Multi-class Boosting for Early Classification of Sequences

Katsuhiko Ishiguro
ishiguro@cslab.kecl.ntt.co.jp
Hiroshi Sawada
sawada@cslab.kecl.ntt.co.jp
Hitoshi Sakano
keen@cslab.kecl.ntt.co.jp

NTT Communication Science Laboratories,
NTT Corporation
Kyoto, 610-0237, Japan

Abstract

We propose a new boosting algorithm for sequence classification, in particular one that enables early classification of multiple classes. In many practical problems, we would like to classify a sequence into one of K classes as quickly as possible, without waiting for the end of the sequence. Recently, an early classification boosting algorithm was proposed for binary classification that employs a weight propagation technique. In this paper, we extend this model to a multi-class early classification. The derivation is based on the loss function approach, and the developed model is quite simple and effective. We validated the performance through experiments with real-world data, and confirmed the superiority of our approach over the previous method.

1 Introduction

In this paper, we are interested in multi-class classification of sequence data. For example, consider the problem of driver behavior recognition from images captured by a camera installed in a vehicle [11]. Recognition of driver behavior is crucial for driver assistance systems that make driving comfortable and safe, and we would like to predict and classify a behavior **as quickly as possible**: we don't want to wait until the behavior is over. If we detect a sign of dangerous movements such as mobile phone use while driving, we would like to warn the driver quickly before the behavior causes any accidents. If the driver looks into a side mirror frequently, the system may predict there will be a lane change. Thus the assistance system can suggest to the driver the best timing for changing lanes.

This kind of classification task is called **"early classification (recognition),"** and is important for many practical problems including on-line handwritten character recognition [1], and speech recognition systems [7].

We can apply either a generative model such as HMM [9] or a discriminative model such as SVM [14], CRF [8] to general sequence recognition problems [3]. In general, generative models such as HMM are flexibly applicable to the recognition of (sub)sequence $x_{1:t}$ but are trained to maximize the likelihood of the "total" sequence $x_{1:T}$, not for early classification of sequences. Discriminative models such as SVM and CRF are said to be superior in the classification score to generative models. However, we need to fix the length of sequences

a priori in typical applications. Thus we cannot compute the classification results if the test sequence is shorter than the predefined length of sequences.

In this paper, we focus to another famous discriminative model, i.e. Adaboost [4, 5], because some researchers have proposed early classification extensions. Sochman and Matas [12] combined the idea of sequential decision making problem in order to balance the trade-off between the precision and the quickness when the decision (recognition) should be made. [6] have also proposed a boosting scheme which is able to handle early classification and variable length sequence inputs. Uchida and Amamoto [13] proposed a new boosting algorithm that we refer to "Earlyboost." Earlyboost enables early classification of sequences by time frame-wise weak classifiers and weight propagation technique which preferentially corrects the classifiers to reduce misclassifications made by the previous time frames.

Unfortunately, in these three models the authors have studied only a binary classification (the number of classes is $K = 2$) problem. However many real-world applications including the driver behavior recognition example require multi-class ($K > 2$) classification scheme. The authors of [13] handled such cases by constructing 1 vs. 1 strong classifiers for all possible $\binom{K}{2}$ class pairs. Obviously, this approach is inefficient with large K. Also the resultant strong classifier is not optimized for K-class classification problem because it is just a collection of independent binary classifiers.

To this end, we present a multi-class extension of Earlyboost, called Earlyboost.MH (Fig. 1). In our model, the number of weak classifiers to be trained is proportional to K. The idea is to combine the weight propagation technique with existing multi-class Adaboost [10]. This combination is intuitively understandable and practically useful. Based on the exponential loss approach [5], we derive and validate the soundness of the model formulation. In the next section, we briefly provide some background. In the third section, we present our new model, and we examine its performance through real-world sequence data in the fourth section. The last section concludes the paper.

