Journal of Machine Learning Research 15 (2014) 3013-3036 Submitted 2/14; Revised 5/14; Published 10/14

Bayesian Co-Boosting for Multi-modal Gesture Recognition

Jiaxiang Wu JIAXIANG.WU@NLPR.IA.AC.CN
Jian Cheng* JCHENG@NLPR.IA.AC.CN
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

Editor: Isabelle Guyon, Vassilis Athitsos, Sergio Escalera

Abstract

With the development of data acquisition equipment, more and more modalities become available for gesture recognition. However, there still exist two critical issues for multi-modal gesture recognition: how to select discriminative features for recognition and how to fuse features from different modalities. In this paper, we propose a novel Bayesian Co-Boosting framework for multi-modal gesture recognition. Inspired by boosting learning and co-training method, our proposed framework combines multiple collaboratively trained weak classifiers to construct the final strong classifier for the recognition task. During each iteration round, we randomly sample a number of feature subsets and estimate weak classifier's parameters for each subset. The optimal weak classifier and its corresponding feature subset are retained for strong classifier construction. Furthermore, we define an upper bound of training error and derive the update rule of instance's weight, which guarantees the error upper bound to be minimized through iterations. For demonstration, we present an implementation of our framework using hidden Markov models as weak classifiers. We perform extensive experiments using the ChaLearn MMGR and ChAirGest data sets, in which our approach achieves 97.63% and 96.53% accuracy respectively on each publicly available data set.

Keywords: gesture recognition, Bayesian co-boosting, hidden Markov model, multi-modal fusion, feature selection

1. Introduction

As one of the most natural and intuitive ways for human computer interaction, gesture recognition has been attracting more and more attention from academe and industry. With automatic gesture recognition techniques, one can use his/her hands to freely interact with computers. It has been widely applied to sign language recognition (Zafrulla et al., 2011; Oz and Leu, 2011), robot control (Raheja et al., 2010), games (Roccetti et al., 2011), etc. In the early days, accelerometer-based approaches were especially popular for gesture recognition, due to their simpleness and accuracy in data acquirement (Mantyla et al., 2000; Chambers et al., 2002; Pylvänäinen, 2005; Liu et al., 2009). As an extension to the accelerometer, the inertial measurement unit (IMU) can be adopted to collect more information, such as linear acceleration and angular acceleration. There are also several IMU-based gesture recognition methods proposed recently (Zhang et al., 2013; Yin and

*. Corresponding author.

Davis, 2013). Nevertheless, the requirement of wearing accelerometers or IMUs limits the applicability of the above approaches. Vision-based approaches, which do not need to wear any extra devices, offer an appealing approach to gesture recognition. However, vision-based approaches are vulnerable to illumination, self-occlusion, and variation of gesture. Moreover, visual feature representation is still an open problem.

As an alternative, depth-aware camera (e.g., Microsoft® Kinect™) can capture RGB image, depth image, and audio, which makes gesture recognition less sensitive to illumination changes, self-occlusion, and can offer strong information for background removal, object detection, and localization in 3D space. With the prevalence of depth-aware camera, the study of gesture recognition is extremely stimulated and multi-modal based approaches are becoming a hot topic. Recently, there are many research works to utilize multiple modalities acquired by depth-aware camera for gesture recognition (Wu et al., 2012; Lui, 2012a; Malgireddy et al., 2012; Bayer and Silbermann, 2013; Nandakumar et al., 2013; Chen and Koskela, 2013). Since 2011, ChaLearn has organized a series of competitions based on the multi-modal gesture data captured by Kinect™. The tasks include one-shot-learning of gestures (Guyon et al., 2012) and continuous gesture spotting and recognition (Escalera et al., 2013). Many of participants achieved satisfactory performances on gesture recognition. However, for multi-modal based approaches, there still exist two critical issues for gesture recognition: how to select discriminative features for recognition, and how to fuse features from different modalities.

