the new feature space of Fisher scores is mainly representative of that class. In [6], where the idea of Fisher kernels is proposed, the authors applied Fisher scores to a binary classification problem.

For a binary classification problem, one might have three different score spaces based on likelihoods:

(1) LSS from the generative model of class 1
(2) LSS from the generative model of class 2
(3) LRSS from the generative models of class 1&2

LRSS contains discriminative features from each class and provides a good representation for binary classification problems. Experiments show that it gives slightly better results with respect to LSS, which is based on single class information [7].

For a multi-class problem, one needs a similar combination method: Fisher scores from each generative model must be combined to obtain a good multi-class representation. A general method to extend the Fisher scores to multi-class classification problems is to apply binary classification and combine the results via decision or score level combination. In the next section, we summarize four schemes that are commonly used for the multi-class classification on Fisher scores. The first three schemes, B_{1vs1}, B_{1vs1R}, and B_{1vsALL} are the commonly used multi-class classification techniques based on a binary classifier, such as 1vs1 and 1vsAll. Alternatively, in [12] authors use a feature level combination approach (M_{FLC}) and concatenate all the Fisher scores into a single feature vector. Then, they apply multi-class classification on the combined feature vector. We present our approach, (M_{DLC}) in Section 4.

A summary of all the schemes is given in Figure 1 and Table 2. For the strategies that use decision/score level combination, the fusion is done via weighted voting: a test example is given to each classifier and the final result is obtained by selecting the class with the maximum weighted vote, with posterior

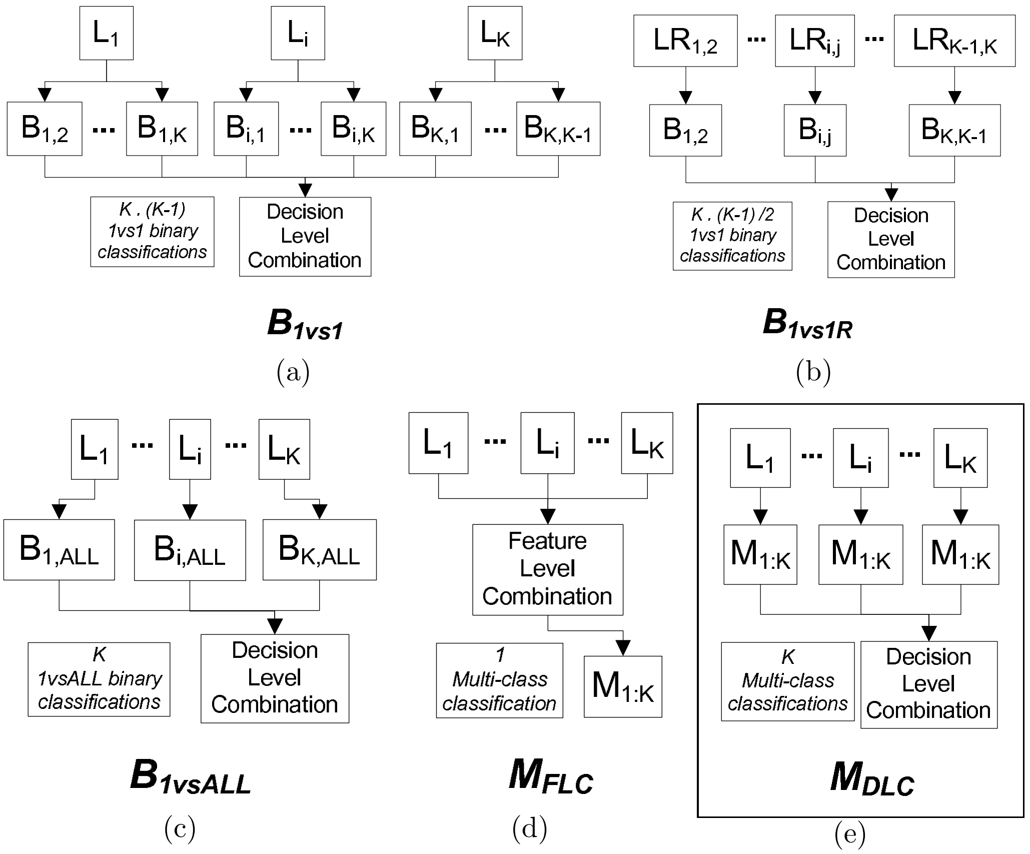

L1
Li
LK
B1,2
B1,K
Bi,1
Bi,K
BK,1
BK,K-1
K . (K-1) 1vs1 binary classifications
Decision Level Combination
B1vs1
(a)
LR1,2
LRi,j
LRK-1,K
B1,2
Bi,j
BK,K-1
K . (K-1) /2 1vs1 binary classifications
Decision Level Combination
B1vs1R
(b)
L1
Li
LK
B1,ALL
Bi,ALL
BK,ALL
K 1vsALL binary classifications
Decision Level Combination
B1vsALL
(c)
L1
Li
LK
Feature Level Combination
1 Multi-class classification
M1:K
MFLC
(d)
L1
Li
LK
M1:K
M1:K
M1:K
K Multi-class classifications
Decision Level Combination
MDLC
(e)

Fig. 1. Multiclass classification strategies: (a) B_{1vs1}, (b) B_{1vs1R}, (c) $B_{1vsALL},(d)M_{FLC}$, and (e) proposed method, M_{DLC}

Table 2
Multiclass classification strategies for Fisher scores. K denotes the number of classes

Strategy	Score Space Used	# & type of classifiers	Combination of Score Spaces
B_{1vs1}	LSS	$K \cdot (K-1)$, binary	Decision/score level
B_{1vs1R}	LRSS	$\frac{K \cdot (K-1)}{2}$, binary	Decision/score level
B_{1vsALL}	LSS	K, binary	Decision/score level
M_{FLC}	LSS	1, multiclass	Feature level
M_{DLC}	LSS	K, multiclass	Decision/score level

probabilities as the weights.

3.1 $\mathbf{B_{1vs1}}$*: One-vs-One binary classification based on LSS*

We can form a multiclass scheme by utilizing the LSS of each class: For each class pair (i, j), a binary classification is performed to classify whether the example belongs to class i or j.

Note that the binary classifier for class pair (i, j) uses the LSS of class i and the binary classifier for class pair (j, i) uses the LSS of class j. Hence, (i, j) and (j, i) are not symmetric problems.

For a problem of K classes, $K \cdot (K - 1)$ binary classifiers must be trained with examples from class i and j for each (i, j) pair, where $i, j = 1 \ldots K$ *and* $i \neq j$.

3.2 $\mathbf{B_{1vs1R}}$*: One-vs-One binary classification based on LRSS*

Since the LRSS is the best performing score space for binary problems, a multiclass scheme which performs a binary classification for each pair of classes, (i, j), by using the LRSS of classes i and j can be formed.

For each class pair (i, j), a binary classification is performed to determine whether the example belongs to class i or j. The binary classifier for class pair (i, j) uses the LRSS of classes (i, j). Unlike the previous scheme, (i, j) and (j, i) are now symmetric problems: the LRSS of (i, j) is the same as that of (j, i), except for a sign difference (see Table 1).