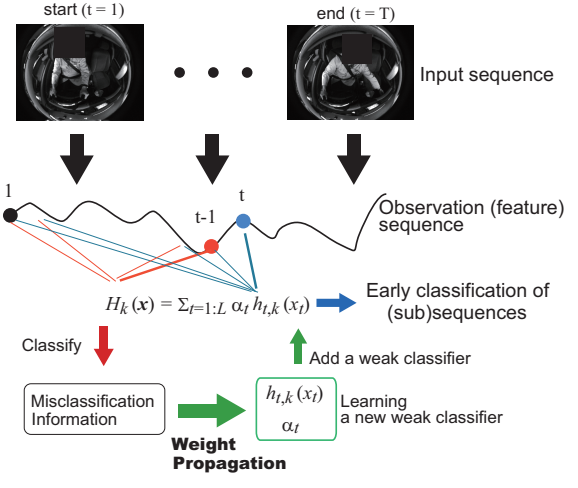

Figure 1: Overview of a concept of the multi-class early classification boosting. Final strong classifiers consists of time frame-wise weak classifiers. The weak classifiers are learnt through weight propagation technique to achieve early classification of (sub)sequences.

2 Background

2.1 Adaboost

Let us first review the well-known Adaboost [4]. We have a training dataset $\mathscr{D} = \{x_i, y_i\} : i = 1, 2, \ldots, N$ where $x_i \in \mathbb{R}^d$ is an observation and $y_i \in \{1, -1\}$ is its class label. The goal of Adaboost learning is to construct a strong classifier $H : \mathbb{R}^d \to \{1, -1\}$ using $\mathscr{D}$ as follows:

$$H(x) = \mathrm{sign}\left(\sum_{m=1}^{M} \alpha_m h_m(x)\right), \tag{1}$$

where $h_m : \mathbb{R}^d \to \{1, -1\}$ is the m-th weak classifier, $\alpha_m \in \mathbb{R}$ is its importance weight and M is the total number of weak classifiers. Let us denote the m-subset of the final strong classifier as follows:

$$H^m(x) = \mathrm{sign}\left(\sum_{l=1}^{m} \alpha_l h_l(x)\right). \tag{2}$$

Therefore the final strong classifier H is equivalent with H^M.

The empirical loss function J for H^m is defined by the following equation [5]:

$$J(H^m) = \sum_{i=1}^{N} \left(\exp\left(-y_i H^m(x_i)\right)\right) = \sum_{i=1}^{N} \left(\exp\left(-y_i \sum_{l=1}^{m} \alpha_l h_l(x_i)\right)\right). \tag{3}$$

In the learning phase, we find h_m and α_m which minimizes the loss $J(H^{m-1} + \alpha_m h_m)$ iteratively, incrementing $m \leftarrow m + 1$. As a result, we obtain the following equations:

$$h_m = \arg\min_h \epsilon_m, \tag{4}$$

$$\alpha_m = \frac{1}{2} \log\left(\frac{1 - \epsilon_m}{\epsilon_m}\right), \tag{5}$$

$$\epsilon_m = \sum_{i : y_i \neq h(x_i)} D_m(i). \tag{6}$$

$D_m(i) \in \mathbb{R}$ is a sample weight of x_i at m-th training, defined recursively as follows:

$$D_m(i) \propto D_{m-1}(i) \exp\left(-\alpha_{m-1} y_i h_{m-1}(x_i)\right). \tag{7}$$

Intuitively, Eq. (7) will be large if the classification by the weak classifier h_{m-1} and the true label are inconsistent, and will be small otherwise.

2.2 "Earlyboost"

A novel attempt to use the boosting scheme for early sequence classification was proposed by Uchida and Amamoto [13]. We call this method "Earlyboost" for convenience, though the authors of [13] did not specify a name for their model.

