Recognizing Two Handed Gestures with Generative, Discriminative and Ensemble Methods via Fisher Kernels

Oya Aran, Lale Akarun

Dept. of Computer Engineering, Bogazici University,
34342, Istanbul, Turkey
{aranoya,akarun}@boun.edu.tr

Abstract. Use of gestures extends Human Computer Interaction (HCI) possibilities in multimodal environments. However, the great variability in gestures, both in time, size, and position, as well as interpersonal differences, makes the recognition task difficult. With their power in modeling sequence data and processing variable length sequences, modeling hand gestures using Hidden Markov Models (HMM) is a natural extension. On the other hand, discriminative methods such as Support Vector Machines (SVM), compared to model based approaches such as HMMs, have flexible decision boundaries and better classification performance. By extracting features from gesture sequences via Fisher Kernels based on HMMs, classification can be done by a discriminative classifier. We compared the performance of this combined classifier with generative and discriminative classifiers on a small database of two handed gestures recorded with two cameras. We used Kalman tracking of hands from two cameras using center-of-mass and blob tracking. The results show that (i) blob tracking incorporates general hand shape with hand motion and performs better than simple center-of-mass tracking, and (ii) in a stereo camera setup, even if 3D reconstruction is not possible, combining 2D information from each camera at feature level decreases the error rates, (iii) Fisher Score methodology combines the powers of generative and discriminative approaches and increases the classification performance.

1 Introduction

The use of gestures in HCI is a very attractive idea: Gestures are a very natural part of human communication. In environments where speech is not possible, i.e, in the hearing impaired or in very noisy environments, they can become the primary communication medium, as in sign language [1]. Their use in HCI can either replace or complement other modalities [2, 3]. Gesture recognition systems model spatial and temporal components of the hand. Spatial component is the hand posture or general hand shape depending on the type of gestures in the database. Temporal component is obtained by extracting the hand trajectory using hand tracking techniques or temporal template based methods, and the extracted trajectory is modeled with several methods such as Finite

State Machines (FSM), Time-delay neural networks (TDNN), HMMs or template matching [4]. Among these algorithms, HMMs are used most extensively and have proven successful in several kinds of systems.

There have been many attempts to combine generative models with discriminative classifiers to obtain a robust classifier which has the strengths of each approach. In [5], Fisher Kernels are proposed to map variable length sequences to fixed dimension vectors. This idea is further extended in [6] to the general idea of score-spaces. Any fixed length mapping of variable length sequences enables the use of a discriminative classifier. However, it is typical for a generative model to have many parameters, resulting in high-dimensional feature vectors. SVM is a popular choice for score spaces with its power in handling high dimensional feature spaces. Fisher scores and other score spaces have been applied to bioinformatics problems [5], speech recognition [6], and object recognition [7]. The application of this idea to hand gesture recognition is the subject of this paper. We have used Kalman blob tracking of two hands from two cameras and compared the performance of generative, discriminative and combined classifiers using Fisher Scores on a small database of two handed gestures. Our results show that enhanced recognition performances are achievable by combining the powers of generative and discriminative approaches using Fisher scores.

2 Fisher Kernels and Score Spaces

A kernel function can be represented as an inner product between feature vectors:

$$K(X_i, X_j) = < \phi(X_i), \phi(X_j) > \tag{1}$$

where ϕ is the mapping function that maps the original examples, X, to the feature vectors in the new feature space. By choosing different mapping functions, ϕ, one has the flexibility to design a variety of similarity measures and learning algorithms. A mapping function that is capable of mapping variable length sequences to fixed length vectors enables the use of discriminative classifiers for variable length examples. Fisher kernel [5] 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. *Fisher Score*, U_X, is defined as the gradient of the log likelihood with respect to the parameters of the model:

$$U_X = \nabla_\theta log P(X|\theta) \tag{2}$$

The unnormalized Fisher Kernel, U_X, is defined using Fisher Scores as the mapping function. This form of the Fisher Kernel can be used where normalization is not essential. In [5], Fisher Information Matrix is used for normalization. In this work, we normalized the score space using the diagonal of the covariance matrix of the score space estimated from the training set.