For a problem of K classes, $\frac{K \cdot (K-1)}{2}$ binary classifiers must be trained with examples of class i and j for each (i, j) pair, where $i, j = 1 \ldots K$ and $i \neq j$ since i-vs-j and j-vs-i are symmetric problems.

3.3 $\mathbf{B_{1vsALL}}$*: One-vs-All binary classification based on LSS*

Another multiclass scheme can be formed by using one-vs-all scheme. The one-vs-all scheme is shown to be as successfull as other multi-class classification schemes[19].

We use the LSS of each class and apply a binary classification: for each class i, a binary classification is performed to classify whether the example belongs to class i or any other class.

For a problem of K classes, K binary classifiers must be trained with all the examples of the training set where for each classifier $C_i, i : 1 \ldots K$, examples of class i are labeled as 0 and all other examples are labeled as 1.

3.4 $\mathbf{M_{FLC}}$*: Multi-class classification based on feature level combination of LSS*

In this scheme, the score spaces of each class are combined into a single feature space and a multiclass classification is performed on this new feature space, as proposed in [12]. For a problem of K classes, a single multi-class classifier is trained, using the original class labels.

The main disadvantage of this scheme is the memory consumption since the resulting feature vector is the combination of multiple Fisher scores. When the input dimensionality, the number of parameters of the generative models and the number of classes are high, the dimensionality of the combined feature space will be extremely high, making it hard, or sometimes impossible, to keep the training data in the memory.

4 A New Multi-Class Classification Scheme for Fisher Scores

We propose a new strategy, M_{DLC}, which applies a multi-class classification on the LSS of each class and then combines the decisions of each classifier. M_{DLC} is especially suitable for applications where the number of classes is large and computational resources are critical.

M_{DLC} uses the Fisher scores of each class for the discrimination of all the other classes, not just for the class that produces the scores. In the above schemes, for a binary classification between class i and class j, Fisher scores extracted for related generative models (models for class i or j) are used. However, in this study, we show that Fisher scores extracted from class i may provide a discrimination for classes other than i (i.e. discrimination between class j and k).

For a problem of K classes, we train K multi-class classifiers with all the examples of the training set, using the original class labels. The main difference of this scheme is that, each of the K classifiers is performing a multi-class classification, whereas in all the above schemes except M_{FLC}, each classifier is a binary classifier.

To demonstrate this multi-class scheme, let us consider a toy problem. We have generated 2-D random data from four different Gaussian distributions (Figure 2) and added a small amount of noise. Each example consists of the x, y values and the corresponding class label. Fisher scores are extracted for each Gaussian model for the parameters $(\mu_1, \sigma_1, \mu_2, \sigma_2)$. An example plot for the score space of class 4 is shown in Figure 3. It can be seen from the derivative plots that the Fisher scores obtained from a single parameter, μ or σ, provide good discrimination among classes. Similar behavior is observed in the score spaces of other classes as well.

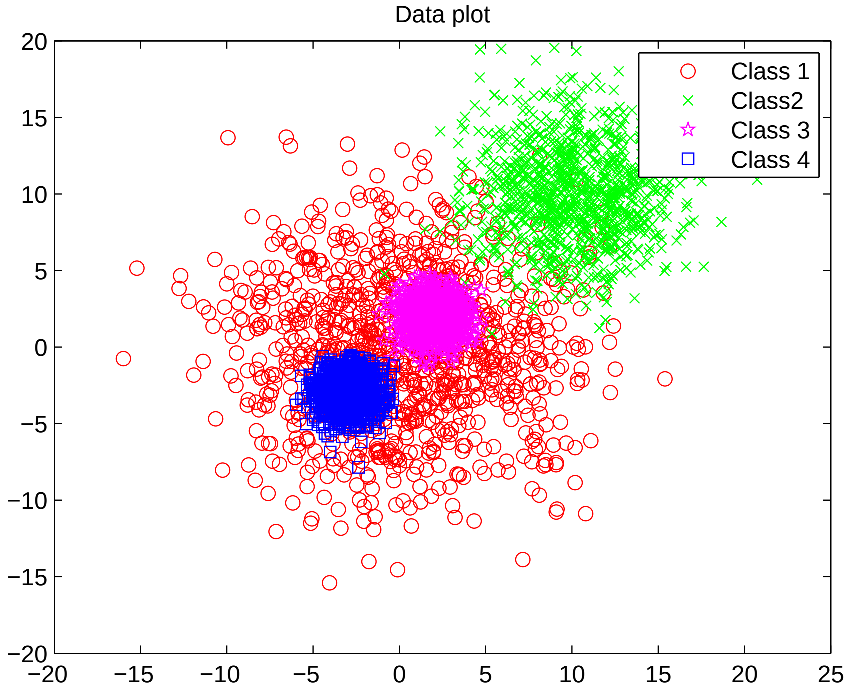


Fig. 2. Artificial data generated from four 2D Gaussian distributions

The classification results of applying a multi-class classification on each score space are shown in Table 3. Different rows show the score space used and the columns show the classification accuracy among all the classes and also of each class pair. The classification on each score space is performed with SVMs that apply multi-class classification. Note that the multi-class classification strategy used by the SVMs is not related to the multi-class classification strategy used for the Fisher scores and is completely independent. Apart from SVMs, other discriminative multi-class classifiers may also be employed.

Classes 2, 3 and 4 are easily separated by any of the score spaces, where LSS_4 provides the best discrimination among classes. As can also be seen from the distribution of the classes in Figure 2, using only the generative model of i.e. class 1, one can obtain very high classification performance. This small experiment shows that the score space obtained from the generative model of class i is not only capable of discriminating between class i and other classes, it also provides valuable discriminative information for classes other than i, such as j and k. For example, in Table 3, we observe that the score spaces of class 1 and 4, LSS_1 and LSS_4, provide the highest accuracy in discriminating between classes 2 and 3.

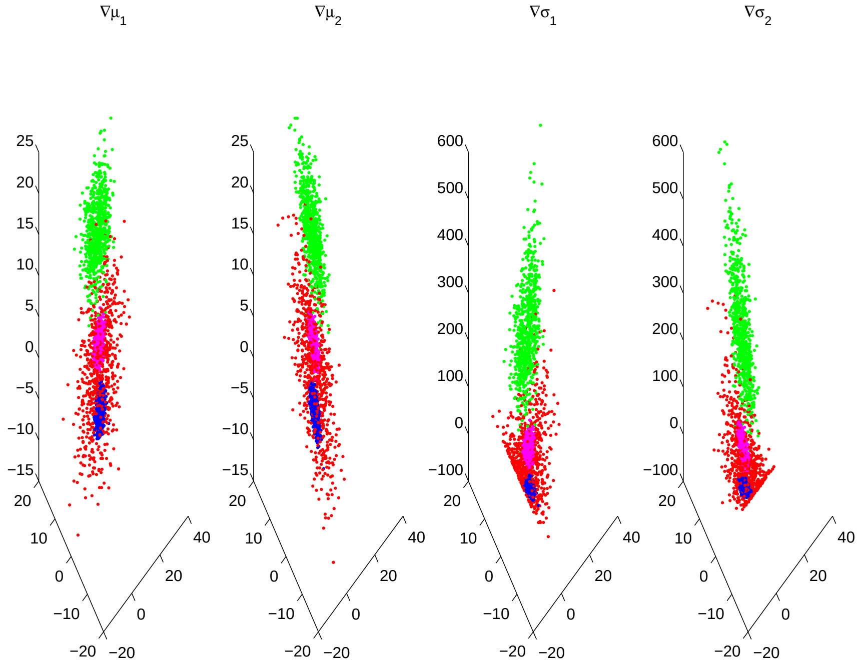


Fig. 3. Score space plot of Class 4, $N(-3,1)$

Table 3
Classification results of applying multi-class classification on each score space. The highest accuracy in each column is shown in bold.