Let us denote that i-th sequence $\boldsymbol{x}_i$ has a number of time frame elements $x_{i,t}$: i.e. $\boldsymbol{x}_i = \{x_{i,t} \in \mathbb{R}^d\}, t = 1, 2, \ldots, T$. The number of sequences is N: thus $i \in \{1, 2, \ldots, N\}$. T is the length

of time sequences, and $t \in \{1,2,\ldots,T\}$ is the time index. The objective of Earlyboost is to learn the strong classifier H from the training data $\mathscr{D} = \{\boldsymbol{x}_i, y_i\}$:

$$H(\boldsymbol{x}_i) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x_{i,t})\right) \tag{8}$$

where $h_t : \mathbb{R}^d \to \{1,-1\}$ is the t-th frame-wise weak classifier that only accepts the samples on the t-th time frame $\{x_{i,t}\}_{i=1,2,\ldots,N}$. At each t, we find h_t and α_t by the following equations:

$$h_t = \arg\min_h \epsilon_t, \tag{9}$$

$$\alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right), \tag{10}$$

$$\epsilon_t = \sum_{i:y_i \neq h(x_{i,t})} D_t(i). \tag{11}$$

The contribution of [13] is the definition of $D_t(i)$ based on the weight propagation technique. Since h_{t-1} and h_t are learned on different sets of samples, we need to devise a way to connect them. The authors put weights over sequences i, not samples $x_{i,t}$ and propagate the weights over time frames $t-1$ and t:

$$D_t(i) \propto D_{t-1}(i)\exp\left(-\alpha_{t-1} y_i h_{t-1}(x_{i,t-1})\right). \tag{12}$$

In the classification phase, a test (sub)sequence $\boldsymbol{x}_j = \{x_{j,t}, t = 1,2,\ldots,L\}$ will be classified as follows:

$$y_j = H\left(\boldsymbol{x}_j\right) = \mathrm{sign}\left(\sum_{t=1}^{L} \alpha_t h_t\left(x_{j,t}\right)\right) \tag{13}$$

Eq. (12) implies that each weak classifier h_t learns the classification boundary at time t to minimize the classification error induced by the information up to time $t-1$. Thus the resulting strong classifier will be good for early classification of sequences, even if the sequence is short ($L \leq T$).

We can validate the formulation of Earlyboost by using the loss function approach, which has not been examined in the original paper [13]. First we define the t-subset of the final strong classifier H like Eq. (2):

$$H^t(\boldsymbol{x}_i) = \mathrm{sign}\left(\sum_{s=1}^{t} \alpha_s h_s(x_{i,s})\right). \tag{14}$$

Then we define the exponential loss function as follows:

$$J(H^t) = \sum_i \left(\exp\left(-y_i H^t(\boldsymbol{x}_i)\right)\right) = \sum_i \exp\left(-y_i \sum_{s=1}^{t} \alpha_s h_s(x_{i,s})\right). \tag{15}$$

Then Eqs.(9-12) are derived by seeking for α_t and h_t which minimizes the loss below:

$$\alpha_t, h_t = \arg\min_{\alpha,h} J\left(H^{t-1} + \alpha_t h_t\right). \tag{16}$$

Thus Earlyboost iteratively augments the strong classifier H which minimizes the loss above w.r.t. α_t and h_t, those who are dependent only on the t-th frame observations. The derivation is almost the same with Adaboost [5]. Please find details in the supplemental material.

3 Proposed: Earlyboost.MH

3.1 Algorithm

The problem with Earlyboost is that it is only a binary classifier. We propose an efficient multi-class Earlyboost that we refer to Earlyboost.MH. Our model is inspired by Adaboost.MH [10], which is a standard multi-class Adaboost. The Earlyboost.MH is derived by applying the idea of weight propagation to Adaboost.MH.