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 a discriminative classifier of any choice. However, the dimensionality of this new feature space can be high when the underlying generative model consist of many parameters and original feature space is multivariate. Thus, SVMs becomes a good choice of a classifier since they do not suffer from curse of dimensionality.

2.1 Fisher Kernel based on HMMs

In 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 [8] for gesture recognition problems.

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, which are modelled by mixture of K multivariate Gaussians:

$$b_i(O_t) = \sum_{k=1}^{K} w_{ik} N(O_t; \mu_{ik}, \Sigma_{ik})$$ (3)

where O_t is the observation at time t and w_{ik}, μ_{ik}, Σ_{ik} are weight, mean and covariance of the Gaussian component k at state i.

For a left-to-right HMM, 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_{ik}, mean, μ_{ik} and covariance, Σ_{ik} of each Gaussian component. The first order derivatives of the loglikelihood, $P(O|\theta)$ with respect to each parameter are given below:

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

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

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

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

where $\gamma_i(t)$ is the posterior of state i at time t and $\gamma_{ik}(t)$ is the posterior probability of component k of state i at time t. 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.

These gradients are concatenated to form the new feature vector which is the Fisher score. More information on these gradients and several score spaces can be found in [6]. We have used the loglikelihood score space where loglikelihood itself is also concatenated to the feature vector (Equation 8).

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

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* [6] for normalizing variable length gesture trajectories by using normalized component posterior probabilities, $\hat{\gamma}_{ik}(t) = \frac{\gamma_{ik}(t)}{\sum_{t=1}^{T} \gamma_i(t)}$, in the above gradients.

3 Recognition of Two Handed Gestures

We have worked on a small gesture dataset, with seven two-handed gestures to manipulate 3D objects [9]. The gestures are a push gesture and rotate gestures in six directions: back, front, left, right, down, up. Two cameras are used, positioned on the left and right of the user. The users wear gloves: a blue glove on the left and a yellow glove on the right hand. The training set contains 280 examples recorded from four people and the test set contains 210 examples recorded from three different people. More information on the database can be found in [9].

3.1 Hand Segmentation and Tracking

The left and right hands of the user are found by thresholding according to the colors of the gloves. Thresholded images are segmented using connected components labelling (CCL), assuming that the component with the largest area is the hand. Then a region growing algorithm is applied to all pixels at the contour of selected component to find the boundary of the hand in a robust fashion (Figure 1). The thresholds are determined by fitting a 3D-Gaussian distribution in HSV color space by selecting a sample from the glove color. The thresholds are recalculated at each frame which makes hand segmentation robust to lighting and illumination changes. Following the hand segmentation step, a single point on the hand (center-of-mass) or the whole hand as a blob is tracked and smoothed using Kalman Filtering. Blob tracking provides features that represent the general hand shape. An ellipse is fitted to the hand pixels and center-of-mass (x,y), size (ellipse width and height) and the orientation (angle) of the ellipse are calculated at each frame for each hand.

In this camera setup, one hand may occlude the other in some frames. However, when occlusion occurs in one camera, the occluded hand can be located clearly in the other camera (Figure 2). The assumption of the hand detection algorithm is that the glove forms the largest component with that color in the camera view. In case of occlusion, as long as this assumption holds, the center-of-mass and the related blob can be found with a small error which can be tolerated by the Kalman filter. Otherwise, the component and its center of mass found
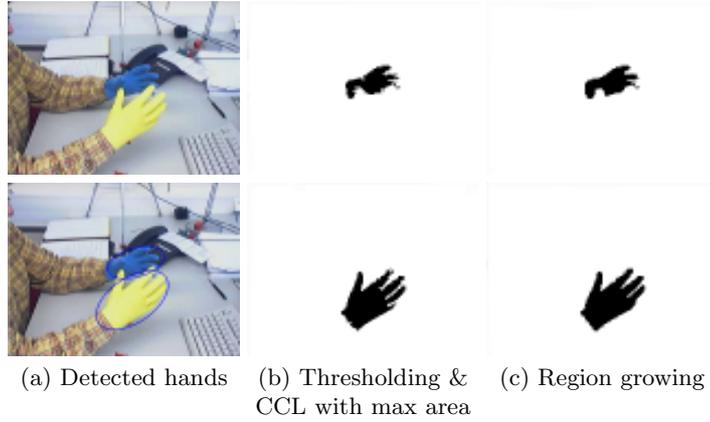
| (a) Detected hands | (b) Thresholding & CCL with max area | (c) Region growing |

Fig. 1. Hand detection

by the algorithm has no relevance to the real position of the hand. If these false estimates are used to update Kalman Filter parameters, the reliability of the Kalman Filter will decrease. Therefore, when the area of the component found by the algorithm is less than a threshold, parameters of the Kalman Filter are not updated. If total occlusion only lasts one or two frames, which is the case for this database, Kalman Filter is able to make acceptable estimates.