	Classification Accuracy (%) for						
Score Space(s) used	All classes	Classes 1&2	Classes 1&3	Classes 1&4	Classes 2&3	Classes 2&4	Classes 3&4
LSS_1	93.25	88.83	**88.50**	88.50	**98.00**	98.00	97.67
LSS_2	93.00	**89.33**	88.00	88.33	97.67	98.00	96.67
LSS_3	93.08	88.83	88.33	88.33	97.83	97.83	97.33
LSS_4	**93.33**	88.83	**88.50**	**88.67**	**98.00**	**98.17**	**97.83**
LSS_{1234}	93.08	88.83	88.33	88.33	97.83	97.83	97.33

5 Reducing the Computational Cost

In this section, we first list the elements that affect the computational cost and then discuss possible techniques that can be used to reduce it. The computational complexity of Fisher score usage arises from two phases: extraction of Fisher scores and training.

Table 4
Techniques to reduce the computational cost

	Time required to extract a single score space	# of score spaces to extract	Training/ Testing time
Feature/parameter selection	↓	-	↓
Dimensionality reduction	-	-	↓
Score space selection	-	↓	↓

In the extraction phase, critical parameters are the number of parameters of the underlying generative model, the length of the input data sequence, and the number of classes/models. In the training phase, the Fisher score dimensionality (dependent on the number of parameters of the generative model) and the number of classes/models are the critical parameters.

The computational cost of the extraction phase can be decreased by either using simpler base models and simpler feature vectors or by considering only a subset of the parameters of the base model for Fisher score extraction. For the rest of this section we will assume that the models and the features of the sign sequences are fixed and we will concentrate on reducing the complexity of processes directly related to the Fisher score extraction and training. Table 4 shows three possible techniques that can be used to reduce the computational cost. Parameter selection and dimensionality reduction are commonly used techniques. We propose score space selection to reduce the computational cost which follows the idea of our proposed multi-class classification strategy. In the M_{DLC} strategy, we can obtain classification decisions for all classes even with a single Fisher score mapping. Similar to the parameter selection, a score space selection strategy can be applied to find a reduced set of score spaces that performs well enough. This selection will affect both the Fisher score extraction and the training time since the number of score spaces to work on will be smaller.

5.1 Parameter Selection

Selecting a subset of the model parameters effects both the Fisher score extraction time and the training time since dimensionality of the Fisher score spaces will be smaller. Some of the parameters may have a greater effect on the classification performance and the optimal reduced parameter set with an acceptable performance can be determined in a validation phase. For the case of HMMs, Fisher scores extracted from the transition parameters and the observation parameters can be of different importance during the discriminative training and thus have a different effect on the accuracy.

5.2 Dimensionality Reduction

The training time can be further reduced by applying dimensionality reduction techniques, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), on the extracted Fisher scores [20].

5.3 Score Space Selection Strategies

In a multi-class classification task with Fisher scores, multiple score spaces are obtained from the generative model of each class. In the M_{DLC} strategy, as each score space is able to provide a decision for all of the classes, a subset of these score spaces can be used instead of using all. Then, one should select the best subset that gives the highest recognition accuracy. As the number of classes increases, the number of possible subsets increases exponentially. Hence, efficient and effective search techniques should be applied. We will use and compare the following search techniques, which are commonly used in the literature:

Sequential Floating Forward Search (SFFS) is originally proposed as a feature selection algorithm that applies a top down search strategy [21]. At each iteration of SFFS, the algorithm attempts to add one of the features to the combined feature set, which is initially empty. The feature to be added is selected such that, when added to the combined set, it increases the accuracy the most. This step is called the forward optimization or the inclusion. The floating property comes from the fact that after each forward optimization, a backward optimization (exclusion) step, which attempts to remove one of the features from the combined feature set, is applied. The feature to be removed is selected such that, when removed from the combined set, it increases the accuracy the most. The algorithm stops when there is no improvement in the accuracy at the end of forward and backward optimization steps.

We use SFFS as a score space selection method and for each score space, obtained from the generative model of each class, we train a classifier. Our aim is to select the score spaces (classifiers) to be combined by running Sequential Forward Floating Selection (SFFS) with the classification accuracy as the objective function.

Sequential Floating Backward Search (SFBS) is a variant of SFFS, in which, instead of starting with an empty set, the algorithm starts with all the features in the combined set and attempts to remove the features at each iteration. Thus, SFBS first performs a backward optimization step, followed by a forward optimization step.

Selecting the best performing *N* score spaces (BNSS) attempts to find a subset of N score spaces with respect to their single subset accuracies. The method orders the score spaces with respect to their accuracy and selects the first N as the subset.

6 Experiments

We have performed experiments on the eNTERFACE American sign language database [22], which contains 19 signs with both manual and non-manual components. The manual components are the ones that are performed by the hand motion, hand shape and position and the non-manual components are performed by the facial expressions, and head and body movements. The signs in the eNTERFACE database are multimodal (contain hand and head modalities) and are selected such that some signs have exactly the same manual component and can only be differentiated by the non-manual components.

We have concentrated on the manual component and extracted features only from the hand motion, shape and position. The videos are first processed for hand and face detection and segmentation. Then, sign features are extracted for manual signs (hand motion, hand shape, hand position with respect to face). For hand motion analysis, the center of mass (CoM) of each hand is tracked and filtered by a Kalman filter. The posterior states of each Kalman filter, x, y coordinates of CoM and horizontal and vertical velocities, are the hand motion features. Hand shape features are appearance based shape features calculated on the segmented hand images. These features include the width, height and orientation parameters of an ellipse and seven Hu moments [23] calculated on the binary hand image. Hand position features are the normalized horizontal and vertical distances of the hand center to the face center. As a result, the feature vector dimensionality is 32 per frame (four hand motion, 10 hand shape and two hand position features for each hand). More information on the database and feature extraction can be found in [22,24].

6.1 Experimental Setup

We followed a signer independent protocol in the experiments where the subjects in the training set and in the validation and test sets are different. The eNTERFACE database is collected from eight subjects where each subject performed five repetitions for each of the 19 signs.

