Supervised Multi-Modal Action Classification

Robert W.H. Fisher
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
Andrew ID: rwfisher
rwfisher@cs.cmu.edu

Prashant P. Reddy
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
Andrew ID: preddy1
preddy1@cs.cmu.edu

Abstract

We consider multi-modal data from a first person video camera and from five inertial measurement units (IMUs) as they capture subjects making brownies. We analyze this data to extract a set of relevant features and identify algorithms that can help with two supervised learning tasks: (i) frame classification, and (ii) sequence classification. An inherent challenge in working with this data is the difficulty of generalization from one subject to another. Spriggs *et al* [3] provide baseline results that we build upon. We use features extracted using dimensionality reduction techniques such as gist [2] and PCA. We apply several supervised learning techniques and note varying performances. In particular, we highlight the role of effective smoothing techniques.

1 Introduction

The CMU Multi-Modal Activity (CMU-MMAC) database contains multiple measures of the human activity of subjects performing tasks involved in cooking and food preparation. This paper presents initial findings from our analysis of a subset of this data. In particular, we consider data from a first-person video camera and from five inertial measurement units (IMUs) as they capture subjects making brownies.

We analyze a set of extracted relevant features and identify algorithms that can help with two supervised learning tasks: (i) frame classification, and (ii) sequence classification. Frame classification is the problem where we identify the action being performed by the subject using a test sample of a single frame of data from the first person video and from the IMU sensors. This classification process is then repeated for every frame in the testing corpus. In sequence classification, the learning algorithm is tested on a sequence of continuous frames which are known to be encoding a common subject activity (e.g. cracking eggs). A frame classification algorithm allows the reconstruction of an entire set of subject activities without requiring any prior knowledge. The sequence classification problem requires the starting and ending time of every activity to be known, but a sequence classification algorithm will produce more accurate predictions. In our work, we have tried several classifiers, including Naive Bayes, Neural Networks, Support Vector Machines (SVMs), Hidden Markov Models(HMMs), and K-Nearest Neighbors (K-NN) algorithms. Our most effective classification method was a combination of SVMs, K-NN, and the forward-backward HMM algorithm all working in unison. In particular, we employ a variant of the combined K-NN/SVM algorithm devised by Zhang *et al*[4].

2 Multi-Modal Data

The CMU-MMAC database was collected in Carnegie Mellon's Motion Capture Lab[1]. A kitchen was built and forty subjects have been recorded cooking five different recipes: brownies, pizza, sandwich, salad, and

[1] The data used in this paper was obtained from kitchen.cs.cmu.edu and the data collection was funded in part by the National Science Foundation under Grant No. EEEC-0540865.

scrambled eggs. Multi-modal data was recorded from five static video cameras, one wearable video camera, five microphones, a motion capture system, five wired IMUs, four wireless IMUs, a galvanic skin response sensor, and an RFID bracelet. We use the video data from the wearable camera and the wired IMU data in our analysis primarily because previous research indicates that these are the most reliable data sources. Moreover, these two data sources are time-synchronized into discrete frames, so they are easily combined.

The medium resolution (800x600) video streams are recorded at 30Hz. The video is discretized into a sequence of images which are then processed using a feature extraction algorithm that reduces the dimensionality of the images. The IMU device contains 5 3DM-GX1 IMUs, each with a triaxial accelerometer, gyroscopic and magnetometer sensor sampling at 125 Hz. The IMUs are placed on each of the subject's wrists, ankles, and one on the waist. We use post-processed IMU data that has been sampled down to 30Hz from 125Hz to synchronize the frames with the video data. Furthermore, in order to reduce the dimensionality of the combined video and IMU data, PCA was applied to the data, and we are able to extract features in sorted order by their eigenvalues.

An inherent challenge in working with this data is the difficulty of generalization from one subject to another. Since the subjects are given a great deal of freedom in how to execute a given recipe, the exact set of actions performed and the sequence in which they are performed vary among subjects. Moreover, there is significant variance in how a particular subject performs a certain action due to various factors (e.g., left-handed versus right-handed subjects, cracking eggs using a fork versus cracking them on the edge of a bowl).

3 Prior Work

Spriggs *et al* [3] provide baseline results for supervised frame classification, as well as unsupervised clustering. We do not address recipe classification in our analysis. Specifically, they apply Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) in an unsupervised setting and HMMs and K-Nearest Neighbors (K-NN) techniques in a supervised setting. Their experiments use data from seven subjects making brownies. We include additional data that has been collected since, yielding a total of 16 subjects in our experiments.

