Recognizing Events with Temporal Random Forests

David Demirdjian
Toyota Research Institute
Cambridge, MA 02142, USA

Chenna Varri
Toyota Research Institute
Cambridge, MA 02142, USA

ABSTRACT

In this paper, we present a novel technique for classifying multimodal temporal events. Our main contribution is the introduction of temporal random forests (TRFs), an extension of random forests (and decision trees in general) to the time domain. The approach is relatively simple and able to discriminatively learn event classes while performing feature selection in an implicit fashion. We describe here our ongoing research and present experiments performed on gesture and audio-visual speech recognition datasets comparing our method against state-of-the-art algorithms.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Models; H.1.2 [**Models and Principles**]: User/Machine Systems—*human information processing*

General Terms

Algorithms, Performance, Reliability

Keywords

Temporal event recognition, decision trees

1. INTRODUCTION

The classification of temporal events is a fundamental issue in fields such as computer vision, speech recognition and signal processing in general. In recent years, the development of temporal event classification has been driven by applications such as human-machine interaction and smart surveillance, where understanding human behavior and activities from various inputs is critical. Learning classes of temporal events is a very challenging problem because it involves building flexible and comprehensive models of signal dynamics that can account for cross-user and intra-user variations. For example, gesture recognition techniques need to model large variations in the temporal and spatial domains, e.g. subjects performing the same gesture with different styles and speed.

Graphical models such as Hidden Markov Hodel (HMM) and Conditional Random Fields (CRFs) have become state-of-the-art for modeling sequences. They have proven to be well adapted to the recognition of speech, human gestures and activities. However, the training process for graphical models is still challenging. The difficulty partly comes from the representation of the observations used in these models, e.g. sequences of arbitrary length, but also from the high dimensionality of the input data, which requires feature selection to be used in the classification process. In this paper, we address these challenges by introducing an approach for temporal event classification based temporal random forests (TRFs).

ICMI-MLMI'09, November 2–4, 2009, Cambridge, MA, USA.

2. PREVIOUS WORK

There is an extensive literature dedicated to modeling temporal sequences. We report here approaches most relevant to the gesture and speech recognition communities.

Trajectory-based approaches have been proposed. Work such as [1, 7] relies on an elastic similarity between temporal sequences and Dynamic (Space-)Time Warping techniques to align query and model gestures. In such approaches, accurate learning of complex trajectories or long sequences (e.g. activity recognition) can be computationally difficult. In addition, they usually do not scale well to problems where the data is high-dimensional or multimodal.

Bag-of-words representations have also become popular for the recognition of signals in the computer vision and language communities. Zelnik-Manor *et al* [17] first introduced marginal histograms of spatial-temporal gradients at several temporal scales to cluster and recognize video events. More recently, [8, 5] have used 'bag of video words' representations to categorize human actions. Although these techniques are attractive as they show good performance on simple human actions, they are limited because temporal information is partially lost and therefore are unable to learn complex temporal event classes.

Graphical models have been the most successful in modeling temporal sequences. Directed graphical models, like Hidden Markov Models (HMM) [10], and many extensions have been used successfully to recognize speech, gestures, sign language and activities. Undirected graphical models have also been used. Sminchisescu *et al.* [12] applied Conditional Random Fields (CRFs) [4] to classify human motion activities (i.e. walking, jumping, etc); their model can also discriminate subtle motion styles like normal walk and wan-

der walk. Although these graphical models have been successful in recognizing temporal sequences, they are sensitive to training parameters (e.g. setting the number of hidden states for HMMs). In theory, models such as CRFs are able to incorporate features corresponding to long range dependencies. In practice, however, the use of this property is limited because of the additional data requirement, e.g. the number of unknowns in the CRF model grows linearly with the 'dependency' window size.

In this paper we propose a novel approach to classify event classes using Temporal Decision Trees (TDTs). Decision trees and random forests have been long time used in data mining and machine learning communities. Recently, [6] introduced such structures for object classification. [15] introduced Temporal Decision Trees (TDTs) to classify simple event categories using discrete measurements. Our extension of TDTs is two-fold. First, we use a more general notion of time by using time windows, which allows a more efficient usage of local signal variations. Second, we train TDTs using ensemble learning by learning a forest of TDTs, which is less prone to over-fitting and therefore yields better generalization properties.

