Improving the Supervised Learning of Activity Classifiers for Human Motion Data

Liyue Zhao, Xi Wang, and Gita Sukthankar
Department of EECS
University of Central Florida

Revised Mar 31, 2011

The ability to accurately recognize human activities from motion data is an important stepping stone toward creating many types of intelligent user interfaces. Many supervised learning methods have been demonstrated for learning activity classifiers from data; however, these classifiers often fail due to noisy sensor data, lack of labeled training samples for rare actions and large individual differences in activity execution. In this chapter, we introduce two techniques for improving supervised learning of human activities from motion data: 1) an active learning framework to reduce the number of samples required to segment motion traces and 2) an intelligent feature selection technique that both improves classification performance and reduces training time. We demonstrate how these techniques can be used to improve the classification of human household activities, an area of particular research interest since it facilitates the development of elder-care assistance systems to monitor household occupants.

1 Introduction

Human activity recognition has become an increasingly important component of many domains such as user interfaces and video surveillance. In particular, enabling ubiquitous home user assistance systems for elder care requires rapid and robust recognition of human action from portable sensor data. Motion trajectories, gathered from video, inertial measurement units, or mocap, are a critical cue for identifying activities that require gross body movement, such as walking, running, falling, or waving. Human motion data typically needs to be segmented into activities to be utilized by any application. A common processing pipeline for motion data is:

1. segment data into short time windows;
2. recognize low-level human activities from repetitive patterns of motion executed by the human user within a time window;
3. identify a high-level intention or plan from sequences of activities.

For instance, one possible high-level household activity would be "baking pizza" which would consist of low-level activities such as "beating an egg" or "kneading dough" which could be recognized by the motion patterns and objects manipulated.

Although domain knowledge and common-sense reasoning methods are important for reasoning about the human's high level intentions, segmentation and activity classification have been successfully addressed by a variety of data-driven approaches, including supervised classifiers, such as support vector machines, hidden Markov models, dynamic Bayes nets, and conditional random fields. In the best case, supervised learning can yield classifiers that are robust and accurate. However, two problems frequently occur in supervised learning settings.

lack of data: gathering and labeling the data is time-consuming and expensive. In some cases, the activities are highly repetitive in nature (stirring), whereas other actions are infrequent and short in duration (opening the refrigerator). To classify these short actions, learning techniques need to be sample-efficient to leverage relatively small amounts of labeled training data.

feature selection: sensors yield data that is both noisy and high-dimensional. Learning classifiers based on the raw sensor data can be problematic and applying arbitrary dimensionality reduction techniques does not always yield good results.

In this chapter, we present a case study of how we addressed these problems while performing segmentation and activity recognition of human household actions. First, we introduce an active learning method in which the classifier is initialized with training data from unsupervised segmentation and improved by soliciting unlabeled samples that lie closest to the classification hyperplane. We demonstrate that this method can be used to reduce the number of samples required to classify motion capture data using support vector machine (SVM) classifiers.

Second, we present a method to improve classification through intelligent feature selection. The signal data is converted into a set of motifs, approximately repeated symbolic subsequences, for each dimension of IMU data. These motifs leverage structure in the data and serve as the basis to generate a large candidate set of features from the multi-dimensional raw data. By measuring reductions in the conditional log-likelihood error of the training samples, we can select features and train a conditional random field (CRF) classifier to recognize human actions.

Our techniques were evaluated using the CMU Multimodal Activity database (De la Torre et al., 2008) which was collected to facilitate the comparison of different activity recognition techniques for recognizing household activities. The dataset contains video, audio, inertial measurement unit (IMU), and motion capture data; we demonstrate the utility of our techniques on segmenting motion capture data and recognizing inertial measurement unit (IMU) data.

2 Background

In this section, we give an overview of the concepts that our approach relies on 1) active learning 2) feature selection and 3) motif detection, in addition to a detailed discussion of the operation of the conditional random field classifier.

2.1 Active Learning

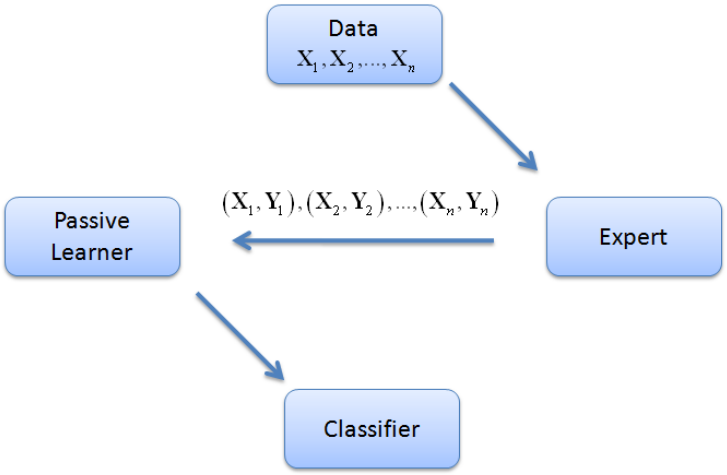
Active learning can be regarded as a subfield of supervised learning in which the aim is to achieve the same classification with a smaller set of labeled data. As shown in Figure 1, standard (passive) learning techniques require humans to annotate the entire set of input training data that the learner uses to build the classifier. By contrast, in active learning, the annotation of data is performed through a series of interactions between the automated learner and the human. By analyzing the unlabeled data, the active learner selects a subset of the data that has a high degree of label uncertainty to be annotated by the expert. Currently, two main issues of active learning research are how to pose queries and identify informative instances.

There are three general methods for posing queries: *membership queries*, *selective sampling* and *pool-based sampling*. Membership queries algorithms (Angluin, 1988, 2004) generate new instances in the input space and request these labels from the expert. However, such algorithms may construct unexpected instances which are difficult for human experts to label (Baum and Lang, 1992). The selective sampling algorithm was introduced to overcome this problem (Atlas et al., 1990; Cohn, 1996). In selective sampling, the learner receives distribution information from the environment and queries the expert on parts of the domain. The learner sequentially draws individual instances and decides whether or not to request its label by evaluating the *region of uncertainty*. The *region of uncertainty* is reduced after each new instance is added; more instances are included if the uncertainty is not reduced efficiently.

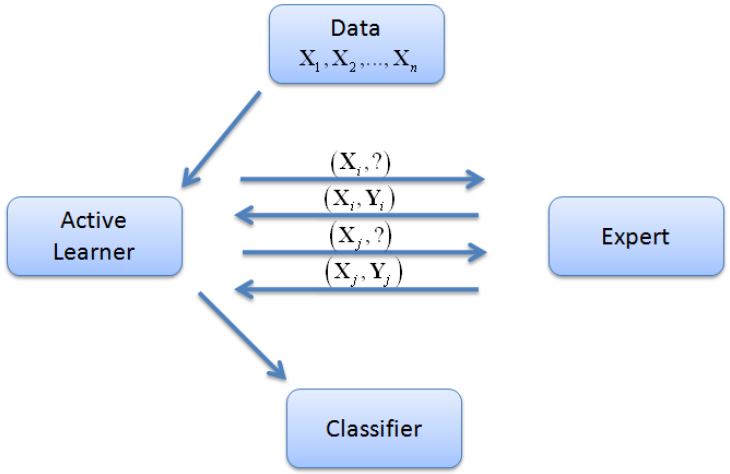
Currently, the most popular query strategy in active learning is pool-based sampling, in which samples are requested from an existing closed set of instances. These algorithms share the common assumption that there is a small set of labeled data and a large pool of unlabeled data. The learner generated from the labeled set is applied to evaluate the informativeness measure of instances in the unlabeled pool. The label of the most informative instance is requested, and the instance is then added to the labeled set.