We have a set of training data $\mathscr{D} = \{\boldsymbol{x}_i, y_i\}$. Because of multi-class classification, class labels range in K values: $y_i \in \{1, 2, \ldots, K\}$ which makes contrast to Adaboost and Earlyboost. A weak classifier $h_{t,k}(x) : \mathbb{R}^d \to \{1, -1\}$ only accepts the samples on the t-th time frames. $h_{t,k}(x)$ returns 1 if x belongs to class k, and returns -1 otherwise. We also define $g_k(y) : \{1, 2, \ldots, K\} \to \{1, -1\}$ returns 1 if $y = k$, and returns -1 otherwise.

The strong classifier of Earlyboost.MH consists of K classifiers H_k, namely $H = \{H_k\}$. H_k is a one vs. all type classifier which computes the likelihood of the sequence being the member of class k, described as follows:

$$H_k(\boldsymbol{x}_i) = \sum_{t=1}^{T} \alpha_t h_{t,k}(x_{i,t}). \tag{17}$$

We also define the t-subset of H_k as:

$$H_k^t(\boldsymbol{x}_i) = \sum_{s=1}^{t} \alpha_s h_{s,k}(x_{i,s}). \tag{18}$$

We define an empirical loss function for Earlyboost.MH as follows:

$$J(H^t) = \sum_{i=1}^{N}\sum_{k=1}^{K}\Big(\exp\Big(-g_k(y_i)H_k^t(\boldsymbol{x}_i)\Big)\Big) = \sum_{i=1}^{N}\sum_{k=1}^{K}\exp\left(-g_k(y_i)\sum_{s=1}^{t}\alpha_s h_{s,k}(x_{i,s})\right). \tag{19}$$

We can readily obtain above by plugging Eq. (15) to the loss of Adaboost.MH [5, 10].

In the learning phase, we seek for $\{h_{t,k}, \alpha_t\}$ which minimizes the loss given H_k^{t-1}:

$$\alpha_t, h_{t,k} = \arg\min_{\alpha,h} \sum_{i=1}^{N}\sum_{k=1}^{K}\exp\Big[-g_k(y_i)\Big(H_k^{t-1}(\boldsymbol{x}_i) + \alpha_t h_{t,k}(x_{i,t})\Big)\Big]. \tag{20}$$

Note that the predicate of exp is expanded by α_t and $h_{t,k}$ that are to be estimated.

We solve Eq. (20) w.r.t. $h_{t,k}$ and α_t to find the minimum. After simple calculations, we obtain the following equations Eqs.(21-24). For the details of the derivation, please find the supplemental material. An optimal $h_{t,k}$ is computed as follows:

$$h_{t,k} = \arg\max_{h} r_{t,k}, \tag{21}$$

$$r_{t,k} = \sum_{i=1}^{N} g_k(y_i) h_{t,k}(x_{i,t}) D_t(i,k). \tag{22}$$

$r_{t,k}$ is a class- and frame-wise classification score which only depends on the observation at the t-th frame. The score $r_{t,k}$ will be large if the estimated label by a weak classifier and g_k

match correctly, and will be small otherwise. $D_t(i,k) \in \mathbb{R}$ is a weight of a sequence $\boldsymbol{x}_i$ for the k-th classifier at t-th frame, propagated from the time $t-1$:

$$D_t(i,k) \propto D_{t-1}(i,k) \exp\left(-\alpha_{t-1} g_k(y_i) h_{t-1,k}(x_{i,t-1})\right). \tag{23}$$

The importance weight α_t is computed as follows given maximized $r_{t,k}$:

$$\alpha_t = \frac{1}{2} \log\left(\frac{1+\sum_k r_{t,k}}{1-\sum_k r_{t,k}}\right). \tag{24}$$

In the classification phase, we estimate the label y_j of a test (sub)sequence $\boldsymbol{x}_j$ whose length is $L < T$ as follows:

$$y_j = \arg \max_{k=1,2,\ldots,K} H_k(\boldsymbol{x}_j) = \arg \max_{k=1,2,\ldots,K} \sum_{t=1}^{L} \alpha_t h_{t,k}(x_{j,t}). \tag{25}$$

The resulting strong classifier tries best to distinguish K classes given shorter L samples thanks to the early classification property brought by Eq. (23).