In the experiments, we used training and validation sets for parameter and model selection, and an independent test set to assess the generalization performance of our methods. To perform signer independent experiments, we applied eight-fold, leave-one-subject-out cross validation and in each fold, we separated the examples of one subject as the test set. Figure 4 shows the experimental setup for the 8^{th} fold. For the rest of the seven subjects, we performed a seven-fold cross validation where in each fold, we put examples of one subject in the validation set, and the rest in the training set. For each classifier in the experiments, we performed seven trainings and obtained results on the validation set, where the average and standard deviation are reported. All the decisions for parameter selection, score space selection, etc. are given with respect to the accuracies on the validation set. The test set is completely independent and never used either during training or selection processes. For each training set in each fold, we also calculated the performance on the independent test set and report the average and standard deviation. Note that, as a result of this setup, each training set consists of examples from six subjects, since one subject is used in the validation set and one subject is used in the test set.

As the generative model, we trained a left-to-right continuous HMM for each sign. Therefore, 19 HMMs are trained. Each HMM has four states, and a single Gaussian density is used in each state. Fisher score spaces are calculated for each HMM and the discriminative classification is done via SVM. The SVM runs are performed with the LIBSVM toolbox [25]. LIBSVM uses one-vs-one strategy to perform multi-class classification. However, note that the multi-class classification strategy used by the SVMs is not related to the multi-class classification strategy used for the Fisher scores and is completely independent. We use the multi-class SVM as a black box and use the classification result in our multi-class strategy of combining Fisher scores. Before any training, for each strategy, the kernel type (RBF or linear) and the parameters are determined separately by cross-validation on the training set, over a set of parameter values (cost:10/100/100, epsilon:0.001/0.0001, gamma:0.1/0.001/0.0001). We use the probability outputs calculated by LIBSVM when a score level combination method, such as sum or product rule is used.

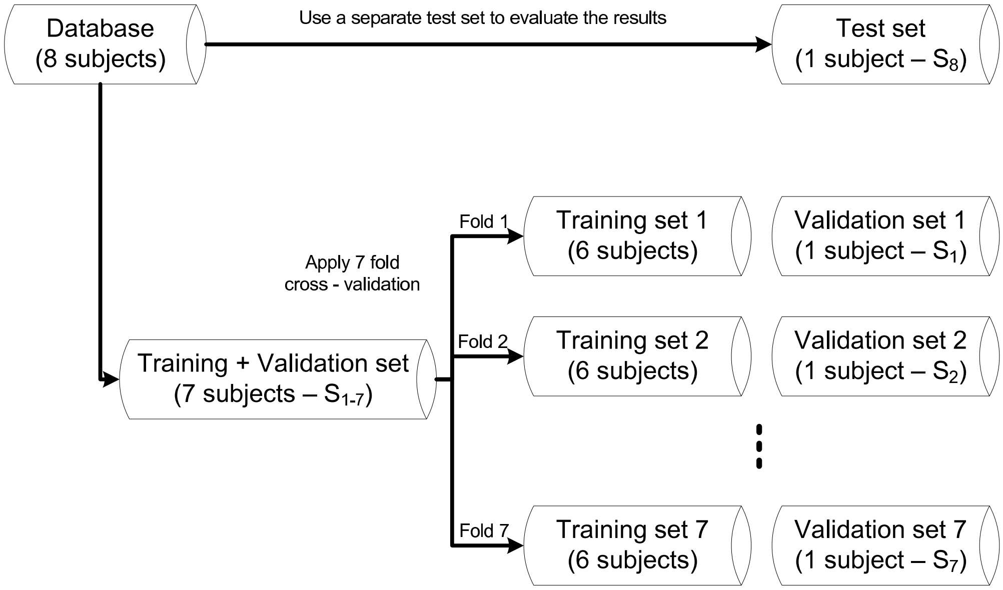


Fig. 4. Signer independent experimental setup for the 8^{th} fold in eight fold cross validation. In each fold, i, the examples of subject i are separated as the test set. The rest of the dataset is divided into training and validation sets via seven fold cross validation.

6.2 Comparison of Multiclass Strategies

Comparison of the different multi-class strategies, together with the performance of the underlying generative model are given in Table 5. The baseline accuracies, obtained by HMMs, on validation and test sets are 53,78% and 52,82% respectively. Note that this is a difficult dataSET and some clsses are indistinguishable [22]. As the combination method we used and compared sum and product rules, where necessary. The proposed method M_{DLC} outperforms all, demonstrating that using all score spaces to discriminate each class pair is an advantageous strategy. Instead of using binary classifiers, using multi-class classifiers on each score space increases the accuracy and provides a better strategy for multi-class classification. Note that the accuracy even drops down with some of the binary classifiers. Although M_{FLC} has a comparable performance with M_{DLC}, the memory requirement, proportional to the number of classes, is extremely high as a result of combining high dimensional score spaces into a single one. M_{DLC} not only has less memory requirement but also has a better accuracy than M_{FLC}: around 13% increase in the baseline accuracy is observed both on the validation set and test set, with sum combination rule. As this result is the highest accuracy on the validation set, M_{DLC} strategy with sum rule will be used in the following experiments.

The results in Table 5 are the averages of eight fold cross validation. We performed a paired t-test with 0.05 significance level and the results of M_{DLC}

Table 5
Comparison of different multi-class schemes. Average validation and test accuracies ((%)±std) are reported.

Baseline Algorithm

	Validation Set	Test Set
HMM	53,78 ± 7,78	52,82 ± 5,55

Fisher Score MultiClass Strategy

	Validation Set		Test Set	
	Sum	Product	Sum	Product
B_{1vs1}	61,37 ± 6,19	61,24 ± 6,58	61,65 ± 4,18	61,39 ± 4,78
B_{1vs1R}	59,53 ± 6,90	60,43 ± 6,42	61,13 ± 5,26	61,03 ± 5,13
B_{1vsALL}	54,08 ± 7,89	45,88 ± 8,01	54,66 ± 6,27	46,09 ± 5,89
M_{FLC}	61,35 ± 5,99		61,18 ± 4,82	
$\mathbf{M_{DLC}}$	**66,73 ± 7,80**	**66,18 ± 7,99**	**65,68 ± 3,84**	**65,71 ± 3,91**

Table 6
Detailed signer independent performance of the baseline model, HMM and M_{DLC} with sum rule. Average test accuracies for each subject ((%)±std) and the increase in the accuracy are reported. Statistically significant differences are shown in bold.

Test subject	HMM	M_{DLC}	Accuracy increase
# 1	52,93 ± 5,53	**67,67 ± 4,77**	14,74
# 2	63,46 ± 5,95	66,77 ± 2,57	3,31
# 3	51,13 ± 7,19	**67,22 ± 7,18**	16,09
# 4	56,24 ± 6,98	58,50 ± 2,64	2,26
# 5	47,37 ± 4,42	**58,05 ± 2,68**	10,68
# 6	48,87 ± 5,60	**75,64 ± 3,24**	26,77
# 7	47,82 ± 4,12	**57,74 ± 4,32**	9,92
# 8	54,74 ± 4,63	**73,83 ± 3,30**	19,10

are statisticall significantly better than both the baseline HMM and the other multi-class classification strategies. We also present the details of each fold for the best performing approach, M_{DLC} strategy with sum rule. Table 6 shows the average test accuracies of each fold in eight fold cross validation. The results show that using M_{DLC} we can obtain higher accuracies in each fold, in comparison to the baseline model, HMM. The increase in the accuracy varies from 2,3% to 26,8%. The statisticaly significant results are shown in bold.