One question is how best to identify the most informative instances for future label requests. Although there are different ways to evaluate the informativeness of unlabeled instances, the main approaches are *query by uncertainty* and *query by committee*. The query by uncertainty algorithm starts by building the learner using the labeled data set. The learner is used to provide a confidence score for each unlabeled instance to probe for instances where the learner is the least certain of the currently classified labels. Those uncertain labels are requested from the experts, and the instances become the members of labeled data. This process repeats until the learner is confident about all of the unlabeled data. Query by uncertainty is probably the most straightforward approach and has been demonstrated in several real-world applications (Tong and Chang, 2001; Arikan et al., 2003; Chang et al., 2005).

An alternative query strategy is the query by committee algorithm. The essential idea behind this approach is to narrow the possible hypotheses in the *version space* (the set of classifiers consistent with the training set) (Mitchell, 1982). A committee of classifiers is generated from labeled data to evaluate the unlabeled data. The instance with the most classifier disagreement is deemed to be the most informative instance. As shown in Figure 2, by querying new labels, the version space narrows to reduce the



(a) Passive learning



(b) Active learning

Figure 1: In the standard passive learning approaches (as shown in a) the expert needs to annotate all data to feed the passive learner. In active learning (shown in b), the active learner identifies a subset of the most informative data for the expert to annotate.

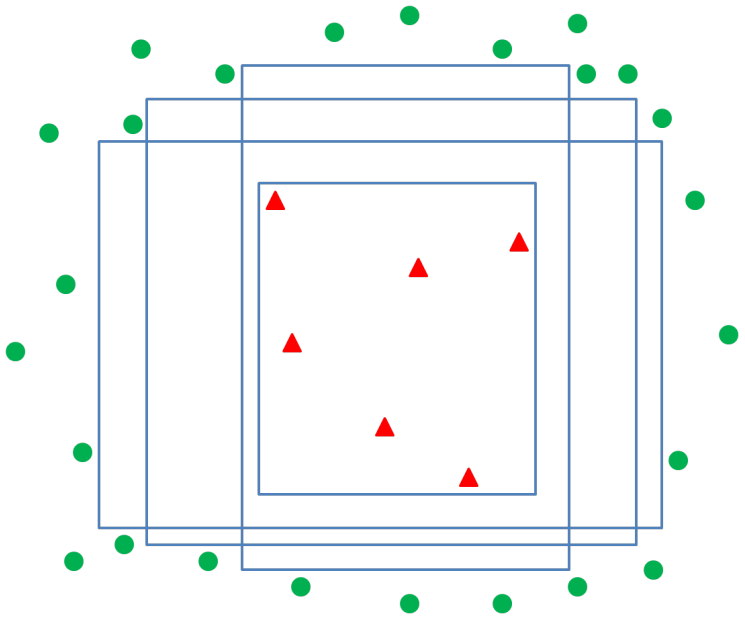


Figure 2: Version space for rectangle hypotheses in two dimensions. Assume the red triangles are positive samples and the green circles are negative samples. All blue rectangles agree with those training samples and are possible hypotheses in the version space.

number of possible hypotheses. Therefore, the optimal learner will be reached once there are no disagreements among hypotheses learned with all the instances.

Active learning has been successfully applied to many classification problems. Tong demonstrated the use of Support Vector Machines (SVMs) to construct pool-based active learners for both text classification (Tong and Koller, 2002) and image retrieval (Tong and Chang, 2001). In the binary classification problem, it is assumed that the most informative instances should split the version space into two equal parts; this formulation can also be extended to multi-class classification. Several new strategies have been proposed for evaluating the informativeness of unlabeled data (e.g., Chang et al. (2005)). Also, Wang et al. (2003) proposed a new bootstrapping strategy for SVM active learning.

However, many research issues remain. First, in most applications, the evaluations of the unlabeled data are greedy and myopic. Negligence of the global distribution of data may cause the "best" identified learner to converge to a locally optimal hypotheses. A second problem is that the uncertainty sampling is inherently "noise-seeking" and may take a long time to converge. Hierarchical clustering approaches such as Dasgupta and Hsu (2008) have been devised to ameliorate some of these problems. The unlabeled data is used to perform a hierarchical clustering in which leaves on the tree are pruned or grown, depending on whether the leaf is pure or mixed. This clustering approach provides the learner with more global information about the data and avoids

the tendency of the learner to converge to local maxima.

2.2 Motion Capture Segmentation

In our work, we apply active learning toward the problem of segmenting motion capture data. Here we discuss other strategies that have been applied to that problem. Barbic et al. (2004) introduced several approaches to motion capture segmentation based on the general concept that there is an underlying generative model of motion and that cuts should be introduced at points where the new data diverges from the previous model. In one of their proposed methods, principal component analysis (PCA) is used to create a lower-dimensional representation of the motion capture data at the beginning of a motion sequence. The main insight is that if the observed motion diverges from the data used to create the PCA basis, such as when the actor starts to perform a new action, then projecting the data of the new action using the old model will lead to large reconstruction errors. The moment that reconstruction errors increase quickly will occur at or near action boundaries.

However, in practice this approach leads to several problems. The method relies on building the PCA basis with frames from the current action, which requires about 300 frames or 2.5 secs of data. Unfortunately in our dataset, action changes can occur within that time frame, yielding a mixed basis capable of representing both actions without large reconstruction errors. Hence this technique cannot be used to accurately segment datasets with many short duration actions. Additionally, since PCA is a completely unsupervised approach, it is unable to distinguish between an activity that consists of multiple actions and boundaries between two semantically unrelated activities.

In cases where user labels can be easily obtained, segmentation can be done in a completely supervised fashion using interactive SVMs to label the data (Arikan et al., 2003). Initially, users label a small training set of data. Then with kernel function Φ, the SVM classifier maps the training data into a high dimensional space which makes the data linearly separable. Since the partition hyperplane may not fit the unlabeled data, the user can add new labels to the training set and retrain the classifier. The method strives to balance classification accuracy and the user's labeling workload. However, their selection of new samples are based on the empirical judgment of the user and therefore susceptible to human error.

Our approach draws from both these methods, using an unsupervised PCA segmentation to initialize the clustering and a semi-supervised method to train the SVM classifiers. Unlike the interactive SVM segmentation proposed by (Arikan et al., 2003), our approach utilizes the unlabeled data sets in the initial training. In the second phase, we automatically determine which instances from the unlabeled data are most useful to solicit labels from the user in the next iteration. Thus, the user is freed from selecting unlabeled samples and merely needs to label a small number of informative instances; this eliminates human bias and aims to reduce the amount of data that requires manual attention.

2.3 Conditional Random Fields

A number of supervised learning frameworks have been used to recognize low-level human activities from repetitive patterns of motion. In addition to support vector machines, we have experimented with the use of conditional random fields (CRF). Conditional random fields (CRFs) are undirected graphical models $G = (V, E)$ that represent the conditional probabilities of a label sequence $\mathbf{y} = \{y_1, ..., y_T\}$, when given the observation sequence $\mathbf{x} = \{x_1, ..., x_T\}$. When conditioned on $\mathbf{x}$, the label y_i holds the Markov property

$$p(y_i|\mathbf{y}_{G\setminus i}, \mathbf{x}) = p(y_i|\mathbf{y}_{\mathcal{N}_i}, \mathbf{x}) \tag{1}$$

where $\mathbf{y}_{\mathcal{N}_i}$ represents all the neighborhoods that connect to y_i. The Hammersley-Clifford Theorem tells us that this equation is equivalent to $p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C(\mathbf{y},\mathbf{x})} \psi_c(\mathbf{y}_c, \mathbf{x}_c)$ if the graphical models G obey the Markov assumption, where $\psi_c(\mathbf{y}_c, \mathbf{x}_c)$ is the non-negative potential functions of clique c and $Z(\mathbf{x})$ is a normalization constant. For our problem, the labels correspond to activities, such as "chop vegetables", while the observations consist of a set of features computed over the raw data.