3.2 Short remarks

Our Earlyboost.MH has two advantages over the original Earlyboost. At first, Earlyboost.MH optimizes the loss Eq. (20), which equally evaluates all K classes. Thus the learnt model is optimal for K class classification problem. On the other hand, the original Earlyboost [13] trains a $\binom{K}{2}$ set of strong binary classifiers which discern between the class k and k'. These strong classifiers are optimized for each binary classification, thus the performance in K class classification will degrade.

Also, our Earlyboost.MH only requires training of KT weak classifiers for K class classification problems. It effectively reduces the number of weak classifiers compared to the original Earlyboost, which requires $T\binom{K}{2}$ weak classifiers. Some practical tasks such as Chinese character recognition and large-vocabulary speech recognition have large cardinality of classes ($K > 100, 1000, \ldots$). In such cases, this computational advantage of Earlyboost.MH may be essential to develop the system since learning, installing and using $\binom{K}{2}$ strong classifier becomes inhibitory.

In our model, we construct a time frame-wise weak classifier $h_{t,k}$, which only considers sample distribution at time t. We can naturally adapt the classifier to the dynamics of sequence by setting T weak classifiers for T length sequence as in Eq. (21). We can extend the weak classifiers to take care of multiple frames easily, but this does not change the description of the model.

4 Experiments

We conduct two experiments to validate the performance of Earlyboost.MH. One is the online handwritten digits recognition, and another is the behavior recognition of drivers (Fig. 2).

4.1 Datasets

The same dataset used in [13] is tested in online handwritten digits recognition task (Fig. 2(A)). The dataset is called Ethem Alpaydim Digit, which is a collection of digit ("0"-"9") writing

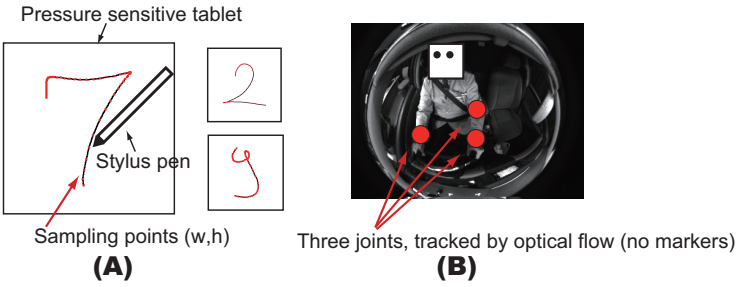

Figure 2: Dataset used in the experiments. (A) On-line handwritten digits recognition task is conducted using Ethem Alpaydim Digit dataset. Trajectories are collected through a pressure sensitive tablet and a wireless stylus pen. (B) For driver behavior recognition tasks, seven subjects are recorded their driving simulations by a consumer video camera. Three joints are tracked by optical flow, without any markers or special attachments.

trajectories. Following [13], we align the lengths of all sequences to $T = 50$ by simple linear interpolations. Observed features $x_{i,t}$ consists of i) 2D coordinates of a stylus pen tip calculated by a pressure sensitive tablet and ii) 2D velocity of the pen tip computed by frame difference. These $d = 4$ dimensional data are whitened in prepossessing.

This task is a $K = 10$ class classification problem. Each sequence corresponds to a writing trajectory of one digit. We have approximately $N = 11000$ sequences in total. We employed kNN classifiers ($k = 5$) for a weak learner, where the samples are resampled based on the weights $D_t(i,k)$. Learning and classification is conducted in 6 fold cross validation.

Our second task is driver behavior recognition (Fig. 2(B)). A video camera was installed in a real-scale driving simulator and recorded the driving behaviors of seven people. Each person drove 30 times. The movies were recorded in 60 FPS. We tracked several of the drivers' joints using optical flows, and finally obtained joint coordinate' sequences of the left wrist, right wrist and left elbow on the 2D images [11]. Thus, the observation data is a sequence of $d = 6(= 3 \times 2)$ dimensional vectors. Sequence lengths were not normalized.