In the context of dynamic gesture recognition, an instance is represented by a time series sequence. Most of existing feature extraction methods for time series are mainly based on the self-defined criterion functions to evaluate each feature dimension's contribution (Kashyap, 1978; Mörchen, 2003; Yoon et al., 2005). For face detection, Viola and Jones (2001, 2004) constructed a strong classifier by selecting a small number of important features using AdaBoost. Foo et al. (2004) and Zhang et al. (2005) employed boosting learning for the single-modal gesture recognition task. However, boosting learning could be prone to be overfitting in practice when training data is rather small. As a late fusion strategy, co-training alternately uses the most confident unlabeled data instance(s) in one modality to assist the model training of another modality, to overcome the problem of insufficient training samples (Blum and Mitchell, 1998). Furthermore, Yu et al. (2008, 2011) proposed a Bayesian undirected graphical model interpretation for co-training methods in the context of semi-supervised multi-view learning. These two publications clarified several fundamental assumptions underlying these models and can automatically estimate how much trust should be given to each view so as to accommodate noisy views.

Inspired by boosting and Bayesian co-training methods, we present a novel Bayesian Co-Boosting training framework to realize effectively the multi-modal fusion for gesture recognition task.[1] In our framework, weak classifiers are trained with weighted data instances through multiple iterations. In each iteration round, several feature subsets are randomly generated and weak classifiers are trained on different feature groups. Only the weak classifier, which achieves the minimal training error, together with the corresponding feature subset is retained. Instance's weight is updated according to the classification result given by the weak classifiers of two modalities, so that the difficult instances will

1. Our preliminary work of multi-modal fusion on ChaLearn MMGR challenge 2013 achieved the 1st prize on gesture recognition (Wu et al., 2013).

gain more focus in the subsequent iterations. The strong classifier is constructed with all retained weak classifiers, and the classification decision is determined by the voting result of all weak classifiers. The weak classifier's voting weight is related to its prediction error on the training set.

The main contributions of this paper are concluded as follows:

1. The proposed framework is illuminated in a Bayesian perspective, and its error upper bound is minimized through iterations, which is guaranteed in theory.
2. Feature selection and multi-modal fusion are naturally embedded into the training process of weak classifiers in each Co-Boosting iteration round and bring significant improvement to the recognition performance.
3. A novel parameter estimation method is presented to address the training problem of hidden Markov model on the weighted data set.

This paper is organized as follows. In Section 2, commonly used approaches for gesture recognition is reviewed. We describe our proposed approach and related theoretical derivation in Section 3. Section 4 presents the experimental result of our method, comparing with several state-of-the-art methods. Finally, we conclude our work in Section 5.

2. Related Work

Gesture recognition has been an important research topic in human computer interaction and computer vision field. There already exist a few published surveys in this area, such as Gavrila (1999), Mitra and Acharya (2007), Weinland et al. (2011), and Suarez and Murphy (2012). As concluded in these literatures, classifiers commonly used in gesture recognition include k-nearest neighbours (Malassiotis et al., 2002), hidden Markov model (Eickeler et al., 1998), finite state machine (Yeasin and Chaudhuri, 2000), neural network (Yang and Ahuja, 2001), and support vector machine (Biswas and Basu, 2011).

Gesture recognition based on accelerometers has been investigated by many researchers (Mantyla et al., 2000; Chambers et al., 2002; Pylvänäinen, 2005; Liu et al., 2009). As an extension to the accelerometer sensors, the applications of inertial measurement unit (IMU) have also been explored recently. Ruffieux et al. (2013) collected a benchmark data set with Kinect™ and XSens IMU sensors for the development and evaluation of multi-modal gesture spotting and recognition algorithms. With this data set, Yin and Davis (2013) presented a hand tracking method based on gesture salience, and concatenated hidden Markov models were applied to perform gesture spotting and recognition.

Considering the inconvenience of wearing accelerometers or IMUs while performing gestures, it is more natural to develop vision-based gesture recognition systems. Single or stereo camera is mostly widely used in research, but Kinect™ sensor has been attracting increasing interest, due to its ability to capture both color and depth images simultaneously. ChaLearn has organized several competitions focused on the Kinect™-based gesture recognition ever since 2011 (Guyon et al., 2012; Escalera et al., 2013).