Linear chain CRFs are graphical models defined with a linear chain structure such that the current state y_i only relates to the previous state y_{i-1} and the observation $\mathbf{x}_c$. Linear chain CRFs require no assumptions of independence among observations, thus $\mathbf{x}_c$ can be any part of the observation sequence $\mathbf{x}$ and the conditional probability $p(\mathbf{y}|\mathbf{x})$ can be written as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^{T} \psi_i(y_{i-1}, y_i, \mathbf{x}, i) \tag{2}$$

where $Z(\mathbf{x}) = \sum_y \prod_{i=1}^{T} \psi_i(y_{i-1}, y_i, \mathbf{x}, i)$ and $\mathbf{x}$ represents observations over the whole sequence.

Since CRFs are log-linear models, the potential function can be written as the linear sum of feature functions as $\psi_i(y_{i-1}, y_i, \mathbf{x}, i) = \exp(\sum_j w_j f_j(y_{i-1}, y_i, \mathbf{x}, i))$, where w_j represents the weight corresponding feature $f_j(y_{i-1}, y_i, \mathbf{x}, i)$. The conditional probability can then be calculated as

$$p(\mathbf{y}|\mathbf{x}; \mathbf{w}) = \frac{1}{Z(\mathbf{x})} \exp(\sum_{i=1}^{T} \sum_j w_j f_j(y_{i-1}, y_i, \mathbf{x}, i)) \tag{3}$$

where $Z(\mathbf{x}) = \sum_y \exp(\sum_{i=1}^{T} \sum_j w_j f_j(y_{i-1}, y_i, \mathbf{x}, i))$.

Assuming a fully labeled dataset with pairs of training samples $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$ the CRF parameter vector $\mathbf{w} = \{w_1, ..., w_M\}$ can be obtained by optimizing the sum of conditional log-likelihood $L(\mathbf{Y}|\mathbf{X}; \mathbf{w})$.

$$L(\mathbf{Y}|\mathbf{X}; \mathbf{w}) = \sum_k \log p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}; \mathbf{w}) \tag{4}$$

$$= \sum_k (\sum_{i=1}^{T} \sum_{f_j \in \mathcal{S}} w_j f_j(y_{i-1}^{(k)}, y_i^{(k)}, \mathbf{x}^{(k)}, i) - \log Z(\mathbf{x}^{(k)})) \tag{5}$$

The first derivative of the log-likelihood is:

$$\frac{\partial L(\mathbf{Y}|\mathbf{X};\mathbf{w})}{\partial w_j} = \sum_k (\sum_{i=1}^{T} \sum_{f_j \in \mathcal{S}} f_j(y_{i-1}^{(k)}, y_i^{(k)}, \mathbf{x}^{(k)}, i) \tag{6}$$

$$- \sum_y \sum_{i=1}^{T} \sum_{f_j \in \mathcal{S}} p(\mathbf{y}|\mathbf{x}^{(k)};\mathbf{w}) f_j(y_{i-1}, y_i, \mathbf{x}^{(k)}, i)) \tag{7}$$

Since the CRFs log likelihood is convex with respect to the weight vector w, standard optimization methods such as conjugate gradient and limited memory BFGS (Liu and Nocedal, 1989) can be used to discover the weights. In this work, we learn both the set of features $f_j(.)$ and their associated weight parameters w_j, for multi-class classification.

2.3.1 Applications of CRFs

CRFs have been used for a variety of classification problems, including natural language processing (Lafferty et al., 2001; Culotta and McCallum, 2004), computer vision (Kumar and Hebert, 2003; Tappen et al., 2007; Levin and Weiss, 2009; Plath et al., 2009), human activity recognition (Sminchisescu et al., 2005; Liao et al., 2007; Vail et al., 2007) and bioinformatics (Fu et al., 2009).

However, effective learning and inference of CRFs remain challenging problems. If the optimization cost function is nonlinear and nonconvex, learning and inference of CRFs is often implemented with sampling-based algorithms that require a long time to converge. However, recent work by Levin and Weiss (2009) showed the use of first order approximation to efficiently estimate conditional likelihood. For Gaussian CRFs, Tappen et al. (2007) demonstrated it is possible to perform efficient parameter optimization by minimizing the error in the model MAP solution.

More specifically, linear CRFs have been widely applied to classification and segmentation of sequential data. Lafferty et al. (2001) demonstrate the use of linear CRFs to solve the label bias problem in natural language processing. The main superiority of linear CRFs is their ability to effectively take advantage of overlapping, non-independent features. For instance, Fu et al. (2009) empriically evaluated the predictive power of using different feature sets; their experiments indicated that their linear CRF could effectively leverage discriminative features found in alternate feature sets unlike their earlier HMM-based classification system (Lin et al., 2008). Similarly, Sminchisescu et al. (2005) found that their CRF-based classifier surpassed an HMM model at an activity recognition task using images and motion capture data sequences with overlapping observation features. In summary, since the linear CRF model permits the utilization of non-independent features, feature generation and selection can dramatically effect classification performance.

2.4 Feature Selection

There are three general strategies used in feature selection: *filters*, *wrappers* and *embedded* (Vail, 2008). *Filters* treat feature selection as the pre-processing step, independent from the training stage. *Wrappers* consider all possible combinations of potential features to select the best feature set according to the learning performance. *Embedded* strategies perform feature selection during the training stage and iteratively grow the feature set.

Filters apply heuristic methods, such as correlation, mutual information and information gain (Guyon and Elisseeff, 2003), to select features prior to training. Those methods rank all potential features by evaluating their relevance; features with high relevance scores are used as the feature set for training. It is worth noting that that such methods evaluate potential features only once; no features can be added or eliminated during training, which makes the filter strategy computationally efficient. However, the tradeoff is that the filters may select irrelevant features while ignoring highly relevant features if the heuristics do not perform well in a particular domain.

Wrappers perform an exhaustive search on all possible combination sets of potential features. Every set of features is used to train the model and evaluated with cross-validation (Bishop, 2006). However, the number of combination of feature sets increases exponentially with the number of potential features. Although this strategy avoids the *the weakly relevant feature* problem in filters, obviously it is computational inefficient if the feature pool is large.

The *embedded* method can be viewed as a tradeoff between filters and wrappers. This method generates a small set of features for training while leaving other features as candidates for selection. Candidate features are added to the set evaluated by the learning model. The feature set is iteratively grown through adding features with high evaluation scores. This method is more computationally efficient than wrappers and does not rely on potentially fallible heuristics. Currently, many real-world applications are based on embedded methods.

2.5 Feature Selection for CRFs

Prior work (Sminchisescu et al., 2005) has examined the relative benefits of using discriminative conditional random fields (CRFs) vs. commonly-used generative models (e.g., the Hidden Markov Models) on diverse activity recognition problems such as video recognition and analyzing Robocup team behavior. The consensus has been that many of features commonly used in activity recognition problems are based on overlapping time windows that nullify the observational independence assumptions of many generative graphical models. However, feature representation and selection does have an impact on both the performance and computational accuracy of CRFs.

Due its resiliency to classification problems arising from non-independent features, CRFs can often leverage arbitrary and complex features derived from some basic features. Unfortunately, the strategy of adding as many features as possible to guarantee no loss of information creates certain problems. First, large feature sets can cause overfitting. Since each feature corresponds to one parameter in the CRF model, a large feature set implicitly means that there is a large set of parameters which can fit the

training data perfectly while performing poorly in the test set. Convergence during learning becomes harder with large numbers of weights. McCallum (2003) proposed a feature induction method for linear-chain CRFs in which features with the highest conditional likelihood gain are iteratively selected. Initially, the model generates a pool of feature candidates. Each feature candidate is then added to the current feature set to evaluate its contribution to the conditional likelihood of current model. Those features with highest gain are selected as new training features and the CRFs model is re-trained. Vail et. al. Vail et al. (2007, 2008) applied a similar feature selection framework for robotic activity recognition. Vail's method generates a large pool of complex, overlapping features of robot activities, such as positions, velocities and distances. l_1 and l_2 regularization are applied to reduce feature quantity and speed up the selection process. However, the pool-based feature selection strategy is time consuming, since the conditional likelihood has to be re-calculated with every potential feature at each step, which is not practical if there are a large number of candidate features. Our proposed feature selection method draws from these ideas, but relies on an approximation technique to reduce the computation time.