They find that the unsupervised methods perform quite well, in general, for recipe classification but not very well for frame classification. In the supervised frame classification setting, they find that HMMs can achieve accuracy of about 9.38% using video data only, about 10.4% using IMU data only, and about 12.34% using multi-model data. Using K-NN with K=1, they achieve accuracy of 48.64%, 56.8% and 57.8% respectively.

Video is an important part of the multi-modal dataset, and the manner in which features are extracted from image or video data has been shown to greatly influence the success or failure of many applications in computer vision[1]. The 'gist' algorithm is a context-based approach to image feature extraction that attempts to discover the salient quality of an image through the use of low-level feature channels[2]. Gist features have been used in a variety of video applications, including previous work with the CMU-MMAC data.

4 Methodology

For this project, we focus on the supervised classification of actions from a corpus of 203,581 frames taken from 16 subjects preparing the brownies recipe. Data from wired IMU's as well as gist features are given for all frames in the dataset. The frames have been manually annotated with action labels; however some of the frames have been given no label, therefore these frames have been excluded from our dataset. All algorithms were run on 32 PCA features extracted from the IMU and gist data.

We have considered many classification models in this work: K-Nearest Neighbors, Feed-forward Neural Networks, Naive Bayes, Hidden Markov Models, and Support Vector Machines with a radial basis kernel function. We also consider combinations of these constituent algorithms. The supervised learning task for our classification algorithms is to predict the correct label for a frame or for a sequence of frames taken from the data. We have taken the data from 15 subjects to use as training data, and retained data from the remaining subject for testing.

4.1 Smoothing

For the frame classification algorithm, we note that the predictions for the entire testing set do not obey the interval structure of the ground truth, in which a single action is performed continuously for some time.

Therefore, we apply a smoothing operation on the predicted labels in order to approximate this structure. The first smoothing algorithm we employed is called *fixed smoothing*, in which we segment the n frames into continuous intervals of size c, for some constant c. This gives us a set of intervals of frames, denoted $\mathbb{I}$:

$$\mathbb{I} = \{[x_1, x_c], [x_c + 1, x_{2c}]...[x_{kc} + 1, x_n]\}$$

Where x_i represents the i_{th} frame, and $k = floor\left[\frac{n}{c}\right]$. For every interval in $\mathbb{I}$, we change the label of all frames in the interval to the mode label for the interval. We have used 10-fold cross validation to discover effective values of c. This technique is intended as a baseline of comparison for similar smoothing techniques. The fixed smoothing technique is the simplest type of smoothing, and provides a useful lower bound on the effectiveness of a smoothing operation.

We have also employed a dynamic smoothing algorithm, which allows for windows of varying width. In this framework, we define a minimum and maximum size for the windows W_{min} and W_{max}. We wish to find an ideal window size for an interval beginning at frame S. For every possible window size w, we take the predicted labels $\hat{Y}$ and compute the mode label in the window and denote this mode by m_w. We then select the value of w that minimizes a variance metric, as described in the following formula:

$$W_{Ideal,S} = \max_{w \in A} \frac{1}{w^{1.15}} \sum_{i=S}^{S+w} I(\hat{Y}_i = m_w)$$
$$A = [S + W_{min}, S + W_{max}]$$

We divide the number of agreeing labels in the interval by $w^{1.15}$, a slightly super-linear quantity, in order to avoid interval lengths that are over-long. In particular, we note that the activity 'stir mix' occurs very often in our data set (20%-40% of frames represent this activity, depending on the subject). We observed that the variance of an interval would drop considerably once the window contained a sequence of the stirring action, so W_{Ideal} for an arbitrary S was often picked to be close to W_{max} in order to capture the closet stirring action. To prevent this phenomenon, we make the window sizes pay a slight cost in the metric function for becoming too large.

After the dynamic window sizes are computed, the estimate for the intervals on n test frames becomes:

$$\mathbb{I} = \{[X_1, W_{Ideal,X_1}], [X_2 = W_{Ideal,X_1} + 1, W_{Ideal,X_2}]...[X_{Ideal,k}, n]\}$$

Once again, every frame in a given interval is assigned the mode predicted label for that sequence of frames. We note that for sequence classification, the ground truth interval lengths are known ahead of time, so smoothing techniques are unnecessary.