The number of behavioral classes (patterns) is $K = 12$, which includes "manipulating A/C," "adjusting the rear-view mirror," and so forth. All time frames were manually labeled, and were segmented into a series of behavior motion sequences. Every sequence starts from the home position: the driver holds the steering wheels by both hands. This is a natural assumption because a driver holds the steering wheel in the most part of driving. The number of sequences N was different for each person, but roughly $N = 660$ sequences were collected for each person. We chose a decision stump [2] for a weak classifier. Classification precision was computed via 6-fold cross validation for each person.

4.2 Compared method

We compare the proposed method with the original Earlyboost [13]. We need to take care of multi-class situation because Earlyboost is a binary classifier. Instead of constructing $\binom{K}{2}$ "one vs. one (k vs. k')" classifiers as in [13], we took another approach for a fair comparison in terms of the computational cost.

We have implemented "one vs. all" type binary classifiers for K classes by Earlyboost. Each binary strong classifier H_k estimates whether the test sequence $\boldsymbol{x}_j$ "belongs to class k

($y = 1$)", or "belongs to the other classes ($y = -1$)". The difference is that K strong binary classifiers are learnt independently in our implementation of Earlyboost. For classification, we compute the sum of weak classifiers' output scores and classify the sequence to the class of **arg max** as in Eq. (25). We can expect that the comparison is fair since the number of weak classifiers and computational costs for learning and classifications become almost the same with Earlyboost.MH in this setup. Please remember that all classifiers for K classes are optimized simultaneously in the proposed Earlyboost.MH.

4.3 Results

First, we show the results of online handwritten digits recognition task. Fig. 3 presents the averaged classification precision score for $K = 10$ class. The vertical axis denotes the precision, and the horizontal axis the length L of the input test (sub)sequence (Eq. (25)). The thick blue line is the averaged precision of propose Earlyboost.MH, and the thin red line is that of Earlyboost.

Earlyboost.MH is superior to the previous Earlyboost in many frames. Especially, Earlyboost.MH marked relatively good scores around $40 \sim 45$ frames. However we also admit the proposed method is little inferior to Earlyboost in the first 15 frames, which is not pleasing for early classification. Yet we think that we can assume the proposed multi-class boosting approach is good, at least comparable, in early sequence classification of handwritten digits.

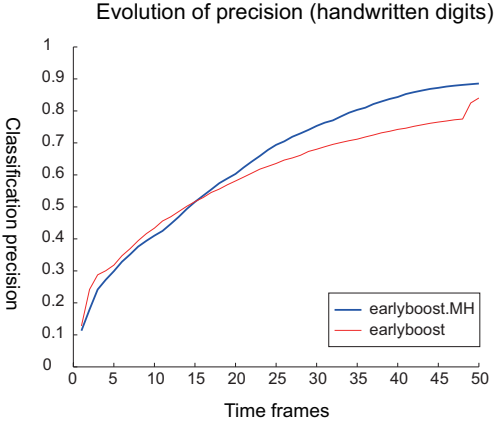

Figure 3: Time evolution of averaged classification precision in handwritten digits recognition task. The horizontal axis denotes the length of the input, the vertical axis denotes the precision. The thick red lines is the results of Earlyboost.MH, and the thin blue line is the results of Earlyboost.

Next, we show the temporal evolution of the classification precision score of the driver behavior recognition task in Fig. 4. In this figure, we present the average of seven drivers results. It is clear that the proposed Earlyboost.MH (denoted by the thick red line) again outperforms Earlyboost (the thin blue line) for the sequence early classification.

However, there is a notable gap in achieved improvements of classification precision between two tasks. We think this is due to the characteristics of the task (and the dataset). In the case of online handwritten digits recognition, we expect that the trajectories of each

digit class have "uniqueness" or independence between classes. For example, digits "1" and "5" are very different in not only the start position of writing, but also in the curvature of the trajectories. Therefore, we achieve relatively good scores by Earlyboost which does not incorporate the correlations between classes.