2.6 Motif Discovery

Feature selection enables the selection of the most discriminative features from an initial set of candidate features. For noisy sensor data from inertial measurement units, an open question is which features should be used in the initial candidate set. Unsupervised motif discovery has been demonstrated as a promising technique for identifying and matching imperfect repetitions in time series data; by using motifs as the basis for our CRF features, we can robustly identify subtle patterns of peaks and valleys in the IMU data.

The problem of repeated subsequences in time series spans multiple research areas. Researchers have primarily addressed two problems: 1) initial motif selection criteria (Tanaka et al., 2005) and 2) rapid and robust matching (Chiu et al., 2003). Lin et al. (2002) initially presented a formal definition of *motifs*, the approximately repeated subsequences in time series. Their approach for detecting motifs in time series suffers from certain limitations. A brute force algorithm requires a quadratic number of comparisons corresponding to the length of the time series. Lin et al. (2002) proposed a triangular inequality approach to improve the time complexity of the comparisons; however their method requires the manually setting of large constant factors which makes it hard to extend to massive databases. Also this approach is sensitive to noise, as demonstrated by the example shown by Chiu et al. (2003) in which the matching of two subsequences is corrupted if the subsequence has a noisy downward spike. To solve those two issues, Chiu et al. (2003) propose a novel motif discovery algorithm to efficiently discover motifs in which the time series is discretized as a symbolic sequences using Piecewise Aggregate Approximation (PAA) (Keogh et al., 2001). By switching to a symbolic representation, they achieve some noise reduction. In order to speed up the computation of motif matching, they apply random projection to approximate the comparison of every potential subsequence. Keogh (2002) extend this method to finding arbitrarily scaled motifs in a massive time series database.

Another issue is how to generalize single dimensional motif techniques to the prob-

lem of detecting and matching motifs in multi-dimensional data (Vahdatpour et al., 2009). Tanaka et al. (2005) propose the use of a Minimum Description Length (MDL) principle to extract motifs from time series data. Principal Component Analysis (PCA) is then applied to synchronize motifs in multidimensional time series data. Vahdatpour et al. (2009) demonstrated a graph clustering approach for grouping single dimensional motifs to solve the synchrony issue. In our work, we identify and match motifs in each dimension separately and use the conditional random field to learn the linkage between motif occurrences across different dimensions and action class labels.

Motif discovery is a powerful tool for analyzing time series data delivered by wearable sensors and has been employed in wearable systems such as SmartCane (Wu et al., 2008) and SmartShoe (Dabiri et al., 2008) to identify activities based on accelerometers, gyros and press sensor data. String-matched activity templates have been used to recognize continuous activities in a car assembly scenario (Stiefmeier et al., 2007, 2008).

Minnen et al. (2006) used motifs in conjunction with HMMs to distinguish six arm techniques from a single IMU, but the assumption was made that each action could be characterized by a single six-dimensional motif. Their work synthesizes several existing approaches to estimate the hidden location probability and motif model parameters. First, the algorithm generates a pool of candidate motifs and selects the best motifs based on their scores of informative-theoretic criterion. Then the seed motifs are refined by splitting different motifs or merging similar motifs. Finally a Hidden Markov Model (Rabiner, 1989) is built with those seed motifs and corresponding occurrences could be detected by decoding the HMM. In contrast, our data is substantially more complicated (full body data generated from from five IMU sensors) and our action set is composed of 14 unscripted actions performed during a cooking task (e.g., "stir brownie mix", "pour mix in pan", "get fork", "break eggs", and "pour oil").

3 Main Thrust

In this section, we describe the techniques we employ to make learning sample-efficient and to select the best set of features for a given motion dataset.

3.1 Sample Efficiency through Active Learning

In the initial phase, our SVM classifiers are initialized with a small set of training data from the unsupervised clustering. During active learning, the classifiers are iteratively trained by having the users provide labels for a small set of automatically selected samples. Although the classifiers can be initialized by having the user provide labels for randomly sampled frames, we demonstrate that we can improve on that by selectively querying and propagating labels using a clustering approach.

3.1.1 Data Clustering

Several methods have been proposed to cluster data in geometric space (Sindhwani et al., 2005; Zhou et al., 2008). Since the motion segmentation problem is based on

continuous time data sequences, it is possible to base the clustering on temporal discontinuities in the data stream. We use the PCA segmentation approach (Barbic et al., 2004) outlined in the previous section to provide a coarse initial segmentation of the data.

Each raw motion capture frame can be expressed as a pose vector, $\mathbf{x} \in \Re^d$, where $d = 56$. This high-dimensional vector can be approximated by the low-dimensional feature vector, $\theta \in \Re^m$, using the linear projection:

$$\theta = \mathbf{W}^T(\mathbf{x} - \mu), \tag{8}$$

where $\mathbf{W}$ is the principal components basis and μ is the average pose vector, $\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The projection matrix, $\mathbf{W}$, is learned from a training set of $N = 300$ frames of motion capture data. $\mathbf{W}$ consists of the eigenvectors corresponding to the m largest eigenvalues of the training data covariance matrix, which are extracted using singular value decomposition (SVD). Transitions are detected using the discrete derivative of reconstruction error; if this error is more than 3 standard deviations from the average of all previous data points, a motion cut is introduced.

We found that this method provides a better starting point than traditional unsupervised clustering methods, such as k-means, which do not consider temporal information. Many of the clustering errors generated by the coarse segmentation are detected by pruning clusters based on a small set of labels solicited from the user. We ask the user to label the endpoints of the coarse segmentation and perform a consistency check on the labels; if both endpoints have the same label, the segment is potentially pure; however if the labels of the endpoints disagree, we add a new cut in the middle of the segment and query the user for the label of that point. Clusters shorter than a certain duration (1% of total sequence length) are eliminated from consideration. The remaining clusters are used to initialize the support vector machine classifiers; labels from the end points are propagated across the cluster and the data is used to initialize the SVMs. This process requires the user to label only 20–30 frames.

3.1.2 Active Learning

The clusters created by the coarse PCA segmentation, and refined with the user queries, are used to train a SVM classifier with both labeled and unlabeled samples. Semi-supervised support vector machines are regarded as a powerful approach to solve the classification problem with large data sets. Learning a semi-supervised SVM is a non-convex quadratic optimization problem; there is no optimization technique known to perform well on this topic (Chapelle et al., 2008). However, our solution is a little different to the traditional methods based on linear or non-linear programming. Instead of searching for the global maximum solution directly, we use a simple optimization approach which may not identify the optimal margin hyperplane but will help the classifier decide which unlabeled samples should be added into the training set to improve the classification performance. We then query the user for the class labels of each of the selected samples and add them back to the training set. Suppose the labeled samples are denoted by $\mathbf{L} = \{x_1, x_2, ..., x_l\}$ and the unlabeled samples are $\mathbf{U} = \{x_{l+1}, x_{l+2}, ..., x_n\}$, the SVM classification problem can be represented as find-

Input: The complete data set with labeled set $\mathbf{T}$ and unlabeled set $\mathbf{U}$
Output: The optimal SVMs hyperplane to separate the available data into two groups
Initialization: Calculate the initial hyperplane by using SVMs on the clustering data set $\mathbf{T}$;
while *the variation classification hyperplane is not stable* **do**
    Calculate the distance $\mathbf{d}$ between unlabeled set $\mathbf{U}$ and the current SVMs hyperplane $\mathbf{w_i}$;
    Query the unlabeled sample x_{l+i} with the smallest distance d_i;
    Manually label the sample x_{l+i};
    Update the labeled set as $\mathbf{T} = \mathbf{T} \cup \{x_{l+i}\}$ and unlabeled set as $\mathbf{U} = \mathbf{U} \setminus \{x_{l+i}\}$;
    Re-train the SVM classifiers $\mathbf{w_{i+1}}$ with labeled set $\mathbf{T}$;
end