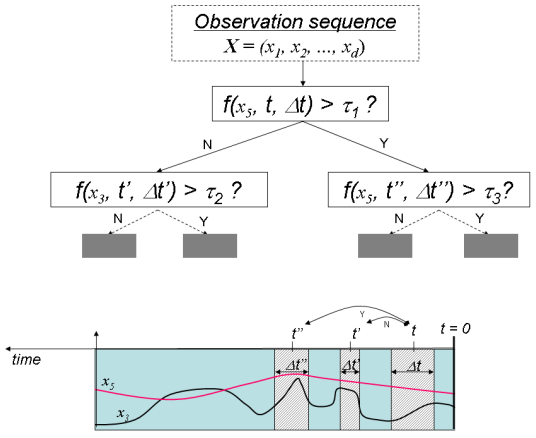

Figure 1: Example of temporal decision tree used for temporal event classification. Each node holds test on a variable $\boldsymbol{x}_k$ over a time window $(t, \Delta t)$

3. TEMPORAL RANDOM FORESTS

In this section, we describe how temporal sequences are recognized in our system using temporal event classification techniques. We introduce here a temporal random forest classifier.

In temporal sequences, we wish to learn a mapping of observations $\boldsymbol{X}$ to class labels $c \in 1, 2, ..., N$, where $\boldsymbol{X} \in I\!R^{d \times L}$ is a set of temporal observations, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_d\}$. Here, $\boldsymbol{x}_k \in I\!R^L$ is a temporal sequence of scalars and called *variable*. L, the sequence length can vary between observations $\boldsymbol{X}$. Here we wish to predict the class $c = y(X)$ of a temporal sequence $\boldsymbol{X}$.

3.1 Decision Trees and Random Forests

A decision tree is a predictive model that maps observations to a class distribution. Each node of the tree corresponds to a variable from the observation. An arc to a child represents a possible value of that variable and a leaf corresponds to the predicted class given the values of the variables.

Decision trees are built from training data in a top-down manner. Variable and corresponding test at each node are chosen so that they best separate the training examples, i.e. maximize the gain of information. Once a node is created, data are split according to the variable-test and the node creation process is repeated for all children. At the end of the learning process, leaves contain the class distributions corresponding to a data point to fall in it.

When data is high dimensional and datasets are large, building optimal decision trees becomes intractable. To overcome this issue, *random forests* [2, 6] have been proposed as an alternative. As many ensemble learning approaches, a random forest consists of multiple suboptimal decision trees, which are individually built from a small random subset of the training data using a limited random subset of tests at each node. At runtime, class distributions from all decision trees are additively combined to produce a global output.

3.2 Temporal Random Forest

A temporal decision tree (TDT) [15] is an extension of a decision tree, which includes temporal information at each node. Each node of a TDT contains a test on a variable at a given time (relative to the end of the sequence to classify). A temporal random forest (TRF) contains multiple TDTs where each TDT yields a different partition of the training data.

In our formulation, temporal information at a node is a continuous time window with center t and width Δt. Decisions are made based on the output of a filter f (e.g. mean or slope of signal) applied to a specific variable $\boldsymbol{x}_k$ over a time window $(t, \Delta t)$. In the rest of this paper, we use $f(\boldsymbol{x}_k, t, \Delta t)$ as notation for the output of filter f. Figure 1 illustrates the overall process.

The learning process involves growing a forest of TDTs in a similar fashion as a random forest [2]. Each TDT is grown as follows:

1. Random set and test selection. Subsets are selected randomly with replacement from the original training set and are used as the training set for growing each tree. While growing the tree, a subset of m variables is selected randomly at each node such that $m << d$. The selected m variables are then used to estimate the node parameters.

2. Node construction. At each node a window $(t, \Delta t)$ and a variable $\boldsymbol{x}_k$ are chosen randomly. A filter $f(\boldsymbol{x}_k, t, \Delta t)$ is then applied to all sequences $\boldsymbol{X}$ present at the node. Let p_i be the proportion of examples belonging to class c_i such that $f(\boldsymbol{x}_k, t, \Delta t) >= \tau$, where τ is a threshold. As in standard decision trees, τ is searched so that it optimizes entropy S:

$$S = \sum_{i=1}^{N} -p_i \log_2(p_i) \tag{1}$$

Optimal parameters $(t, \Delta t, \tau)$ with maximum entropy gain are stored at that node and will be used while testing. When a node contains data from a single class or reaches a maximum depth, the node is considered as terminal. Leaves of the TDT contain the posterior probabilities $p(y(X) = c)$.

3. Tree weighting. Once the tree is grown, it is evaluated on the entire training set to obtain a weight w for the tree (normalized across the entire forest).

3.3 Testing

Let T be the number of trees. A (query) temporal sequence is tested against all the TDTs in the TRF as described below:

1. At each node, stored parameters $(t, \Delta t, \tau)$ are used to test a window from the sequence. The output is used to decide to which child the sequence has to be sent to next.