Approaches based on hidden Markov model (HMM) are widely adopted in vision-based gesture recognition. Elmezain et al. (2008) applied HMM to recognize isolated and continuous gestures in real-time. Spatio-temporal trajectories were converted to orientation dynamic features and then quantized to one of the codewords. The quantized observation

sequence was then used to inference the hidden gesture label. Gaus et al. (2013) compared the recognition performance given by both fixed state HMM and variable state HMM. In Nandakumar et al. (2013), gesture instances in the continuous data stream were segmented using both audio and hand joint information. Three modalities were used for classification: HMM classifier for MFCC feature extracted from audio signal, and SVM (support vector machine) classifier for both RGB (STIP feature) and skeleton (covariance descriptor).[2] Wu et al. (2013) performed automatic gesture detection based on the endpoint detection result in the audio data stream. HMM classifiers were then applied to both audio and skeleton features, and a late fusion strategy was employed to make the final classification decision.

In order to enhance the recognition performance of HMM-based approaches, ensemble learning, especially AdaBoost, has been embedded into the training process of hidden Markov models in a few researches. Adaptive boosting (Freund and Schapire, 1995; F. and E.S., 1997) is a training framework to generate multiple weak classifiers with different training instances' weight distribution, and construct a strong classifier with these weak classifiers to achieve a better classification performance. Foo et al. (2004) proposed a novel AdaBoost-HMM classifier to boost the recognition of visual speech elements. Weak classifiers were trained using biased Baum-Welch algorithm under the AdaBoost framework to cover different groups of training instances. Their decisions on the unlabeled instance were combined following a novel probability synthesis rule to obtain the final decision. In Zhang et al. (2005), a similar approach was applied in the application of sign language recognition. However, both researches neglected the potential noisy dimensions in the feature space, which could cause the deterioration of recognition performance.

Besides HMM-based approaches, there are also many other methods proposed in the context of vision-based gesture recognition. In Lui et al. (2010) and Lui (2012b), action videos were factorized using higher order singular value decomposition (HOSVD) and the classification was performed based on the geodesic distance on the product manifold. Boyali and Kavakli (2012) proposed a variant version of sparse representation based classification (innovated by Wright et al., 2009; Wagner et al., 2009) for gesture recognition. For a more complete overview of commonly used approaches in gesture recognition, we recommend the survey papers mentioned at the beginning of this section.

3. Bayesian Co-Boosting with Hidden Markov Model

For multi-modal gesture recognition task, fusion of features from different modalities is one of the most vital problems. Many existing approaches use a simple weighted-based fusion strategy (Bayer and Silbermann, 2013; Nandakumar et al., 2013). However, this weight coefficient usually needs to be empirically tuned, which is rather difficult if not impossible on large-scale data set. As we mentioned before, Bayesian co-training (Yu et al., 2008, 2011) can automatically determine each view's confidence score, which inspired us to adopt a similar approach to fuse multiple modalities. Boosting learning can perform feature

2. MFCC: Mel-Frequency Cepstral Coefficients (Zheng et al., 2001), a common used audio feature for speech recognition. The feature extraction process is as follows: a) the signal segment is turned into frequency domain using Discrete Fourier Transform; b) the short-term power spectrum is warped into the Mel-frequency; c) the warped power spectrum is convolved with the triangular band-pass filter; d) the MFCC feature is the Discrete Cosine Transform result of the convolved power spectrum.

selection through training multiple weak classifiers, and can be used in gesture recognition to select optimal feature dimensions for the classification problem.

In this section, we introduce a novel Bayesian Co-Boosting training framework for combining multiple hidden Markov model classifiers for multi-modal gesture recognition. Based on the proposed Bayesian Co-Boosting framework, different modalities are naturally combined together and can provide complementary information for each other. We also analyze the minimization of the error upper bound so as to derive the update rule of instance's weight in Co-Boosting process.

3.1 Model Learning

In the task of multi-modal gesture recognition, two or more modalities (in this paper, we constraint the amount of modalities to be two) are simultaneously available for describing gesture instances. Based on the raw data of each modality, a time series sequence of feature vectors can be extracted according to certain feature extraction procedures. This time series sequence data is then used as the input to the pre-trained classifier for model training and evaluation.