Left Cam   Right Cam

Fig. 2. Frames with occlusion

3.2 Normalization

Translation and scale differences in gestures are normalized to obtain invariance. Rotations are not normalized since rotation of the trajectory provides a discrimination among different classes. The normalized trajectory coordinates, $((x'_1, y'_1), \ldots, (x'_t, y'_t), \ldots, (x'_N, y'_N))$, s.t. $0 \leq x'_t, y'_t \leq 1$, are calculated as follows:

$$x'_t = 0.5 + 0.5\frac{x_t - x_m}{\delta} \qquad y'_t = 0.5 + 0.5\frac{y_t - y_m}{\delta} \qquad (9)$$

where x_m and y_m are the mid-points of the range in x and y coordinates respectively and δ is the scaling factor which is selected to be the maximum of the spread in x and y coordinates, since scaling with different factors affects the shape. In blob tracking, apart from the center-of-mass, size of the blob (width and height) is also normalized using the maximum of the spread in width and height as in Eqn 9. The angle is normalized independently.

4   Experiments

For each gesture, four different trajectories are extracted for each hand at each camera: left and right hand trajectory from Camera 1 (*L1* and *R1*), and Camera 2 (*L2* and *R2*). Each trajectory contains the parameters of a hand (center-of-mass, size and angle of blob) in one camera. Hands may occlude each other in a single camera view. Therefore, a trajectory from a single camera may be erroneous. Moreover, by limiting the classifier with single camera information, the performance of the classifier is limited to 2D motion. Although there are two cameras in the system, it is not possible to accurately extract 3D coordinates of the hands for two reasons: the calibration matrix is unknown, and the points seen by the cameras are not the same. One camera views one side of the hand and the other camera views the opposite side. However, even without 3D reconstruction, the extra information can be incorporated into the system by combining information from both cameras in the feature set. We prepared the following schemes to show the effect of the two-camera setup:

	Setup	Feature vector Size
L1R1	Left & right hands from Cam1	4 in CoM, 10 in Blob tracking
L2R2	Left & right hands from Cam2	4 in CoM, 10 in Blob tracking
L1L2R1R2	Both hands from both cameras	8 in CoM, 20 in Blob tracking

Following the above schemes, three classifiers are trained: HMM, SVM with re-sampled trajectories (trajectories are re-sampled to 12 points using spatial re-sampling with linear interpolation – in blob tracking, the size and angle of the re-sampled point are determined by the former blob in the trajectory), and SVM with Fisher Scores based on HMMs (with sequence length normalization and score space normalization with diagonal approximation of covariance matrix). Normalized trajectories are used in each classifier. A Radial Basis Function (RBF) kernel is used in SVM classifiers. In HMMs, a left-to-right model with no skips is used with four states and one Gaussian component in each state. Baum-Welch algorithm is used to estimate the transition probabilities and mean and variance of the Gaussian at each state. It is observed that increasing the number of states or number of Gaussian components does not increase the accuracy. For each gesture, an HMM is trained and for each trained HMM, an SVM with Fisher Scores is constructed. Fisher Scores are further z-normalized and outliers are truncated to two standard deviations around the mean. The parameters of each classifier are determined by 10-fold cross validation on the training set. In

each scheme, HMMs and related SVMs are trained 10 times. For SVMs with re-sampled trajectories single training is performed. Results are obtained on an independent test set and mean and standard deviations are given in Table 1.