6.3 Feature Selection and the Effect of HMM Parameters on the Classification Performance

Although one can use classifiers that do not suffer from the curse of dimensionality (such as SVMs), the dimensionality of the Fisher score space can be extremely high depending on the number of parameters of the underlying generative model and the input dimensionality. In this section, we investigate the effect of each HMM parameter on the classification accuracy. The computational cost of Fisher score calculation can be decreased by considering only the most important parameters.

With the HMM as the underlying generative model, the normalized Fisher likelihood score space is given in Equation 8. The number of features in this new feature space is:

$$F_{log\ p(O_t|\theta)} + F_{a_{ii}} + F_{\mu_{im}} + F_{\Sigma_{im}} + F_{w_{im}} \tag{10}$$

where F_p stands for the number of features extracted with respect to the parameter p. The number of features for each parameter is as follows:

$$\begin{aligned} F_{log\ p(O_t|\theta)} &= 1 \\ F_{a_{ii}} &= N - 1 \\ F_{\mu_{im}} &= NMV \\ F_{\Sigma_{im}} &= NMV^2 \\ F_{w_{im}} &= N(M-1) \end{aligned}$$

where N is the number of states, M is the number of mixtures and V is the input dimensionality.

Among the parameters of the HMM, the discriminative power of all parameter combinations are explored and the results are given in Table 7. The multi-class classifications are performed via the M_{DLC} strategy.

All combinations of feature sets are explored and the best single parameter result is obtained by (Σ) on the validation set, with an increase of 0.15% on the accuracy and the corresponding test set accuracy is a bit higher than the accuracy of the complete set. If the parameters are used alone, the results show that most discriminatory features are the derivatives of the component means, (μ), and covariances, (Σ). There are also other reduced sets with accuracies either equal or better than using the complete parameter set. The log-likelihoods on which the HMM decision is based are found to be less discriminative than expected. This result follows from the fact that each Fisher score space is processed independently with no regard to the relationship between

Table 7
Effect of HMM parameters on the recognition performance. The abbreviations refer to the score spaces: ll for $logp(O_t|\theta)$, a for $\nabla_{a_{ii}}$, μ for $\nabla_{\mu_{ik}}$, and Σ for $\nabla_{\Sigma_{ik}}$

Selected HMM Parameters	Validation Set Acc. (%) & Std	Test set Acc. (%) & Std
(ll, a, μ, Σ)	**66,73** ± **7,80**	**65,68** ± **3,84**
(ll, a, μ)	66,18 ± 7,16	66,32 ± 4
(ll, a, Σ)	**66,80** ± **7,94**	**65,62** ± **3,80**
(ll, μ, Σ)	66,73 ± 7,83	65,68 ± 3,83
(a, μ, Σ)	66,75 ± 7,82	65,68 ± 3,81
(ll, a)	49,91 ± 6,28	50,02 ± 4,11
(ll, μ)	66,32 ± 7,29	66,35 ± 4
(ll, Σ)	66,88 ± 8,01	65,62 ± 3,86
(a, μ)	66,33 ± 7,77	66,62 ± 3,72
(a, Σ)	**66,88** ± **7,98**	**65,70** ± **3,86**
(μ, Σ)	66,77 ± 7,82	65,68 ± 3,80
(ll)	38,63 ± 7,33	38,55 ± 4,05
(a)	47,16 ± 6,96	46,90 ± 4,22
(μ)	66,45 ± 7,42	66,50 ± 3,75
(Σ)	**66,88** ± **7,95**	**65,71** ± **3,83**

log-likelihoods of the HMMs of each class. This relationship is apparently lost in the normalization process of the score spaces since the normalization of each score space is independent of others.

6.4 Dimensionality Reduction of Fisher Scores

We can reduce the dimensionality of the new feature space, by applying state-of-the art dimensionality reduction techniques. We compare two techniques, PCA and LDA, where the former aims to maximize the variance in the features and the latter aims to maximize the class separability in the new feature space. For PCA, we evaluated the performance of using different proportions of variance explained: 90%, 95%, and 99%. When applying LDA, we used the reduced feature space found by PCA and further reduced the dimensionality to 18 ($\# \ of \ classes - 1$), which is the maximum dimensionality that can be

Table 8
Dimensionality reduction

Var. explained	Validation Set		Test Set	
/ # dims	PCA	LDA	PCA	LDA
0.90/ ∼55	65,17 ± 7,82	64,59 ± 6,23	64,64 ± 4,05	63,80 ± 4,32
0.95 / ∼75	64,38 ± 7,23	65,85 ± 6,48	64,12 ± 4,06	65,73 ± 4,17
0.99 / ∼150	66,48 ± 8,95	**67,65 ± 7,93**	66,17 ± 4,36	**67,54 ± 3,71**

achieved with LDA.

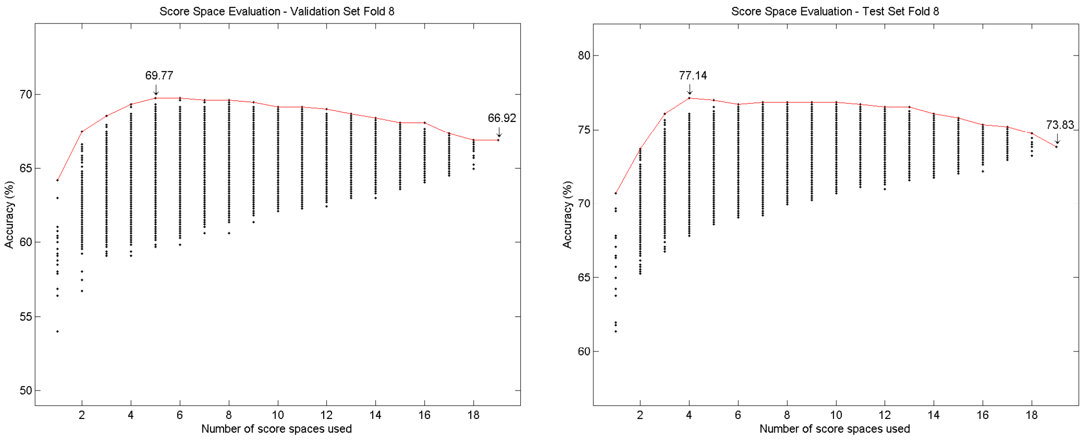
The results are given in Table 8. The first column shows the proportion of variance explained and corresponding average number of PCA dimensions. In general, accuracies obtained by PCA and LDA are comparable. Best result on the validation set is obtained by LDA with 99% proportion of variance explained. With only 18 dimensions, we can achieve higher accuracies both in validation and test sets (1% increase in the validation set and 2% increase in the test set).