The most straightforward approach to this problem is to separately train a classifier for each modality, and then combine their classification results in a late fusion style. However, this approach will bring the following issues. First, feature vectors may contain noisy data dimensions, which will lead to deterioration of classification performance. Second, one classifier for one modality may not be sufficient to achieve a satisfying classification accuracy level. Third, the fusion weights of different classifiers, which have significant impact on the final classification result, are difficult to be tuned manually.

In this paper, we propose an approach to solve all these problems together. Under the Co-Boosting framework, multiple weak classifiers of each modality are trained through a number of iterations. The final strong classifier is a linear combination of these weak classifiers, and each classifier's weight is determined by its prediction error on the training data set. Figure 1 depicts the work flow of our proposed method, and Algorithm 1 describes the detailed procedures in the model training process.

The aim of our proposed Bayesian Co-Boosting framework is to generate a strong classifier for the multi-modal gesture recognition task. As we can see in Figure 1, the resulting strong classifier $H(x_i)$ is the combination of multiple weak classifiers trained on V different modalities through T iterations. In each iteration round, M_v candidate weak classifiers are trained on the v-th modality using different feature dimension subsets, and the best candidate among them is selected as the optimal weak classifier $h^*_{t,v}(x_i)$. The optimal weak classifier is the one which achieves the minimal training error among all candidate weak classifiers for modality v. Then we use all these selected weak classifiers (one weak classifier per modality) obtained at this iteration round to update each training instance's weight.

In the rest of this section, we firstly introduce the training process of a single weak classifier with weighted instances. Secondly, we derive the update rule of the instance's weight to minimize the training error's upper bound from a Bayesian perspective. The construction of the strong classifier $H(x_i)$ is described at the end of this section.

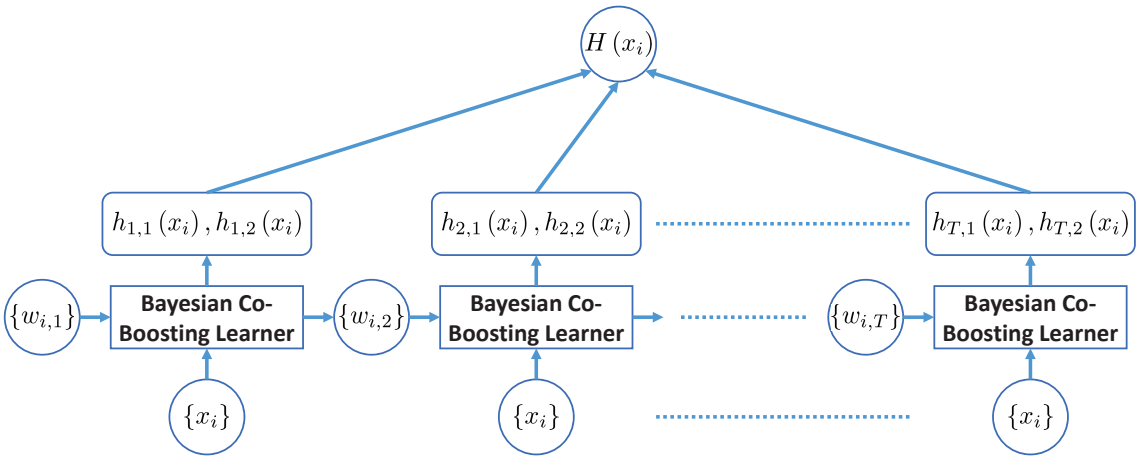

Figure 1: Work flow of Bayesian Co-Boosting training framework.
x_i: training instance; $w_{i,t}$: training instance x_i's weight at the t-th iteration; $h_{t,v}(x_i)$: weak classifier learnt from modality v at the t-th iteration; $H(x_i)$: final strong classifier.