Algorithm 1: Proposed active learning algorithm.

ing the optimal hyperplane with labeled samples that satisfies the equation:

$$\begin{array}{ll} \min_{\mathbf{w},b,\epsilon} & C\sum_{i=1}^{l}\epsilon_i + \|\mathbf{w}\|_2 \\ s.t. & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \quad i = 1, ..., l \end{array} \tag{9}$$

where ϵ_i is a slack term such that if $\mathbf{x_i}$ is misclassified and C is the constant of the penalty of the misclassified samples. All possible hyperplanes that could separate the training data as $f(\mathbf{x_i}) > 0$ for $y_i = 1$ and $f(\mathbf{x_i}) < 0$ for $y_i = -1$ are consistent with the version space $\mathcal{V}$. Tong and Koller (2002) have shown that the best way to split the current version space into two equal parts is to find the unlabeled sample whose distance in the mapping space is close to the current hyperplane $\mathbf{w}_i$. The description of our method is detailed in Algorithm 1.

The traditional initialization method arbitrarily selects samples to include in the training sets. However, randomly choosing samples may lead to sampling bias which make the SVM classifier unable to achieve the global maximum. In our approach, the labels of samples in each viable cluster are set as the majority labels of querying samples. This converts learning a semi-supervised SVM into a classical SVM optimization problem. However, the clustering strategy does not guarantee that the decision boundary is optimal since the clustering step is not reliable. It merely gives a good initial hyperplane; active learning is still required to perfect the solution.

In our experiments, the SVM classifier was implemented with the SVM-KM toolbox using a polynomial kernel (Canu et al., 2005); multi-class classification is handled using a one vs. all voting scheme. Instead of using a *hard margin* for the SVM, our method relies on a *soft margin* restriction in classification. A hard margin forces both labeled and unlabeled data out of the margin area, whereas the soft margin allows unlabeled samples to lie on the margin with penalties. With limited training samples, we find that the hard margin restriction is so restrictive that it may force the separating hyperplane to a local maximum.

To demonstrate this approach, we used the publicly available Carnegie Mellon Mo-

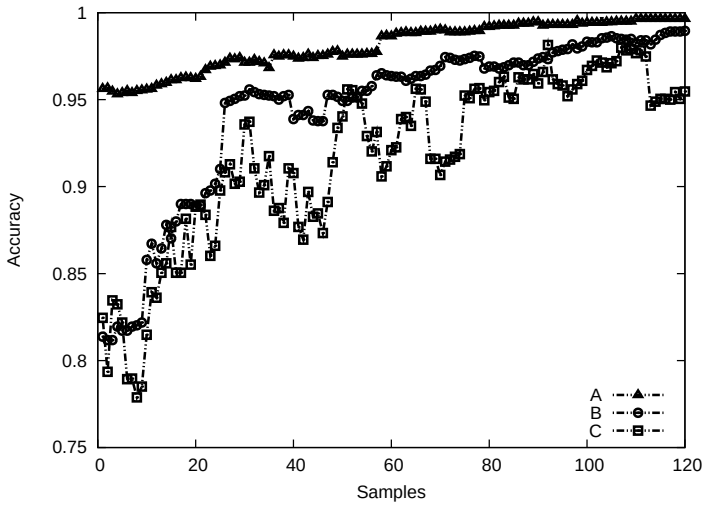


Figure 3: Improvement in quality of SVM segmentation as additional labels are acquired using active learning. The proposed method (A) benefits through intelligent initialization and margin-based selection of active learning queries.

tion Capture dataset (`http://mocap.cs.cmu.edu/`) collected with a Vicon motion capture system. Subjects wore a marker set with 43 14mm markers that is an adaptation of a standard biomechanical marker set with additional markers to facilitate distinguishing the left side of the body from the right side in an automatic fashion. The dataset contains a bunch of sequences with different human actions; to evaluate our method we selected 15 sequences that include actions such as running, swinging, jumping, and sitting. The first baseline (C) is trained using data that is sampled at random (with uniform distribution) from the activity sequence. The second (B) is initialized using a random segmentation but employs our proposed margin-based approach for generating instances for the user to label. The third (A) is our proposed approach and employs an unsupervised clustering to initialize the segmentation followed by margin-based sampling for identifying informative active learning query instances.

We evaluate the quality of segmentation using classification accuracy. Figure 3 shows how this accuracy improves with additional training data for each of the methods. Clearly, adding training data in a haphazard manner (C) leads to an inefficient form of active learning. The second method (B) demonstrates the benefits of our margin-based method for selecting queries for active learning. Finally, the accuracy curve for the proposed method (A) shows the boost that we obtain through intelligent initialization using unsupervised clustering. In comparison to a fully supervised SVM trained with 100 samples, our method achieves the same 95% accuracy with only half the data (40 samples).

3.2 Motif Discovery and Feature Selection

When classifying the IMU data, we focused on using motif discovery and feature selection to improve classification accuracy. Also, without feature selection, the time required to train the CRF on the entire motif-based feature set can become prohibitively large.

Our training approach can be described as follows. First, we discover motifs in the data collection, essentially learning a mapping to convert a given local window of IMU data from a multi-dimensional time series signal to a sequence of discrete symbols. Second, we define a series of low-level binary-valued features over motifs and pairs of motifs. From a large pool of candidate features, we select those that are most informative using an iterative approach, described below. Next, we learn a Conditional Random Field whose observations are defined over the set of selected features and whose output is over the set of action labels. The incremental feature selection and CRF training are iterated until the training set error converges. The final CRF is then evaluated on the test set. Each of these stages is detailed below.

The first step in motif discovery is to discretize the continuous IMU signal into symbolic subsequences. Figure 4(b) illustrates this process. The raw data $T = \{t_1, t_2, \ldots, t_n\}$ (black line) is transformed into a piecewise continuous representation $S = \{s_1, s_2, \ldots, s_m\}$ (green line) using the Piecewise Aggregate Approximation (PAA) algorithm, which computes an average of the signal over a short window: $s_i = \frac{m}{n} \sum_{j=\frac{n}{m}(i-1)+1}^{\frac{n}{m}} t_j$. This is then mapped to a symbolic representation using "break points" (red lines) that correspond to bins; these are generated so as to separate the normalized subsequences (under a Gaussian assumption) into equalized regions. Thus, a continuous 1-D signal can be represented as a sequence of discrete symbols.

3.2.1 Motif Matching

To compare symbolic sequences in a manner that is both computationally efficient and robust to signal noise (i.e., corresponding to symbol mismatch), we use a matching metric that relies on random projections (Chiu et al., 2003). Two motifs are designated as matching if they agree on k symbol positions. Figure 4(c) gives an example with $k = 2$, where the randomly-selected columns 1 and 3 are used to compare motifs. In this example, motifs 1 and k, 2 and 3, and 4 and j all match. These matches can be summarized by incrementing entries in a symmetric match table (where rows and columns correspond to motifs), as shown in Figure 4(d). Accumulating counts in this manner using several different random projections can enable us to efficiently match long motifs in a manner that is robust to occasional symbol errors.