6.5 Score Space Selection

In the M_{DLC} strategy, each single classifier is capable of making a multi-class decision and their decisions are combined at the decision or score level to obtain an improved accuracy. Experiments show that sometimes a small subset of classifiers, or even a single classifier, may perform equally well (see Table 3). Hence our aim is to find techniques to select a subset out of K Fisher score mappings and use only the classifiers based on this subset.

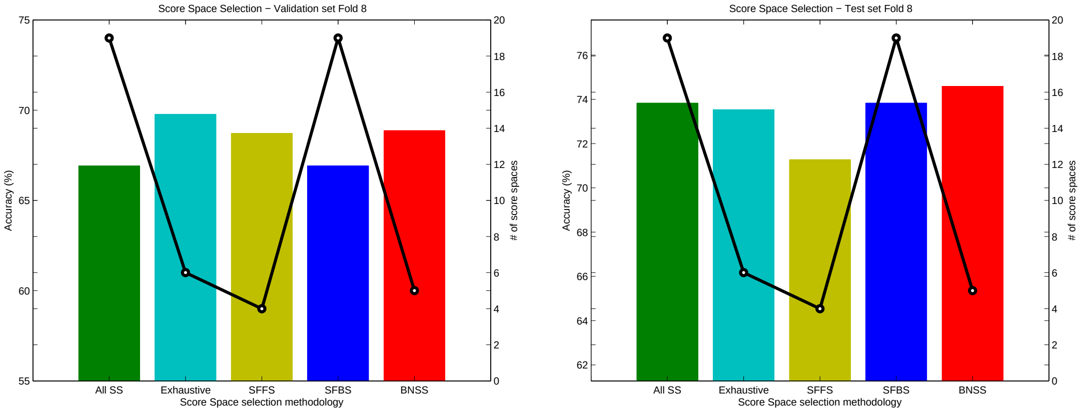
We first run an exhaustive search for all the possible combinations of the score spaces. Figure 5 shows the result of the exhaustive search on the validation and test sets of Fold 8. With the 19 classes in the eNTERFACE dataset, one can have $2^{19} - 1$ possible subsets. The highest accuracy is obtained by using only five and four score spaces on validation and test sets, with an accuracy increase of 2% and 3.5% respectively, when compared to using all of the score spaces. Moreover, both in validation and test sets, we see that there are many subsets that have equal or better accuracy than all score spaces.

Since exhaustive search is impractical for high number of classes, we implemented Sequential Floating Forward and Backward Search (SFFS, SFBS) strategies and also selecting the best of N score spaces (BNSS). We applied these techniques on the validation set and obtained the score space subset selected by each technique. To evaluate the performance on the independent test set, we used the subsets selected on the validation set and calculate its



(a) (b)

Fig. 5. Exhaustive search over all score space combinations in Fold 8: (a) Validation set, (b) Test set.

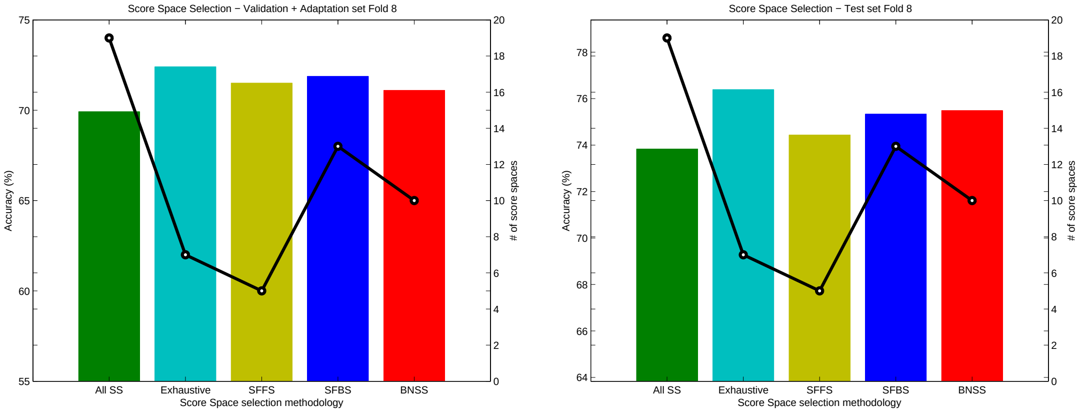


(a) (b)

Fig. 6. Score space selection performances in Fold 8: (a) Validation set accuracies of score space subsets and the number of score spaces used, (b) Test set accuracies of score space subsets selected on validation set and the number of score spaces used. The bars show the accuracies of each selection method and the bold line shows the number of score spaces used.

accuracy on the test set.

We give the score space selection results of Fold 8 in Figure 6. Although the subsets found on the validation set are higher than the accuracy of using all score spaces, these subsets do not always generalize well to the test set. When we apply an exhaustive search directly on the test set, as shown in Figure 5b, we see that there are subsets that have better accuracy than all score spaces. However, these subsets can not be found by the score space selection techniques performed on the validation set.



(a)

(b)

Fig. 7. Signer adaptation results in Fold 8: (a) Adapted validation set accuracies of score space subsets and the number of score spaces used, (b) Test set accuracies of score space subsets selected on the adapted validation set and the number of score spaces used. The bars show the accuracies of each selection method and the bold line shows the number of score spaces used.

6.6 Signer Adaptation

In the previous section, we see that the score space subsets selected on the validation set do not generalize well to the test set, which contains examples of an unseen signer. Due to the presence of significant inter-person differences, the system needs to be adapted to the new signer to achieve better performance [26,27]. We propose a signer adaptation scheme, which is performed during the score space selection phase. Our scheme uses the trained system as is and uses only a small part of data from the new signer to select a better score space subset that generalizes to the new signer. We use three randomly selected examples per sign for the adaptation and apply the score space selection on the validation and adaptation data jointly. The accuracy of each score space subset is calculated as follows

$$Acc = w * Acc_{Va} + (1 - w) * Acc_{Ad} \tag{11}$$

where the total accuracy (Acc) of a subset is the weighted sum of the accuracies on the validation (Acc_{Va}) and adaptation (Acc_{Ad}) sets. We use $w = 0.5$ and give equal importance to validation and adaptation sets.

Figure 7 shows the score space selection accuracies for fold 8 when the signer adaptation scheme is applied. The results show that by applying signer adaptation, all of the score space selection techniques are able to find smaller score space subsets with better accuracy than using all of the score spaces. Although the accuracy of the subsets found by SFFS, SFBS, and BNSS are not as high

Table 9
Score space selection results. Test accuracy (% acc) of each method and the number of score spaces used (# ss) with and without signer adaptation are given.