We implemented the fixed and dynamic smoothing techniques for an SVM classifier, and the results are shown in the next section. We tried a different type of smoothing for a nearest neighbor classifier. Rather than labeling the frames individually, and then smoothing the sequence of predicted labels, we instead grouped v frames into a single input to the algorithm. In this case, the inputs are of dimension $v \cdot m$, where m is the number of features for a single frame. The algorithm is trained on the grouped frames, and an interval of frames is given the ground truth label corresponding to the mode label of the frames in the interval. We implemented this type of smoothing for $v = 10$ and $v = 50$.

4.2 K-NN/SVM and Hidden Markov Models

We employed a classification algorithm that combines K-Nearest Neighbors and Support Vector Machines, which has been shown to work well in practice for image data[4]. In the prior work by Zhang *et al*, a similarity matrix was computed for frame i with regards to its k-nearest neighbors according to a pre-defined distance metric. In our work, we employ a variation of this algorithm. For a frame i, we compute its 9-nearest neighbors according to euclidean distance, we concatenate the features of frame i with the features of its neighbors. Each frame initially begins with 32 PCA selected features, so the new feature set has dimension 320. After this new feature space is computed for all frames, we run an SVM with the radial basis kernel function on the new dataset. For frames in the testing set, we select only nearest neighbors in the training set to concatenate with the frame being input to the SVM.

For the sequence classification problem, we combined the K-NN/SVM method with a forward backward Hidden Markov Model. In the sequence classification problem, we are given a set of starting times of activities

$\{A_1, A_2 ... A_k\}$, such that only a single ground truth activity occurs in an interval $[A_i, A_{i+1} - 1]$. We then label every one of the n frames individually using the K-NN/SVM algorithm. Using the training data, we empirically estimate the probability of activity i being followed by activity j, for all pairs (i, j). We also estimate the probability of activity i occurring first and last in the sequence.

We now compute the alpha and beta parameters for the forward-backward HMM. We denote the label of interval $[A_i, A_{i+1} - 1]$ as Y_i, and X_i represents the features of the frames in the interval:

$$\alpha_1(Y_1) = P(Y_1 \text{ is first label})P(X_1|Y_1)$$

$$\alpha_i(Y_i) = \sum_{y_{i-1}} P(X_i|Y_i)P(Y_i|Y_{i-1} = y_{i-1})\alpha_{i-1}$$

$$\beta_n(Y_n) = P(Y_n \text{ is last label})P(X_n|Y_n)$$

$$\beta_i(Y_i) = \sum_{y_{i+1}} P(X_i|Y_i)P(Y_i|Y_{i+1} = y_{i+1})\beta_{i+1}$$

The factor $P(X_i|Y_i)$ represents the probability of this data given the label. In our case, this is the probability of all the features of the frames in the interval given the label. This quantity is normally estimated empirically, but we instead use the output of our K-NN/SVM algorithm to estimate this quantity. We simply use the proportion of labels in A_i with label Y_i to create our estimate. Specifically, if $\hat{y}_j$ represents the predicted label of frame j given by K-NN/SVM, we have the following approximation:

$$P(X_i|Y_i) = \frac{1}{|[A_i, A_{i+1} - 1]|} \sum_{j=A_i}^{A_{i+1}-1} I(\hat{y}_j = Y_i)$$

After these parameters have been computed, we assign every frame in interval $[A_i, A_{i+1} - 1]$ the following label:

$$\max_{Y_i} \alpha_i(Y_i)\beta_i(Y_i)$$

We applied this HMM framework to the sequence classification problem, but it is less clear how to apply it to the frame classification problem. In particular, if we compute the probability of label i being followed by label j for every frame in the set, we find that this probability is approximately 99.6% for $i = j$. On a frame by frame basis, two adjacent frames share a label nearly all of the time (only around 35/10,000 frames in a testing set will have a different label than the frame that preceded it). The fact that a frame is usually followed by another frame of the same label is a phenomenon that we tried to capture with the smoothing techniques, but making this behavior explicit using the forward-backward algorithm generally results in every testing frame being given the same label. As such, a slightly different approach would be required to apply an HMM to the frame classification problem.