On the other hand, the data of driver behavior recognition is characterized by the fact that all sequences start from the "home position" (holding the steering wheel by both hands) because of expected applications. We also note that some behaviors such as "manipulate A/C" and "using car navigation system" will have very similar trajectories from the first frame to the last frame. These implies that all trajectories are indistinguishable in the early frames, and the distributions of the sequence samples are highly correlated between some classes. Therefore, if we neglect the dependency between classes like Earlyboost, we are not able to obtain a good classification score. We conclude that the proposed method is especially useful for the multi-class dataset where some classes are highly correlated.

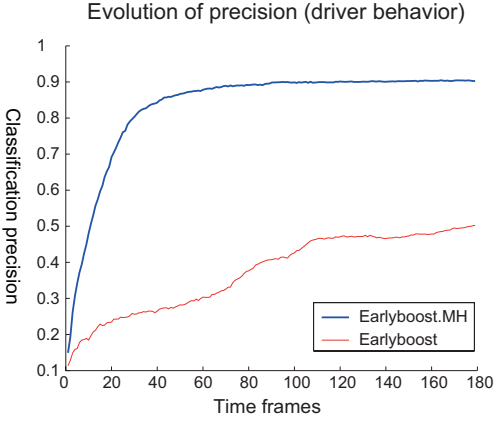

Figure 4: Time evolution of averaged classification precision in driver behavior recognition task. The horizontal axis denotes the length of the input, the vertical axis denotes the precision. The thick red lines is the results of Earlyboost.MH, and the thin blue line is the results of Earlyboost. Averaged results for seven drivers are presented.

5 Conclusion

We presented a new boosting algorithm for early classification of multi-class sequences. We incorporated the idea of weight propagation proposed by [13] into a standard multi-class Adaboost [10], and derived Earlyboost.MH model based on the standard exponential loss approach [5]. The complexity of the model grows only linearly proportional to the number of classes, and the resultant model showed comparable of much better results in experiments on multi-class classification of handwritten digits and driver behaviors.

We will study the capability and the limitation of the model for several types of computer vision tasks, which includes large class ($K > 100$) classification problem such as character recognition. Also studying other types of loss function for multi-class early classification [15] is highly interesting and beneficial for many applications.

References

[1] C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Trans. PAMI*, 26(3):299–310, Mar. 2004.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[3] T. G. Dietterich. Machine learning for sequential data: A review. In T. Caelli, editor, *Proc. SSSPR*, pages 15–30. Springer-Verlag., 2002.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computing Systems and Science*, 55(1): 119–139, 1997.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting (with discussion). *Annals of Statistics*, 28(2):337–407, 2000.

[6] C. J. A. Gonzalez, G. Juan, and J. J. R. Diez. Boosting interval-based literals: Variable length and early classification. In *Proc. ECAI'02 Workshop on Knowledge Discovery from (Spatio-) Temporal Data*, 2002.

[7] T. Hori, C. Hori, and Y. Minami. Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition. In *Proc. Interspeech*, volume 1, pages 289–292, 2004.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random field: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.

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

[10] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[11] Y. A. Sheikh, A. Datta, and T. Kanade. On the sustained tracking of human motion. In *Proc. FG*, 2008.

[12] J. Sochman and J. Matas. Waldboost — learning for time constrained sequential detection. In *Proc. CVPR*, volume 2, pages 150–156, 2005.

[13] S. Uchida and K. Amamoto. Early recognition of sequential patterns by classifier combination. In *Proc. ICPR*, 2008.

[14] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[15] H. Zou, J. Zhu, and T. Hastie. New multicategory boosting algorithms based on multicategory Fisher-consistent losses. *The Annals of Applied Statistics*, 2(4):1290–1306, 2008.