	Test	All		Exhaustive		SFFS		SFBS		BNSS	
	subject	% acc	# ss	% acc	# ss	%	# ss	%	# ss	%	# ss
Without adaptation	# 1	67,67	19	68,72	7	67,82	5	67,52	14	66,92	17
	# 2	66,77	19	67,37	4	67,22	4	66,02	15	64,36	3
	# 3	67,22	19	61,50	3	63,16	3	66,47	12	62,11	2
	# 4	58,50	19	59,40	5	59,40	5	58,50	19	59,70	5
	# 5	58,05	19	59,40	7	58,65	7	57,74	16	58,95	8
	# 6	75,64	19	74,29	4	74,29	4	75,19	13	73,83	5
	# 7	57,74	19	60,00	8	59,85	3	57,44	17	58,05	16
	# 8	73,83	19	73,53	6	71,28	4	73,83	19	74,59	5
	Avg	**65,68**	**19**	**65,53**	**5,50**	**65,21**	**4,38**	**65,34**	**15,63**	**64,81**	**7,63**
With adaptation	# 1	67,67	19	69,77	5	68,87	4	68,87	15	68,72	6
	# 2	66,77	19	68,12	8	66,92	4	67,67	17	67,52	6
	# 3	67,22	19	71,58	5	71,58	4	68,27	16	70,83	6
	# 4	58,50	19	60,30	6	63,31	3	60,15	12	61,05	5
	# 5	58,05	19	61,95	9	61,95	6	59,55	15	60,45	4
	# 6	75,64	19	76,09	6	76,39	7	76,69	15	77,29	10
	# 7	57,74	19	63,31	4	63,31	4	60,75	10	62,11	5
	# 8	73,83	19	76,39	7	74,44	5	75,34	13	75,49	10
	Avg	**65,68**	**19**	**68,44**	**6,25**	**68,35**	**4,63**	**67,16**	**14,13**	**67,93**	**6,50**

as that of the exhaustive search, it is still higher than the accuracy of all score spaces. SFFS selects the smallest sized subset, with five score spaces, with an accuracy of 74.44%, in comparison to using all 19 score spaces, which yields an accuracy of 73.83%.

Overall results for all folds are given in Table 9. We see that without signer adaptation, none of the subset selection methods is able to find a subset that has a better accuracy than using all score spaces. However SFFS finds subsets with an average accuracy of 65.21% with only using around four score spaces in average. This is a very similar accuracy in comparison to using all score spaces, 65.68%, but with a significant decrease on the number of score spaces. When we apply signer adaptation, we are able to obtain higher accuracies with all of the methods. The smallest sized subset is found by SFFS with an average accuracy of 68.35%, using around five score spaces in average, which is almost the same accuracy with that of the exhaustive search, 68,44%, with even fewer score spaces.

In comparison to our baseline test accuracy, 52.82%, by applying Fisher score extraction, M_{DLC} multi-class classification strategy and score space selection with signer adaptation, we are able to obtain an average accuracy of 68.35%.

6.7 Experiments on other datasets and summary of the results

In this section we present an overall summary and discussion of the results together with experiments on other datasets, to show the generalization of the proposed techniques. We perform the experiments on two additional databases:

TWOHAND Idiap Hand Gesture Database: TWOHAND is a hand gesture database, with seven two-handed gestures to manipulate 3D objects. The training set contains 280 examples recorded from four people and the test set contains 210 examples recorded from three different people, with 10 examples per subject. More information on the database can be found in [28]. We extracted features for hand position in each frame (two features per hand per camera), and for hand shape, which is modelled with a simple ellipse and the lengths of the ellipse axes and the rotation angle are used as hand shape features (three features per hand per camera). The feature dimensionality is 20 per frame. More information on feature extraction can be found in [14].

For the experimental setup, we performed leave-one-subject-out cross validation on the training set and tested on the test set. Since the subjects in training and test sets are different, this is again a subject independent setup. We used a left-to-right continuous HMM with 4 states and a single Gaussian per state.

BUHMAP Turkish Non-manual signals database: BUHMAP is a database of non-manual signals, with seven gestures (head movements and facial expressions) used in Turkish sign language. The database contains 210 examples recorded from four people, with 5 examples per subject. As features, we use the x,y coordinates of automatically tracked 52 facial landmarks, normalized and postprocessed to get smoother trajectories. The feature dimensionality is 104 per frame. More information on the database and feature extraction can be found in [29,30].

We followed the same experimental setup in which we performed leave-one-subject-out cross validation on the whole dataset to separate the test set and again a leave-one-subject-out cross validation on the remaining set to get the training and validation sets. This set up provides us four different test sets and for each test set 3 different training-validation set pairs. For generative modeling, we used a left-to-right continuous HMM with 6 states and mixtures of two Gaussians per state with diagonal covariance matrix.

The results on TWOHAND and BUHMAP datasets with different multi-class classification strategies, together with the baseline HMM accuracy are shown in Table 10. The combination of scores in multi-class strategies is done by sum rule. The results on the eNTERFACE dataset are also shown for comparison purposes. Statistically significant results (significance level = 0.05) when compared to the baseline and other multi-class strategies are shown in bold.

Table 10
Average test accuracies (%) of different multi-class schemes on eNTERFACE, TWOHAND and BUHMAP datasets.

	eNTERFACE	TWOHAND	BUHMAP
baseline - HMM	52.82	98.8	66.67
B$_{1vs1}$	61.65	95.36	65.24
B$_{1vs1R}$	61.13	92.74	66.19
B$_{1vsAll}$	54.66	91.55	66.67
M$_{FLC}$	61.18	99.52	67.38
M$_{DLC}$	**66.68**	99.64	**73.57**

On eNTERFACE and BUHMAP datasets, accuracy of M$_{DLC}$ is statistically significantly higher than the other techniques. On TWOHAND dataset, since the baseline accuracy is already quite high, there is no statistical significance as a result of the ceiling effect.

Table 11 shows the results of the score space selection with and without adaptation. Statistically significant results of non-exhaustive score space selection strategies are shown in bold. The bold values in the number of score spaces show the strategy with a statistically significant result and the least number of score spaces.

The results show that

- By performing score space selection, a subset of score spaces can be found that gives a good enough accuracy with fewer number of score spaces
- Score space selection techniques, SFFS, SFBS, BNSS, provide comparable performance to that of exhaustive search.
- SFBS gives the best results but with a trade off: using larger subsets. BNSS, a very light weight score space selection method, gives consistently high accuracies with smaller subsets.
- Proposed signer adaptation technique significantly increases the accuracy when compared to the accuracies without signer adaptation.

7 Conclusions

HMMs provide a good framework for recognizing hand gestures, by handling translation and scale variances and by modeling and processing variable length sequence data. However, performance can be increased by combining HMMs with discriminative models which are more powerful in classification problems.