3.2.2 Feature Selection for Conditional Random Fields

A key aspect of the proposed method is that we automatically select informative features from a large pool of candidates defined over motifs. As validated in our experiments, this leads to a significant improvement over CRFs trained directly on the raw data. We define three types of binary features over our motifs to form a pool of over

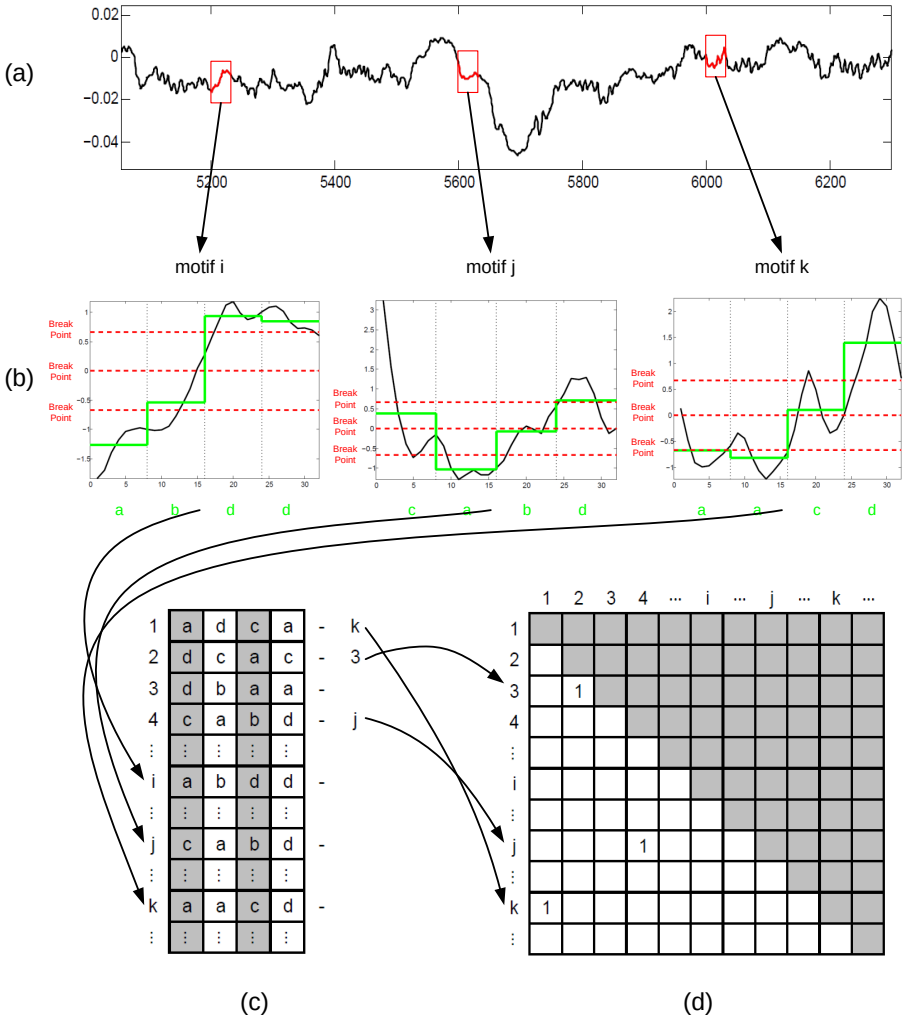


Figure 4: Motif discovery (see text for details). Visualization of motif discovery (illustrated in 1D). (a) Raw data; (b) Discretization using PAA and conversion to symbolic sequence; (c) Random projections (shadowed columns) used for projection; (d) Recorded collisions in the collision matrix.

4000 features, from which our goal is to select a small subset that can maximize the conditional log likelihood, without overfitting to the training data.

We adopt the following greedy forward selection procedure, where each feature in the pool is considered in turn and the best feature at each iteration is added. Specifically, we initialize the candidate set $\mathcal{C}$ with the pool of available features and the subset of selected features $\mathcal{S}$ to be empty. At each iteration, we evaluate every potential feature $f_\lambda \in \mathcal{C}$ individually by considering a CRF with $\mathcal{S} \cup f_\lambda$ and select the feature that maximizes the gain of log-likelihood $G(\lambda, f_\lambda) = L(\mathbf{Y}|\mathbf{X}; \mathbf{w}, \lambda) - L(\mathbf{Y}|\mathbf{X}; \mathbf{w})$. This best feature is added to $\mathcal{S}$ and removed from $\mathcal{C}$. We continue selecting features until the CRF error (computed on a hold out set) ceases to improve.

Unfortunately, a straightforward implementation of this procedure is extremely time consuming since it requires an expensive computation for every potential feature at each iteration. In particular, the normalization term of the CRF, $Z(\mathbf{x}^{(k)})$ must be calculated every time the gain $G(\lambda, f_\lambda)$ is evaluated. Motivated by work on kernel CRFs (Lafferty et al., 2001) and image segmentation (Levin and Weiss, 2009), we employ a first-order approximation method. Consider that the log likelihood function $L(\mathbf{y}|\mathbf{x}; \mathbf{w}, \lambda)$ could be approximated by its first-order Taylor expansion:

$$L(\mathbf{Y}|\mathbf{X}; \mathbf{w}, \lambda) = L(\mathbf{Y}|\mathbf{X}; \mathbf{w}) + \lambda \frac{\partial L(\mathbf{Y}|\mathbf{X}; \mathbf{w}, \lambda)}{\partial \lambda} \mid_{\lambda=0} .$$

In this equation, the second term can be expressed as:

$$\frac{\partial L(\mathbf{Y}|\mathbf{X}; \mathbf{w}, \lambda)}{\partial \lambda} \mid_{\lambda=0} = E[f_\lambda, \lambda] - \tilde{E}[f_\lambda, \lambda],$$

where $\tilde{E}[f_\lambda, \lambda] = \sum_k \sum_{i=1}^{T} f_\lambda(y_{i-1}^{(k)}, y_i^{(k)}, \mathbf{x}^{(k)}, i)$ represents the empirical expectation and $E[f_\lambda, \lambda] = \sum_k \sum_{i=1}^{T} \sum_{y'} p(y'|\mathbf{x}^{(k)}, \mathbf{w}, \lambda) f_\lambda(y_{i-1}, y_i, \mathbf{x}^{(k)}, i)$ is the model expectation. Employing this approximation achieves significant computational benefits in practice.

Our proposed method is agnostic to the choice of features. Motivated by Vail et al. (2008), we employ the following three types of features. In our case, these are computed over motif patterns rather than the raw data, and all are two-valued features. The function $\delta(.)$ is 1 if its argument is true and 0 otherwise.

1. **Identification features:** $f(y_{i-1}, y_i, \mathbf{X}, i) = \delta(y_i = motif_k)$. These features constitute the basic units of actions and are computed at a node level. They verify that the action label at time t corresponds to motif k.
2. **Transition features:** $f(y_{i-1}, y_i, \mathbf{X}, i) = \delta(y_{i-1} = motif_j)\delta(y_i = motif_k)$. These features capture the first-order Markov transition properties between adjacent motifs. The transitions may appear both between different actions or within the same action and are designed to overcome the lack of synchronization between motifs computed over different dimensions of a multi-dimensional signal.
3. **Observation features:** $f(y_{i-1}, y_i, \mathbf{X}, i) = \delta(y_i = motif_k) g_i(motif_k)$. In this definition, $g_t(motif_k)$ represents the magnitude average of motif k. These features make the magnitude information for a motif available to the CRF; that information is lost in a typical symbolic motif representation. Observation features recover it by returning the mean magnitude of the motif.