Algorithm 1 Bayesian Co-Boosting Training Framework.[3]

Input: training instances $\{x_i\}$
Output: strong classifier $H(x_i)$

initialize data weight distribution $\{w_i\}$
for $t = 1, \ldots, T$ **do**
  for $v = 1, \ldots, V$ **do**
    for $m = 1, \ldots, M_v$ **do**
      randomly generate feature subset $\tilde{F}_{t,v,m} \subset F_v, |\tilde{F}_{t,v,m}| = \lambda_v \cdot |F_v|$
      generate training data set $\{(\tilde{x}_i, w_i)\}$ with feature dimensions in $\tilde{F}_{t,v,m}$
      train candidate weak classifier $h_{t,v,m}(x_i)$ (refer to Algorithm 2)
      calculate classifier's training error $\varepsilon_{t,v,m}$
    end for
    select optimal candidate weak classifier $h^*_{t,v}(x_i)$ and feature subset $\tilde{F}^*_{t,v}$
    calculate weak classifier's voting weight $\alpha^*_{t,v}$
  end for
  update instances' weights $\{w_i\}$ (refer to Algorithm 3)
end for
construct strong classifier $H(x_i)$

3. T: the number of Co-Boosting iteration rounds; V: the number of modalities; M_v: the number of candidate weak classifiers for modality v; F_v: all available feature dimensions for modality v; λ_v: the feature dimension selection ratio for modality v.

3.1.1 Weak Classifier Training

As we concluded in Section 2, hidden Markov model is one of the most commonly used classifiers in gesture recognition. Therefore, in this paper, we implement the Bayesian Co-Boosting training framework with HMM-based weak classifiers embedded. However, other weak classifiers can also be easily adopted in our framework.

Hidden Markov model is a statistical model based on Markov process, in which the generation of an observation sequence is modeled as the result of a series of unobserved state transitions (Rabiner, 1989). In order to deal with continuous observation vectors, a multi-variate Gaussian distribution is adopted to determine the observation probability of each observation-state pair. To simplify the subsequent analysis, we define the following symbols:

$x_{i,1:T_i}$: observation sequence of length T_i, composed of feature vectors $x_{i,t}$.

$z_{i,1:T_i}$: state transition sequence; $z_{i,t} \in \{1, \ldots, K\}$, K is the number of states.

$\mathcal{D}$: the training data set consists of N observation sequences $x_{i,1:T_i}$.

π_k: initial state probability, $\pi_k = P(z_{i,1} = k)$.

$A_{j,k}$: state transition probability, $A_{j,k} = P(z_{i,t+1} = k | z_{i,t} = j)$.

μ_k, Σ_k: mean vector and covariance matrix, $P(x_{i,t} | z_{i,t} = k) = \mathcal{N}(x_{i,t} | \mu_k, \Sigma_k)$.

For multiple-class classification problem in gesture recognition, a hidden Markov model is trained for each gesture class, with its parameters denoted as θ_c. The resulting classifier is denoted as

$$\hat{y}_i = \arg\max_c P(x_i | \theta_c),$$

where $x_i = x_{i,1:T_i}$ is the unlabeled gesture instance. $P(x_i|\theta_c)$ measures the probability for model θ_c generating observation sequence x_i and can be rewritten as

$$P(x_i | \theta_c) = \sum_{z_i} P(x_i, z_i | \theta_c),$$

where the full data probability $P(x_i, z_i|\theta_c)$ is given by

$$\begin{aligned} P(x_i, z_i | \theta_c) &= P(z_i | \theta_c) P(x_i | z_i, \theta_c) \\ &= \pi_{z_{i,1}} \prod_{t=1}^{T-1} A_{z_{i,t}, z_{i,t+1}} \prod_{t=1}^{T} \mathcal{N}\left(x_{i,t} | \mu_{z_{i,t}}, \Sigma_{z_{i,t}}\right). \end{aligned}$$

For the parameter estimation problem of HMM, commonly used Baum-Welch algorithm (a variation of EM algorithm) can only deal with unweighted training instances. In boosting learning, however, instances are assigned with different weights, which are adjusted at the end of each iteration round to guide the subsequent weak classifiers focus on more difficult instances. Hence, we need to extend the standard Baum-Welch algorithm (Murphy, 2012) to accommodate the weighted instances' training problem in our approach. Our proposed parameter estimation method is also based on the EM algorithm.