5 Results

Table 1: Result synopsis

Technique	**Problem setting**	**Frame classification accuracy**
Neural Nets	Frame classification	4.32%
Naive Bayes	Frame classification	12.6%
K-NN (K=1)	Frame classification	23.4%
SVM (Unsmoothed)	Frame classification	46.4%
SVM (Fixed smoothing)	Frame classification	50.4%
SVM	Sequence classification	61.4%
K-NN/SVM (Dynamic smoothing)	Frame classification	64.0% (Average over 2 subjects)
K-NN/SVM (Fixed smoothing)	Frame classification	64.1% (Average over 2 subjects)
K-NN/SVM/HMM	Sequence classification	74.8% (Average over 2 subjects)

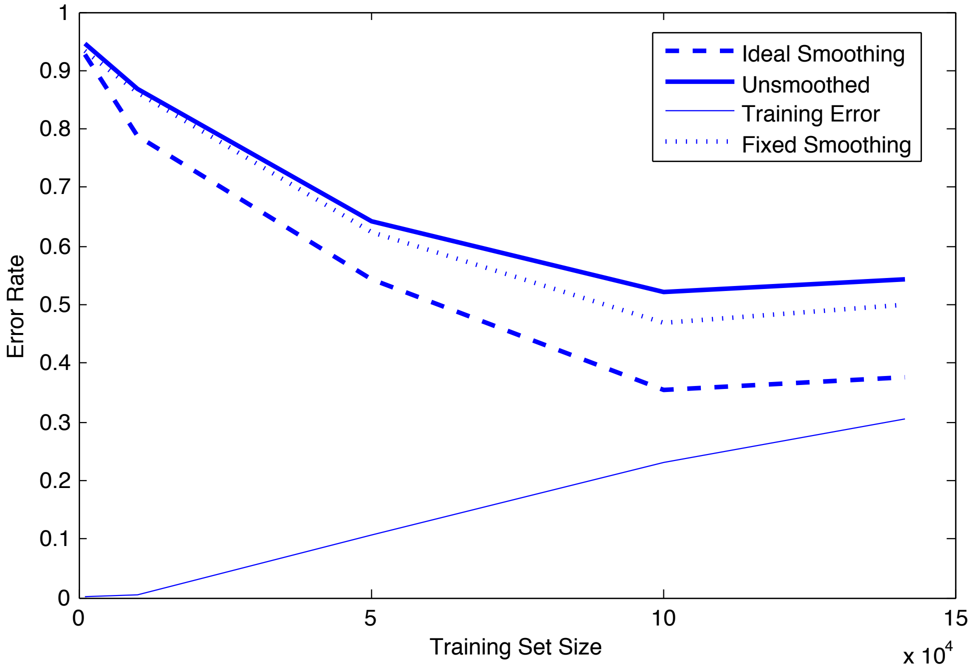

Figure 1: SVM performance

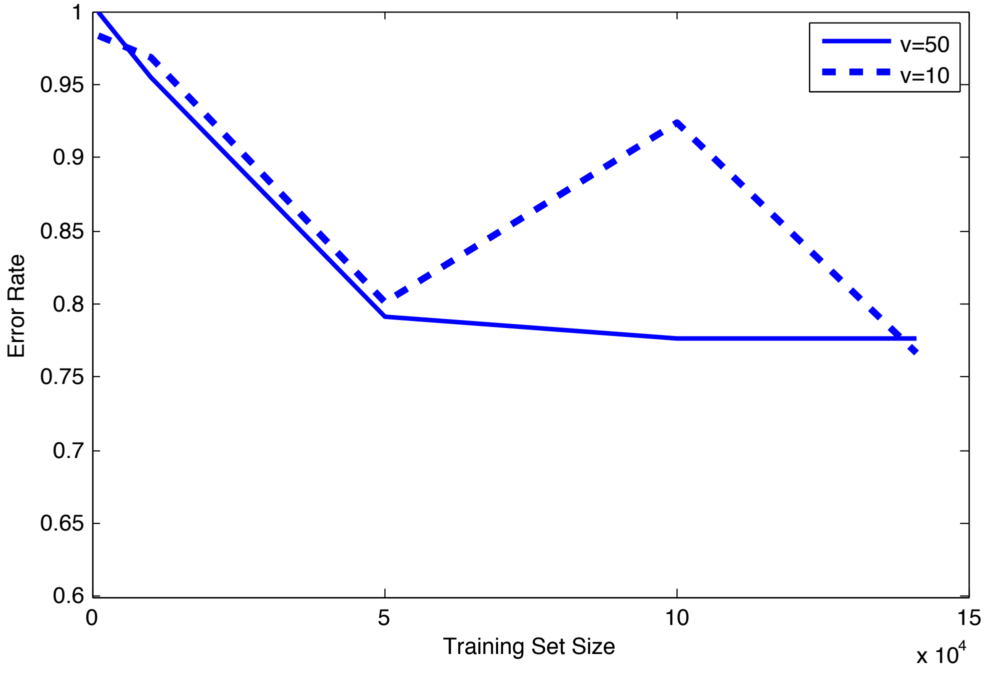

Figure 2: Nearest neighbor performance

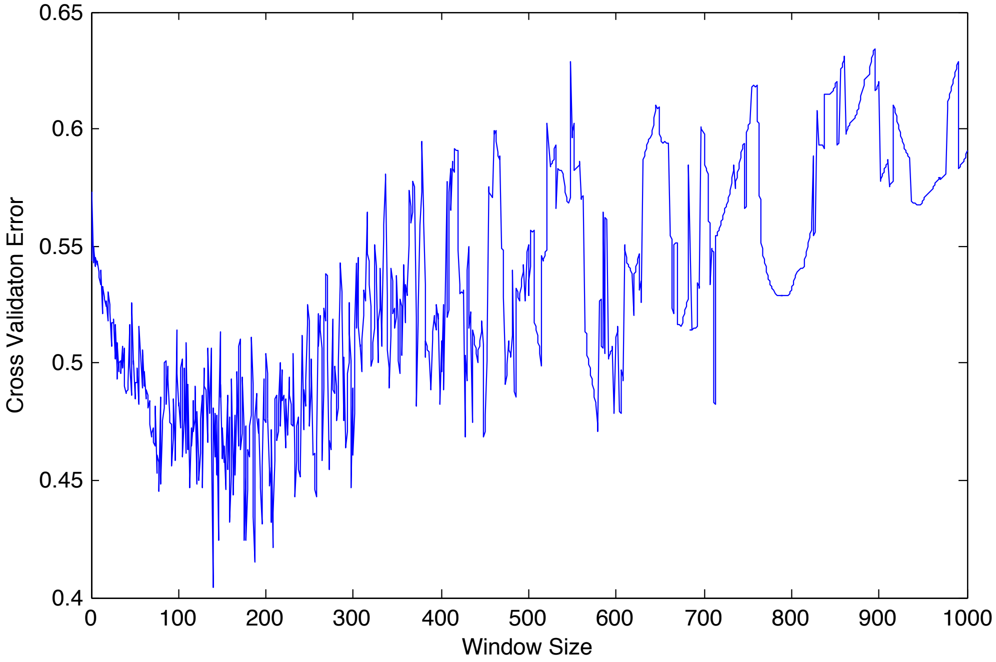

Figure 3: SVM cross validation of window size

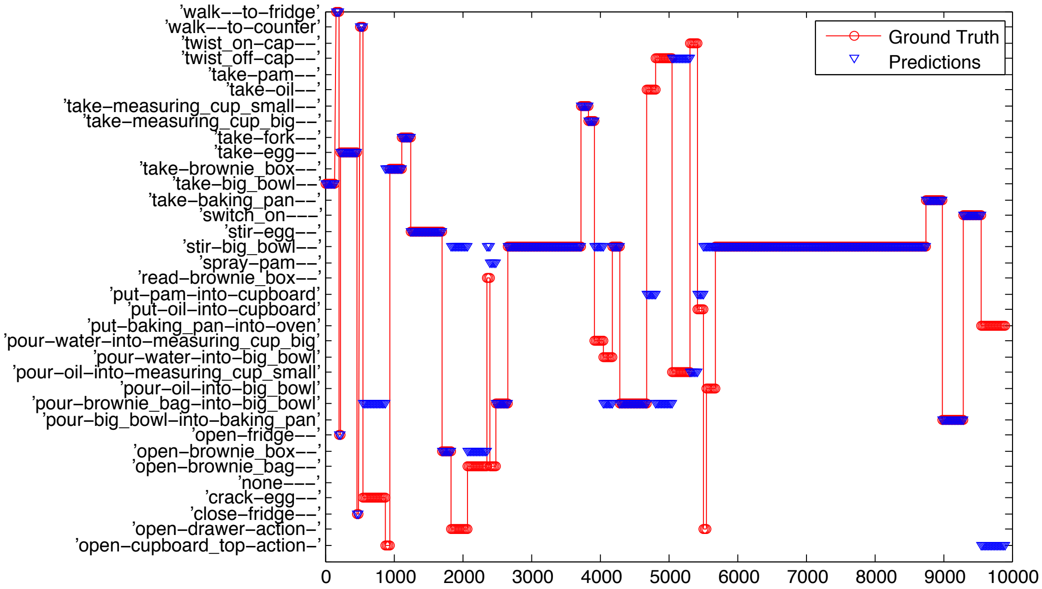

Figure 4: K-NN/SVM/HMM sequence prediction on subject 16
Frame classification accuracy: 73.2%