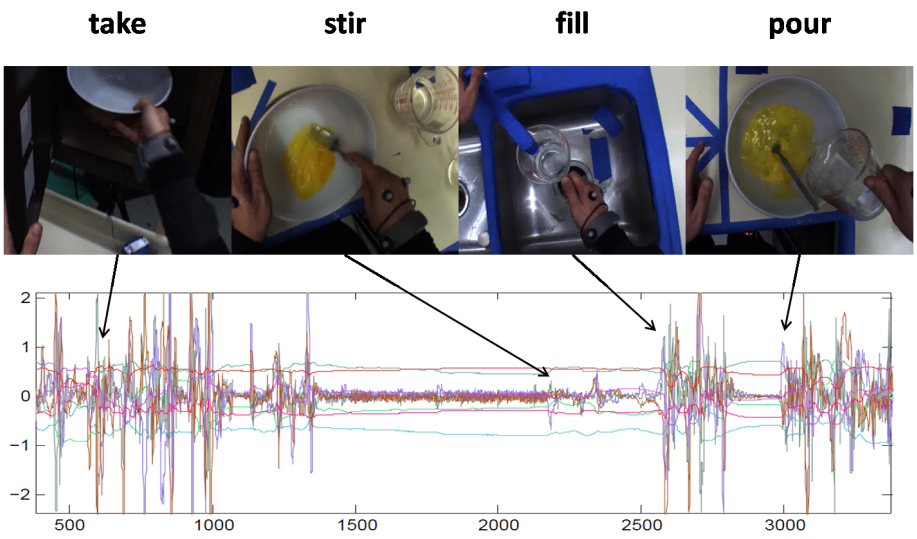


Figure 5: CMU-MMAC IMU dataset: example actions and corresponding data. The plotted data comes from five IMU sensors with 3-axis absolute orientation, angular velocity and instantaneous acceleration.

4 Results

Our experiments employ the publicly-available CMU Multi-Modal Activity Dataset (CMU-MMAC) (De la Torre et al., 2008). Here we describe our classification results on the inertial measurement unit (IMU) portion of the dataset, which was collected by five MicroStrain 3DM-GX1 sensors attached to the subject's waist, wrists and ankles. The IMU is a cost-effective and unobtrusive sensor (see Figure 5) but generates noisier data than the richer motion capture data, making the classification problem substantially more difficult.

Table 1: Comparison of average classification accuracy of CRF on 16 sequences (with feature selection) against several baselines.

Approach	Parameters	Accuracy
HMM	dim=8	8.22%
	dim=16	12.09%
	dim=32	25.60%
	dim=full(45)	16.74%
kNN	k=1	34.47%
	k=3	36.52%
CRF	raw features	30.02%
	proposed	**44.19%**

The dataset consists of unscripted recipes performed by several subjects in a kitchen. Thus, there is considerable variability in the manner in which the task is performed. The data corresponding to a given recipe consists of approximately 10,000 samples collected at 30 Hz over a period of about 6 minutes. Each frame includes 3-axis absolute orientation, angular velocity and instantaneous acceleration from each of the five sensors, leading to a 45-dimensional feature vector. Our experiments focus on the recipes that have been manually annotated into a series of actions (e.g., "open fridge" or 'stir brownie mix"); these correspond to the "make brownies" task. We downsample the raw data by a factor of 10.

Table 1 summarizes the classification results for 14 actions (Table 2). We compare the proposed CRF method against several baselines: CRF on raw features, HMM with various parameters and kNN. Clearly, the proposed method outperforms all of these approaches on the challenging test set. Clearly, the feature set results in significant benefits in terms of improved accuracy on the test set. We compared the overall performance of our method against a set of standard classifiers (K-NN, HMMs, and Bayesian networks). Our method outperforms the best performing HMM (25.6% accuracy) and the best overall alternative (k-NN, K=13, 38.22% accuracy).

Table 3 shows a comparison between the CRF with intelligent feature selection and the SVMs trained with the raw features. We also evaluated combining the SVM with temporal filtering to penalize frequent class label changes. The SVM performance is comparable to the best alternative method (k-NN) but does not do as well as the CRF plus feature selection, even with the temporal filtering.

Figure 6 illustrates the segmentation results of different methods on data sequence 1. For the *stir* activity which appears most frequently in all 14 activities, all approaches exhibit good performance compared with the ground truth (red). Moreover, CRF performs better than other two methods on activities *none* and *pour* which account for a high percentage of the total frames in testing. However, all approaches continue to perform relatively poorly in several activities that barely appeared in the testing sequence.

Table 2: List of actions

1. none	5. pour	9. stir	13. twist on
2. close	6. put	10. switch on	14. walk
3. crack	7. read	11. take	
4. open	8. spray	12. twist off	

Table 3: Classification accuracies for our proposed approach (CRF with feature selection) against SVM and SVM plus temporal filtering. Both SVM approaches use the raw IMU features.

Seq No.	CRFs	SVMs	SVM plus filter
1	49.48%	50.21%	51.66%
2	33.86%	39.07%	38.92%
3	37.97%	44.74%	50.39%
4	53.66%	37.45%	33.47%
5	32.70%	32.80%	33.52%
6	51.78%	48.71%	48.29%
7	37.75%	8.29%	6.90%
8	43.00%	39.13%	37.38%
9	45.93%	11.13%	13.15%
10	44.90%	49.87%	50.56%
11	48.12%	32.72%	36.83%
12	46.80%	49.35%	48.58%
13	47.39%	18.54%	18.17%
14	45.17%	35.74%	34.99%
15	38.10%	50.05%	49.03%
16	41.92%	33.05%	32.84%
Ave	**44.19%**	**36.23%**	**36.32%**

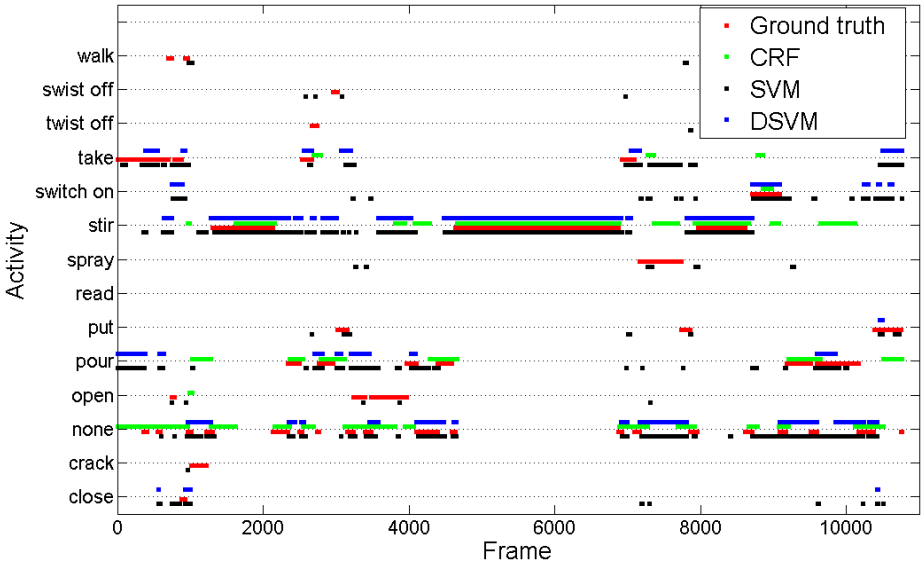


Figure 6: Segmentation results for testing sequence 1 with all three approaches. The x axis shows the 14 classes and the y axis the frame number of the sequence.

5 Discussion

Supervised classifiers have achieved good results on numerous activity recognition tasks. However, labeling a large dataset by hand remains a painful and time-consuming task. Hence, the goal of a system builder is often to achieve a reasonable level of performance while minimizing the number of required labels. To address this problem, we bootstrap our classifiers with a small initial set of data from an unsupervised clustering and use active learning to iteratively identify the most informative set of labels. Our procedure is accurate and sample-efficient at classifying motion capture data. In spite of the inaccuracies of the PCA-based unsupservised segmentation, it provides a good initial point for training a set of SVMs that can be improved by margin-based active learning. Although SVM-based active learning has been used for data annotation, experiment results indicate that our proposed method can converge faster thanks to the initialization.

In applications where less intrusive motion sensors such as inertial measurement units (IMUs) are preferred, our results suggest that attention to feature generation and selection can result in superior classification performance. Motifs are an excellent way to characterize a single dimension of time-series data, and can be rapidly and robustly identified using random projection techniques. In our research, we demonstrate that these single-dimensional motifs are informative features, and that supervised classifiers can be used to learn the linkage patterns between motifs in different dimensions of the IMU data. Since redundant motif features not only increase the computing cost of classification models but also lead to overfitting of the recognition result, we demonstrate the use of feature selection to reduce the large candidate set of motifs.