Given the weighted training data set $\{(x_i, w_i)\}$, parameter estimation problem is to find the optimal parameters that maximize the log likelihood of the observed data, which is defined as

$$\ell(\theta) = \sum_{i=1}^{N} w_i \log P(x_i|\theta) = \sum_{i=1}^{N} w_i \log \left[\sum_{z_i} P(x_i, z_i|\theta) \right].$$

But this is difficult to optimize, since the log cannot be pushed inside the sum. To get around this problem, we define the complete data log likelihood as

$$\ell_c(\theta) = \sum_{i=1}^{N} w_i \log P(x_i, z_i^*|\theta),$$

where z_i^* is the optimal state transition sequence, and is inferred with Viterbi algorithm.

Therefore, the expected complete data log likelihood for data set $\mathcal{D}$ is given by

$$Q(\theta, \theta_{\text{old}}) = \mathbb{E}\left[\ell_c(\theta) | \mathcal{D}, \theta_{\text{old}}\right], \tag{1}$$

and the optimal parameters are estimated by maximizing this.

On the basis of the definition of $P(x_i, z_i|\theta_c)$, Equation (1) can be rewritten as

$$\begin{aligned}
Q(\theta, \theta_{\text{old}}) &= \mathbb{E}\left[\sum_{i=1}^{N} w_i \log P(x_i, z_i^*|\theta)\right] \\
&= \sum_{i=1}^{N} w_i \mathbb{E}\left[\log \prod_{z_i} P(x_i, z_i|\theta)^{\mathbb{I}(z_i^* = z_i)}\right] \\
&= \sum_{i=1}^{N} w_i \sum_{z_i} \mathbb{E}\left[\mathbb{I}(z_i^* = z_i)\right] \log P(x_i, z_i|\theta) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} w_i P\left(z_{i,1}^* = k | x_i, \theta_{t-1}\right) \log \pi_k \\
&\quad + \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \sum_{t=1}^{T_i - 1} w_i P\left(z_{i,t}^* = j, z_{i,t+1}^* = k | x_i, \theta_{t-1}\right) \log A_{j,k} \\
&\quad + \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{t=1}^{T_i} w_i P\left(z_{i,t}^* = k | x_i, \theta_{t-1}\right) \log P(x_{i,t}|z_{i,t} = k).
\end{aligned}$$

In the E step of EM algorithm, we firstly compute two groups of probabilities with forward-backward algorithm, as describe in Murphy (2012)

$$\begin{aligned}
\gamma_{i,t}(k) &= P(z_{i,t} = k | x_i, \theta_{t-1}) \\
\xi_{i,t}(j,k) &= P(z_{i,t} = j, z_{i,t+1} = k | x_i, \theta_{t-1}),
\end{aligned} \tag{2}$$

where $\gamma_{i,t}(k)$ indicates the probability of the hidden state at time t being state k, and $\xi_{i,t}(j,k)$ represents the probability of the hidden state being state j at time t and state k

at time $(t+1)$. Based on these probabilities, we compute the following expectation items

$$\begin{aligned}
\mathbb{E}\left[N_k^1\right] &= \sum_{i=1}^{N} w_i \gamma_{i,1}(k) \\
\mathbb{E}\left[N_{j,k}\right] &= \sum_{i=1}^{N} \sum_{t=1}^{T_i-1} w_i \xi_{i,t}(j,k) \\
\mathbb{E}\left[N_k\right] &= \sum_{i=1}^{N} \sum_{t=1}^{T_i} w_i \gamma_{i,t}(k) \\
\mathbb{E}\left[\bar{x}_k\right] &= \sum_{i=1}^{N} \sum_{t=1}^{T_i} w_i \gamma_{i,t}(k)\, x_{i,t} \\
\mathbb{E}\left[\bar{x}_k \bar{x}_k^T\right] &= \sum_{i=1}^{N} \sum_{t=1}^{T_i} w_i \gamma_{i,t}(k)\, x_{i,t} x_{i,t}^T.
\end{aligned} \tag{3}$$