Table 11

Score space selection results on eNTERFACE, TWOHAND and BUHMAP datasets. Test accuracy (% acc) of each method and the number of score spaces used (# ss) with and without signer adaptation are given.

		eNTERFACE		TWOHAND		BUHMAP	
		Test %	# of ss	Test %	# of ss	Test %	# of ss
w/o adaptation	Exhaustive	65.53	5.50	99.17	4.00	69.52	3.67
	SFFS	**65.21**	**4.38**	96.79	2.00	66.67	2.00
	SFBS	**65.34**	15.63	98.33	5.00	**71.75**	**5.67**
	BNSS	**64.81**	7.63	98.33	4.00	**73.02**	6.67
w/ adaptation	Exhaustive	68.44	6.25	99.29	4.00	72.22	3.67
	SFFS	**68.35**	**4.63**	96.43	2.00	69.05	2.00
	SFBS	**67.16**	14.13	99.05	5.00	**74.60**	**5.67**
	BNSS	**67.93**	6.5	99.05	4.00	**76.98**	6.67

Fisher kernels are suitable for combining generative models with discriminative classifiers and theoretically, the resulting combined classifier has the powers of both approaches and has a better classification accuracy. However, as Fisher kernels are intrinsically binary, a multi-class strategy must be defined properly in order to achieve high recognition accuracies for multi-class classification problems such as gesture and sign recognition.

In this study, we proposed a multi-class classification strategy for Fisher scores. The main idea of our multi-class classification strategy is to use the Fisher score mapping of one model in the classification process for all of the classes. As a result, each mapping is able to discriminate all the classes up to some degree. When all of these mappings are combined, higher accuracies are obtained when compared to the existing multi-class classification approaches in the literature.

As the dimensionality of the Fisher scores is high, we applied several dimensionality reduction techniques on each of the Fisher score mappings and see that it is able to reduce the dimensionality without any decrease in the accuracy. Moreover, we show that we can obtain similar or better accuracies if we combine only a subset of the Fisher score mappings. We compare several score space selection strategies and see that the SFFS strategy finds the smallest sized subsets with performance comparable to that of exhaustive search or all score spaces. The selected subset generalizes to a new signer only when a signer adaptation scheme is applied. We present here a signer adaptation scheme which is able to adapt the system and achieve better performance with only a few number of examples of the new signer.

We have tested our proposed technique, the feature selection idea and the user adaptation scheme on a sign language dataset that is difficult due to the presence of very similar signs. We have seen that performance increases significantly when compared with the baseline model and the other multi-class classification strategies. Also the results with the score space selection show that without a significant decrease in the accuracy, we are able to reduce the computational cost. We have further tested the techniques with two additional datasets: A hand gesture dataset and a sign language expression dataset. We have observed similar gains and have successfully demonstrated that the proposed techniques generalize to other problems.

References

[1] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, in: Proceedings of the IEEE, Vol. 77 of no. 2, 1989, pp. 257–285.

[2] Y. Wu, T. S. Huang, Hand modeling, analysis, and recognition for vision based human computer interaction, IEEE Signal Processing Magazine 21 (1) (2001) 51–60.

[3] S. C. W. Ong, S. Ranganath, Automatic sign language analysis: A survey and the future beyond lexical meaning., IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 873–891.

[4] X. He, L. Deng, W. Chou, Discriminative learning in sequential pattern recognition, Signal Processing Magazine, IEEE 25 (5) (2008) 14–36.

[5] T. G. Dietterich, Machine learning for sequential data: A review, in: Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Springer-Verlag, 2002, pp. 15–30.

[6] T. S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Conference on Advances in Neural Information Processing Systems II, MIT Press, 1998, pp. 487–493.

[7] N. Smith, M. Gales, Speech recognition using SVMs, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems., Vol. 14, MIT Press, 2002.

[8] T. Jaakkola, M. Diekhans, D. Haussler, Using the fisher kernel method to detect remote protein homologies, in: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1999, pp. 149–158.

[9] P. Moreno, R. Rifkin, Using the fisher kernel method for web audio classification, in: IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP '00, Vol. 6, 2000, pp. 2417–2420.

[10] A. D. Holub, M. Welling, P. Perona, Combining generative models and Fisher kernels for object recognition, in: Tenth IEEE International Conference on Computer Vision (ICCV'05), Vol. 1, IEEE Computer Society, Washington, DC, USA, 2005, pp. 136–143.

[11] L. Chen, H. Man, Hybrid IMM/SVM approach for wavelet-domain probabilistic model based texture classification, IEE Proceedings of Vision, Image and Signal Processing 152 (6) (2005) 724–730.

[12] L. Chen, H. Man, A. V. Nefian, Face recognition based on multi-class mapping of Fisher scores, Pattern Recognition 38 (2005) 799–811.

[13] O. Aran, L. Akarun, Multi-class classification strategies for fisher scores of gesture and sign sequences, in: International Conference On Pattern Recognition, 2008.

[14] O. Aran, L. Akarun, Recognizing two handed gestures with generative, discriminative and ensemble methods via Fisher kernels, in: Multimedia Content Representation, Classification and Security International Workshop, MRCS 2006, Istanbul, Turkey, Proceedings, Vol. LNCS 4015, 2006, pp. 159–166.

[15] B. Scholkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2001.

[16] N. Smith, M. Gales, Using SVMs to classify variable length speech patterns, Tech. rep., Cambridge University Engineering Department (2002).

[17] N. Liu, B. C. Lovell, P. J. Kootsookos, R. I. A. Davis, Model structure selection and training algorithms for an HMM gesture recognition system, in: Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 100–105.

[18] O. Aran, Vision based sign language recognition: Modeling and recognizing isolated signs with manual and non-manual components, Ph.D. thesis, Bogazici University, Istanbul, Turkey (2008).

[19] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (2004) 101–141.

[20] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, Inc., 2001.

[21] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.

[22] O. Aran, I. Ari, A. Benoit, P. Campr, A. H. Carrillo, F.-X. Fanard, L. Akarun, A. Caplier, B. Sankur, Signtutor: An interactive system for sign language tutoring, IEEE Multimedia 16 (1) (2009) 81–93.

[23] T. Burger, A. Urankar, O. Aran, L. Akarun, A. Caplier, Cued speech hand shape recognition, in: 2nd International Conference on Computer Vision Theory and Applications (VISAPP'07), Spain, 2007.

[24] O. Aran, T. Burger, A. Caplier, L. Akarun, A belief-based sequential fusion approach for fusing manual and non-manual signs, Pattern Recognition 42 (5) (2009) 812–822.

[25] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm (2001).

[26] S. C. Ong, S. Ranganath, Y. Venkatesha, Understanding gestures with systematic variations in movement dynamics, Pattern Recognition 39 (9) (2006) 1633–1648.

[27] S. C. W. Ong, S. Ranganath, Deciphering gestures with layered meanings and signer adaptation., in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), 2004, pp. 559–564.

[28] A. Just, S. Marcel., Two-handed gesture recognitio, Tech. Rep. 24, Idiap (2005). URL `http://www.idiap.ch/resources/twohanded/`

[29] O. Aran, I. Ari, A. Guvensan, H. Haberdar, Z. Kurt, I. Turkmen, A. Uyar, L. Akarun, A database of non-manual signs in turkish sign language, in: IEEE 15th Signal Processing and Communications Applications (SIU'07), 2007.

[30] I. Ari, Facial feature tracking and expression recognition for sign language, Master's thesis, Bogazici University (2008).