2. Once it reaches a leaf the class probabilities $p_i(y(X) = c)$ at that leaf are returned as the prediction for the sequence. The predicted value returned by each tree is then multiplied by their weight.

Class prediction $\hat{c}$ for the TRF is then obtained by accounting for the output of all trees as:

$$\hat{c} = \underset{c}{\operatorname{argmax}} \sum_{i=1}^{T} w_i p_i(y(X) = c) \quad (2)$$

4. EXPERIMENTS

4.1 Gesture Recognition

We conducted experiments comparing our TRF technique with sequence models such as HMMs and CRFs. We used a gesture dataset [16] containing 13 types of arm gestures performed several times by 13 subjects with an average of 90 gesture sequences per class (each sequence is comprised of one gesture instance). The dataset includes standard gestures used in human-computer interaction, such as *Pointing*, *Circling*, *Waving*, *Greeting*. The dataset also includes some less intuitive gestures, such as *Flip Back (FB)*, *Expand Horizontally (EH)* and *Expand Vertically (EV)*. Figure 2 shows some examples of gestures.

The dataset was collected using a stereo camera. A 3D model-based pose estimation algorithm [3] was applied to recover the pose of the subjects in each frame. The joint angles $\boldsymbol{\theta}$ corresponding to the subject's upper body were retained as observations for our experiments (the dimension of $\boldsymbol{\theta}$ is 20). The techniques compared in this experiment are:

TRF. Our algorithm as described in Section 3. Filter f was chosen as an averaging filter over time windows.

HMM. We trained an HMM model per class. During evaluation, test sequences were passed through each of these models, and the model with the highest likelihood was selected as the recognized gesture.

CRF. We trained a CRF chain where every class label has a corresponding state. The CRF predicts a label per frame. During evaluation, we found the Viterbi path under the CRF model, and assigned the sequence label based on the most frequently occurring gesture label.

In our experiments, we used the implementation of HMM and CRF provided by [14]. When using HMM, we performed preliminary dimensionality reduction using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Different numbers of Gaussian mixtures and states were explored. When using CRF, we considered multiple regularizer values and dependency window sizes. We only report here the results with the highest accuracy.

To test the efficacy of our approach in recognition we divided the available data into equal size training and testing sets. The evaluation metric used for all the experiments was the percentage of sequences of the testing set for which the correct gesture label was predicted.

We performed a set of experiments to compare the different algorithms with respect to the size of the training dataset. We trained the algorithms using only a fraction μ or all of the training set. Table 1 shows the recognition performance for $\mu = 0.2, 0.5, 1.0$. In these experiments, the number of trees in TRF is 250. The TRF model shows similar of better performance than HMM. However, it provides less accurate results than the CRF method (at best, TRF only nears CRF results for $\mu = 1.0$).

μ	TRF	HMM	CRF
0.2	68.2	64.7	69.0
0.5	73.7	73.7	76.1
1.0	77.3	74.9	77.9

T	accuracy
10	74.0
50	75.6
100	77.1
250	77.3

Table 1: (left) Recognition performance (percentage accuracy) for the gesture dataset for varying fraction μ of training dataset (percentage of the original training dataset). (right) Recognition performance of TRF with varying number of trees T.

	12dB	10dB	6dB
AHMM	81.3	73.1	65.9
TRF	76.1	67.2	59.9

VHMM	42.0
TRF	51.4

Table 2: Recognition rate on the audio (left) and visual (right) data from the CUAVE dataset (isolated digits).

4.2 Audio-Visual Speech Recognition

In order to evaluate TRFs on a multimodal recognition task, we carried out some experiments on CUAVE [9], an Audio-Visual Speech Recognition (AVSR) database. The part of the database used in our experiments contains audio-visual sequences of 36 speakers speaking (isolated) English digits from "zero" to "nine". Each subject approximately spoke 50 digits. We used a training set consisting of 10 randomly selected subjects and used the remaining 8 subjects for testing. In a similar way to [11], speech noise was added from the NOISEX database [13] at various signal-to-noise ratios. We trained HMM and TRF models on the clean data and tested them using noisy conditions (at 12,10 and 6 dB). The audio-visual data was processed as follows. For each frame, the audio observations consisted of 14 MFCCs, plus first and second derivatives. Visual observations consisted of 35 discrete cosine transform (DCT) coefficients of a 16-by-16 grayscale mouth subregion, plus first derivatives.

In the following, we refer as AHMM and VHMM, the HMM models trained on audio-only and visual-only data respectively. The audio-visual HMM model is referred as MHMM.