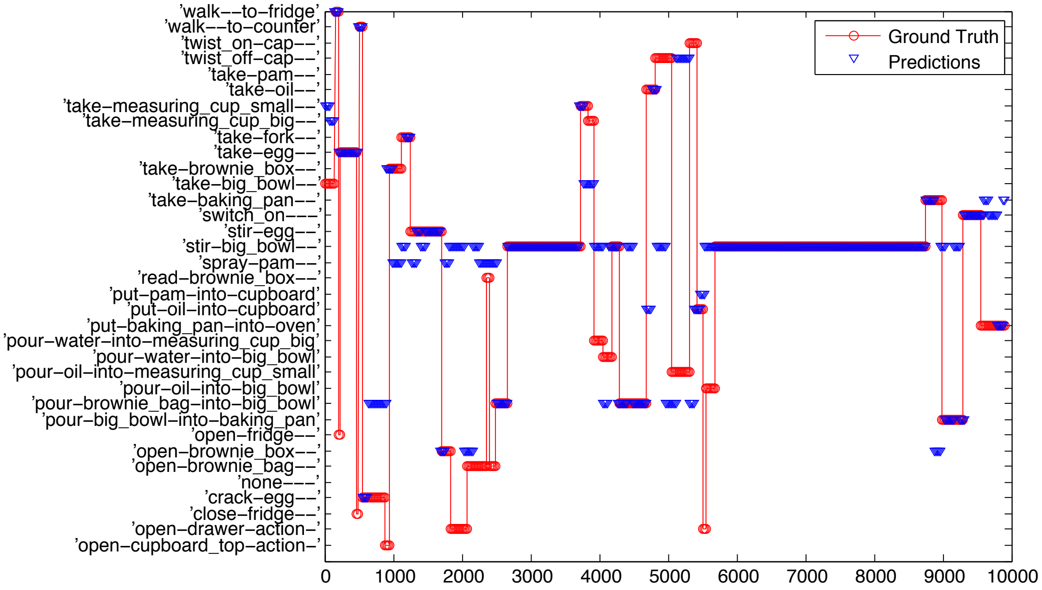

Figure 5: K-NN/SVM frame classification with fixed smoothing on subject 16
Frame classification accuracy: 63.4%

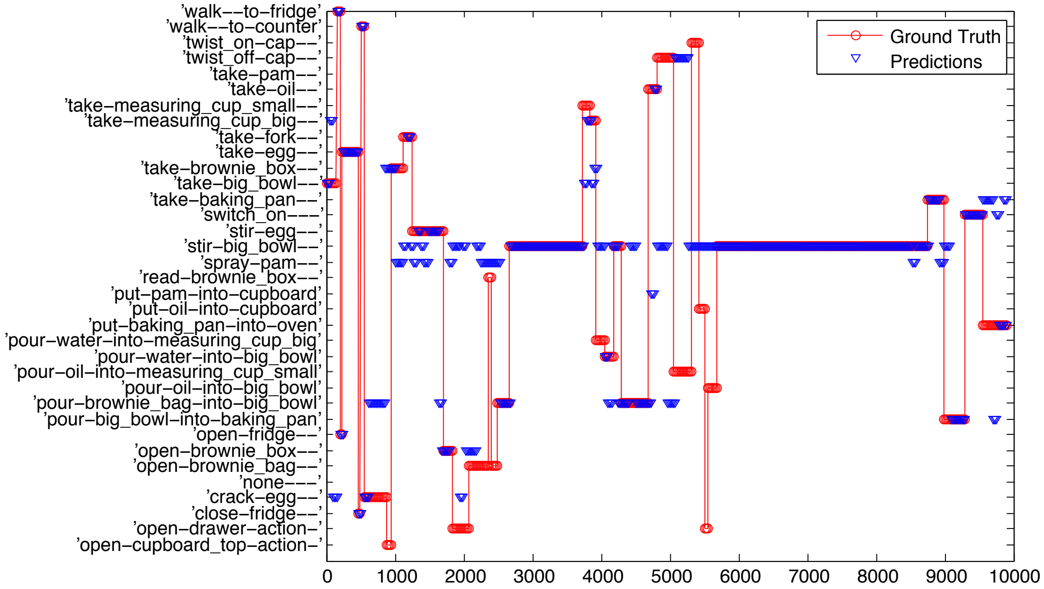

Figure 6: K-NN/SVM frame classification with dynamic smoothing on subject 16
Frame classification accuracy: 62.9%