Although experimental results show that our proposed method is much better than other supervised approaches, the overall classification accuracy on IMU data is still relatively low. This is due to the large amount of transition data in the CMU-MMAC; rathing than executing a sequence of prompted actions, CMU-MMAC contains natural sequences where the subjects are simultaneously performing multiple actions while cooking recipes in a kitchen mock-up. Often, there is no obvious interval between two actions, and even human labelers demonstrate a low rate of inter-coder reliability when labeling more complicated sequences of the motion sequences. Spriggs et al. (2009) summarizes some of the issues of activity recognition with MMAC database. However, since activities in the CMU-MMAC dataset are highly representative of people's actual household activities, we believe that our work represents a promising step toward achieving sample-efficient human activity recognition for home environments.

6 Conclusion and Future Work

In this article, we present two methods for improving the recognition of human activities from motion data: 1) an active learning approach for sample-efficiency and 2) intelligent feature selection for improving classification accuracy. We demonstrate that our segmentation technique is comparable to manual segmentation while requiring only a fraction of the labels needed by a fully-supervised method. Also by linking our feature representation to the existence of 1-D motifs we can improve on classification performance over the raw IMU data. The CRF efficiently learns the cross-dimensional linkages between motifs, eliminating the need for multi-dimensional motif matching. Since increasing the number of features results in a quadratic increase in the number of parameters, we employ greedy feature selection in conjunction with a first-order approximation method based on reductions of the conditional log-likelihood error to achieve robust recognition while retaining computational feasibility. We believe that improving the accuracy and sample efficiency of supervised classification methods for human activity recognition will facilitate the usage of human motion data in future living assistance systems.

Our future work focuses on the problem of higher-level action recognition using hierarchical techniques; rather than recognizing low-level actions (e.g., stir or bake), we seek to identify composite action sequences, such as recipe the person is cooking or the chore that they are performing. Hierarchical Bayesian models have been shown good performances on many applications (Liao et al., 2007; Lin et al., 2008). Our goal is to create sample-efficient hierarchical techniques that can learn model parameters from small amounts of data, by leveraging contextual clues such as object and location.

7 Acknowledgements

This research was supported by the NSF Quality of Life Technology Center under subcontract to Carnegie Mellon and NSF award IIS-0845159.

References

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2(4):319–342.

Angluin, D. (2004). Queries revisited. *Theoretical Computer Science*, 313(2):175–194.

Arikan, O., Forsyth, D., and O'Brien, J. (2003). Motion synthesis from annotations. *ACM Transactions on Graphics*, 22(3):402–408.

Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M., and Marks, I. (1990). Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems*, pages 566–573.

Barbic, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J., and Pollard, N. (2004). Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, pages 185–194.

Baum, E. and Lang, K. (1992). Query learning can work poorly when a human oracle is used. In *International Joint Conference on Neural Networks*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Canu, S., Grandvalet, Y., Guigue, V., and Rakotomamonjy, A. (2005). SVM and kernel methods Matlab toolbox.

Chang, E., Tong, S., Goh, K., and Chang, C. (2005). Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*.

Chapelle, O., Sindhwani, V., and Keerthi, S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233.

Chiu, B., Keogh, E., and Lonardi, S. (2003). Probabilistic discovery of time series motifs. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–498.

Cohn, D. (1996). Neural network exploration using optimal experiment design. *Neural Networks*, 9(6):1071–1083.

Culotta, A. and McCallum, A. (2004). Confidence estimation for information extraction. In *Proceedings of HLT-NAACL*, pages 109–112.

Dabiri, F., Vahdatpour, A., Noshadi, H., Hagopian, H., and Sarrafzadeh, M. (2008). Ubiquitous personal assistive system for neuropathy. In *Proceedings of the Second International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, pages 17–22.

Dasgupta, S. and Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 208–215.

De la Torre, F., Hodgins, J., Bargtell, A., Artal, X., Macey, J., Castellis, A., and Beltran, J. (2008). Guide to the CMU Multimodal Activity Database. Technical Report RI-08-22, CMU.

Fu, W., Ray, P., and Xing, E. (2009). DISCOVER: a feature-based discriminative method for motif search in complex genomes. *Bioinformatics*, 25(12):321–329.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.

Keogh, E. (2002). Efficiently finding arbitrarily scaled patterns in massive time series databases. In *Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 253–265. Springer Verlag.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286.

Kumar, S. and Hebert, M. (2003). Discriminative fields for modeling spatial dependencies in natural images. In *Advances in Neural Information Processing Systems*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmentation and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289.

Levin, A. and Weiss, T. (2009). Learning to combine bottom-up and top-down segmentation. *International Journal of Computer Vision*, 81(1):105–118.

Liao, L., Fox, D., and Kautz, H. (2007). Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26(1):119–134.

Lin, J., Keogh, E., Lonardi, S., and Patel, P. (2002). Finding motifs in time series. In *ACM SIGKDD Workshop on Temporal Data Mining*, pages 53–68.

Lin, T., Ray, P., Sandve, G., Uguroglu, S., and Xing, E. (2008). BayCis: A Bayesian hierarchical HMM for cis-regulatory module decoding in metazoan genomes. In *Proceedings of the 12th Annual International Conference on Research in Computational Molecular Biology*, pages 66–81.

Liu, D. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

McCallum, A. (2003). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*.

Minnen, D., Starner, T., Essa, I., and Isbell, C. (2006). Discovering characteristic actions from on-body sensor data. In *Proceedings of the 10th International Symposium on Wearable Computers*, pages 11–18.

Mitchell, T. (1982). Generalization as search. *Artificial intelligence*, 18(2):203–226.

Plath, N., Toussaint, M., and Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th International Conference on Machine Learning*, pages 817–824.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Sindhwani, V., Niyogi, P., and Belkin, M. (2005). Beyond the point cloud: frame transductive to semi-supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 824–831.

Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). Conditional random fields for contextual human motion recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, pages 1808–1815.

Spriggs, E., De la Torre, F., and Hebert, M. (2009). Temporal segmentation and activity classification from first-person sensing. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–24.

Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Troster, G. (2008). Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2):42–50.

Stiefmeier, T., Roggen, D., and Troster, G. (2007). Fusion of string-matched templates for continuous activity recognition. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, pages 41–44.

Tanaka, Y., Iwamoto, K., and Uehara, K. (2005). Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2):269 – 300.

Tappen, M., Liu, C., Adelson, E., and Freeman, W. (2007). Learning Gaussian conditional random fields for low-level vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 107 – 118.

Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45 – 66.

Vahdatpour, A., Amini, N., and Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *Proceedings of the International Joint Conference on Artficial Intelligence*, pages 1261–1266.

Vail, D., Veloso, M., and Lafferty, J. (2007). Conditional random fields for activity recognition. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8.

Vail, D., Veloso, M., and Lafferty, J. (2008). Feature selection for activity recognition in multi-robot domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1415–1420.

Vail, D. L. (2008). *Conditional Random Fields for Activity Recognition*. PhD thesis, Carnegie Mellon University.

Wang, L., Chan, K., and Zhang, Z. (2003). Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *Proceedings of Computer Vision and Pattern Recognition*, pages 629–634.

Wu, W., Au, L., Jordan, B., Stathopoulos, T., Batalin, M., Kaiser, W., Vahdatpour, A., Sarrafzadeh, M., Fang, M., and Chodosh, J. (2008). The Smartcane System: an assistive device for geriatrics. In *Proceedings of the ICST 3rd International Conference on Body Area Networks*, pages 1–4.

Zhou, F., De la Torre, F., and Hodgins, J. (2008). Aligned cluster analysis for temporal segmentation of human motion. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pages 1–7.