Table 2 and Table 3 show a comparison of the recognition rates estimated for the various algorithms for the audio-only, visual-only and audio-visual data. For the visual-only data, the TRF approach produced significantly better results than the HMM model. However, for audio-only TRF produced much less accurate results. For multimodal audio-visual data, the TRF approach results were similar or slightly worse than the HMM model.

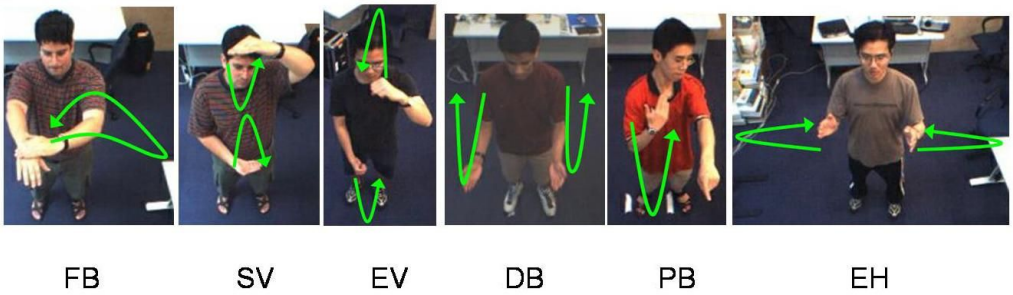

Figure 2: Example of gestures recognized by our system.

	12dB	10dB	6dB
MHMM	83.0	77.7	70.1
TRF	81.4	75.2	69.2

Table 3: Recognition rate on the audio-visual data from the CUAVE dataset (isolated digits).

5. CONCLUSION AND FUTURE WORK

In this paper, we presented our ongoing research on the application of TRFs for classifying events. TRFs learn event classes in a discriminative fashion, while implicitly performing feature selection. Tests performed at each node of the trees capture the local temporal information about the observations and perform simple feature selection by choosing the most discriminative variable. Experiments performed on gesture and AVSR datasets are encouraging and show that TRFs perform well on visual data. However, TRFs do not show a consistent improvement over standard graphical models. On the gesture dataset, CRF always provides similar or better results. On the AVSR dataset HMM outperforms TRF on audio and multimodal data. However, we believe that the results are encouraging and that our approach is promising. Indeed, TRFs do show better performance to HMM on visual data on the gesture and AVSR datasets. Note that in our experiments, the main parameter to set in TRFs was the number of trees (in HMM and CRF methods, the number of states or Gaussian mixtures had to be fine-tuned). A significant advantage of TRFs is that at testing they do not require sliding window mechanisms; they automatically *pick* the appropriate signal in the sequence and therefore perform classification on-the-fly. Finally, improvements of the TRF method can be obtained in many ways. We are currently experimenting with variants of the algorithm, *e.g.* using different filters f, and using slightly stronger classifiers at each node (*e.g.* general linear classifiers) instead of the single variable-based tests.

6. REFERENCES

[1] J. Alon, V. Athitsos, and S. Sclaroff. Accurate and efficient gesture spotting via pruning and subgesture reasoning. In *CVHCI05*, page 189, 2005.

[2] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[3] D. Demirdjian, L. Taycher, G. Shakhnarovich, K. Grauman, and T. Darrell. Avoiding the "streetlight effect": Tracking by exploring likelihood modes. In *ICCV*, pages 357–364, 2005.

[4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In *ICML*, 2001.

[5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[6] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[7] H. Li and M. A. Greenspan. Multi-scale gesture recognition from time-varying contours. In *ICCV*, pages 236–243, 2005.

[8] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[9] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy. CUAVE: A new audio-visual database for multimodal human-computer interface research. In *Proc. ICASSP, Orlando, FL, USA*, May 2002.

[10] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–286, 1989.

[11] K. Saenko and K. Livescu. An asynchronous dbn for audio-visual speech recognition. In *IEEE Spoken Language Technology Workshop*, 2006.

[12] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *Int'l Conf. on Computer Vision*, 2005.

[13] A. Varga, H. Steeneken, M. Tomlinson, and D. Jones. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. In *Tech. Rep., DRA Speech Research Unit, Malvern, England*, 1992.

[14] S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. Accelerated training of conditional random fields with stochastic meta-descent. In *ICML'06*, 2006.

[15] S. yong Koo, J. G. Lim, and D. soo Kwon. Online touch behavior recognition of hard-cover robot using temporal decision tree classifier. In *International Symposium on Robot and Human Interactive Communication*, pages 425–429, 2008.

[16] S. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden Conditional Random Fields for Gesture Recognition. In *CVPR*, 2006.

[17] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, pages 123–130, 2001.