6 Analysis

6.1 Separate Classfiers

The results from the SVM and K-NN models are shown above in figures 1 and 2. In addition, we employed Naive Bayes and Feed-forward Neural Networks to the dataset, but both of these models failed to achieve the best prior accuracy[2]. Naive Bayes trained on 15 subjects achieved a maximum testing accuracy of 12.6%, while the feedforward nets attained 4.32% accuracy. For the SVM classifier shown above, we used cross validation to select a window size of 70 for the fixed smoothing procedure (although the cross-validation

[2] The best prior accuracy is realized by the "stir mix" label, which occurs in 25% of all frames.

curve above shows that larger values may also be effective) and a slack variable value of 10. We used $K = 1$ for the K-NN algorithm.

We also tried partitioning the full set of frames into a randomly selected testing and training sets. We observed that under these conditions the SVM algorithm achieved 90-100% testing accuracy. We believe this is because of the remarkable similarity of consecutive frames in the dataset, so these results do not represent a realistic scenario for classification testing.

We observed expected results for the SVM testing, but the K-NN testing error was lower than anticipated based on prior work[3]. We believe that the corpus of frames may have been too large for K-NN to be useful with $K = 1$, even with the smoothing procedure. We also noted a significant drop in accuracy when K-NN with $v = 10$ was applied to a training set of 100,000 frames. Our testing was conducted on a single subject, and it is possible that this subject's data was not amiable to the nearest neighbor algorithm.

6.2 Combined Classifiers

As expected, the classifier for the sequence classification problem outperformed the frame classification algorithm (figure 4). Without the Hidden Markov Model, the sequence classification algorithm produced 69.5% accuracy, indicating that the HMM framework provides some benefit. However, using the HMM alone for sequence classification (using empirical estimates for $P(X_i|Y_i)$, rather than the K-NN/SVM output) yields only 18.6% accuracy, so the forward-backward algorithm was only seen to be effective when combined with another classifier. The K-NN/SVM/HMM algorithm was also tested on subject 3 with frame classification accuracy **76.3%**, giving an average accuracy over both subjects of 74.8%.

Using K-NN/SVM on the frame classification problem (figures 5 and 6) yielded slightly higher accuracy than SVM alone. What was slightly surprising was that the dynamic smoothing algorithm performed slightly worse than the fixed smoothing algorithm. The dynamic smoothing algorithm was run with $W_{min} = 40$ and $W_{max} = 5,000$. The largest window selected was 1,361 frames, and the average window size was 60.5. It is worth noting that the average dynamic window size was very near the static window size of 75, and we see that the final predictions are very similar. As such, we conjecture that the fixed smoothing algorithm, which also has lower computational complexity, is the preferred method. However, a different metric function for the dynamic smoothing algorithm may produce results more comparable to the sequence classification accuracy. In particular, if a classifier is used that provides confidence measures of a labeled examples (such as the distance from a separating hyperplane in an SVM), these measures could be used directly in the smoothing metric.

The K-NN/SVM frame classifier was also run on subject 3, with fixed smoothing accuracy **64.8%** and dynamic smoothing accuracy **65.1%**. In this case, the dynamic smoothing algorithm performed slightly better, but as with subject 16, the set of predicted labels for the two smoothing algorithms were very similar.

7 Conclusion

Our work indicates that smoothing techniques and consideration of the probable ordering of activities can be used to increase the effectiveness of algorithms for both classification problems. Although our work has focused on a somewhat myopic feature extraction scheme oriented around individual frames, we feel that these techniques could also be successfully applied to features extracted from windows of frames—which may eventually lead to more effective activity recognition algorithms.

References

[1] Due, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641–662, April 1996.

[2] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[3] E. H. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. *Computer Vision and Pattern Recognition Workshop*, 0:17–24, 2009.

[4] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. volume 2, pages 2126–2136, 2006.