Table 1. Test errors and standard deviations

Dataset	SVM	HMM	SVM (Fisher)
CoM			
L1R1 *(cam1)*	95.20 ± 0.000	95.14 ± 0.89	95.10 ± 1.95
L2R2 *(cam2)*	95.70 ± 0.000	96.10 ± 0.44	95.24 ± 1.02
L1R1L2R2	97.14 ± 0.000	98.38 ± 0.46	97.52 ± 0.80
Blob			
L1R1 *(cam1)*	98.57 ± 0.000	98.57 ± 0.32	98.05 ± 0.57
L2R2 *(cam2)*	97.14 ± 0.000	97.52 ± 0.80	98.29 ± 0.68
L1R1L2R2	99.00 ± 0.000	99.00 ± 0.61	$\mathbf{99.57 \pm 0.61}$

For SVM runs, LIBSVM package is used [10]. For each example, Fisher Scores of each HMM are calculated. Fisher Scores calculated from HMM_i are given as input to SVM_i, where SVM_i is a multiclass SVM. Thus, seven multiclass SVMs are trained on the scores of seven HMMs, and outputs of each SVM are combined using majority voting to decide the final output (Figure 3). One-vs-one methodology is used in muticlass SVMs.
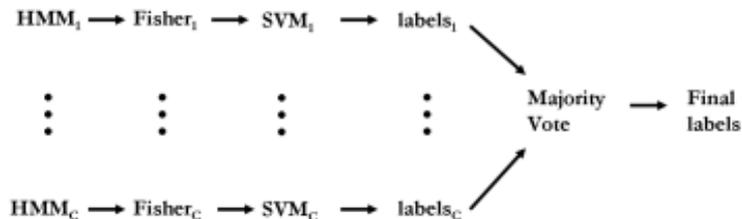

Fig. 3. Combining Fisher Scores of each HMM in SVM training

It can be seen that performance of SVMs with re-sampled trajectories are slightly lower than the other classifiers, which is an expected result since unlike HMMs, the sequential information inherent in the trajectory is not fully utilized in SVM training. However, when combined with a generative model, using Fisher Scores, error rates tend to decrease in general. An exception to these observations is in L1R1 feature set of CoM tracking where the best result is obtained

with re-sampled trajectories. Blob tracking decreases the error rates about 50% in comparison to center-of-mass tracking. A similar decrease in error rates is observed when information from both cameras is used. The best result is obtained by two camera information in blob tracking and using Fisher Scores, in which we have 99.57% accuracy in the test set.

5 Conclusion

HMMs provide a good framework for recognizing hand gestures, by modeling and processing variable length sequence data. However, their performance can be enhanced by combining HMMs with discriminative models which are more powerful in classification problems. In this work, this combination is handled via Fisher Scores derived from HMMs. These Fisher Scores are then used as the new feature space and trained using a SVM. The combined classifier is either superior to or as good as the pure generative classifier. This combined classifier is also compared to a pure discriminative classifier, SVMs trained with re-sampled trajectories. Our experiments on the recognition of two-handed gestures shows that transforming variable length sequences to fixed length via Fisher Scores transmits the knowledge embedded in the generative model to the new feature space and results in better performance than simple re-sampling of sequences.

References

1. Ong, S.C.W., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 873–891
2. Pavlovic, V., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 677–695
3. Heckenberg, D., Lovell, B.C.: MIME: A gesture-driven computer interface. In: Visual Communications and Image Processing, SPIE. Volume 4067., Perth, Australia (2000) 261–268
4. Wu, Y., Huang, T.S.: Hand modeling, analysis, and recognition for vision based human computer interaction. IEEE Signal Processing Magazine **21** (2001) 51–60
5. Jaakkola, T.S., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II, MIT Press (1998) 487–493
6. Smith, N., Gales, M.: Using SVMs to classify variable length speech patterns. Technical report, Cambridge University Engineering Department (2002)
7. Holub, A., Welling, M., Perona, P.: Combining generative models and fisher kernels for object class recognition. In: Int. Conference on Computer Vision. (2005)
8. Liu, N., Lovell, B.C., Kootsookos, P.J., Davis, R.I.A.: Model structure selection and training algorithms for a hmm gesture recognition system. In: International Workshop in Frontiers of Handwriting Recognition, Tokyo. (2004) 100–106
9. Marcel, S., Just, A.: (IDIAP Two handed gesture dataset) Available at http://www.idiap.ch/∼marcel/.
10. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.