In the M step, parameters are updated so that $Q(\theta, \theta_{\text{old}})$ is maximized. Here, we only present the final update rule for each parameter, due to the limitation of space

$$\begin{aligned}
\hat{\pi}_k &= \frac{\mathbb{E}\left[N_k^1\right]}{\sum_{k'=1}^{K} \mathbb{E}\left[N_{k'}^1\right]} \\
\hat{A}_{j,k} &= \frac{\mathbb{E}\left[N_{j,k}\right]}{\sum_{k'=1}^{K} \mathbb{E}\left[N_{j,k'}\right]} \\
\hat{\mu}_k &= \frac{\mathbb{E}\left[\bar{x}_k\right]}{\mathbb{E}\left[N_k\right]} \\
\hat{\Sigma}_k &= \frac{\mathbb{E}\left[\bar{x}_k \bar{x}_k^T\right]}{\mathbb{E}\left[N_k\right]} - \hat{\mu}_k \hat{\mu}_k^T.
\end{aligned} \tag{4}$$

The training procedure of weak classifier is demonstrated in Algorithm 2.

3.1.2 Instance's Weight Updating

In this sub-section, we define the training error for instances in each class, together with its upper bound to simplify the error minimization formulation. Based on this formulation, we derive the update rule for instance's weight in our proposed framework.

In the t-th iteration round of Bayesian Co-Boosting training process, the training error for class c is denoted by $E_{t,c}$, and the corresponding error upper bound is denoted by $B_{t,c}$.

We define the random variable $z_i \in \{1, \ldots, C\}$ to represent the hidden label for observation x_i. The binary prediction value for each candidate class of the strong classifier is determined by

$$H_{t,c}(x_i) = sgn\left(P_{t,c,i} > \bar{P}_{t,c,i}\right) = \begin{cases} +1, & P_{t,c,i} > \bar{P}_{t,c,i} \\ -1, & P_{t,c,i} \leq \bar{P}_{t,c,i} \end{cases},$$

where

$$\begin{aligned}
P_{t,c,i} &= P\left(z_i = c | h_{1,1}(x_i), h_{1,2}(x_i), \ldots, h_{t,1}(x_i), h_{t,2}(x_i)\right) \\
\bar{P}_{t,c,i} &= P\left(z_i \neq c | h_{1,1}(x_i), h_{1,2}(x_i), \ldots, h_{t,1}(x_i), h_{t,2}(x_i)\right),
\end{aligned}$$

Algorithm 2 Weak Classifier Training

Input: weighted training instances $\{(x_i, w_i)\}$
Output: weak classifier $h(x_i)$

for $c = 1, \ldots, C$ **do**
  initialize model parameters θ_c
  for $t = 1, \ldots, T$ **do**
    initialize expectation items
    for $i = 1, \ldots, N$ **do**
      compute $\gamma_{i,t}(k), \xi_{i,t}(j,k)$ according to Equation (2)
      update expectation items according to Equation (3)
    end for
    compute $\theta_c = \left\{\hat{\pi}_k, \hat{A}_{j,k}, \hat{\mu}_k, \hat{\Sigma}_k\right\}$ according to Equation (4)
  end for
end for
construct weak classifier $h(x_i) = \arg\max_c P(x_i|\theta_c)$

and $h_{*,*}(x_i) \in \{1, \ldots, C\}$ represents the predicted class label of weak classifier.

The training error $E_{t,c}$ is defined as the sum of $0-1$ loss of classifier's binary predictions for the c-th class, which is

$$E_{t,c} = \sum_{i:y_i=c} \mathbb{1}(H_{t,c}(x_i) \neq 1) + \sum_{i:y_i \neq c} \mathbb{1}(H_{t,c}(x_i) = 1), \tag{5}$$

where function $\mathbb{1}(\cdot)$ equals to 1 when the inner expression is true; otherwise, its value is 0.

The error upper bound $B_{t,c}$ is given by

$$B_{t,c} = \sum_{i=1}^{N} \left(\frac{\bar{P}_{t,c,i}}{P_{t,c,i}}\right)^{sgn(y_i=c)} = \sum_{i:y_i=c} \frac{\bar{P}_{t,c,i}}{P_{t,c,i}} + \sum_{i:y_i \neq c} \frac{P_{t,c,i}}{\bar{P}_{t,c,i}}. \tag{6}$$

Theorem 1 *$E_{t,c} \leq B_{t,c}$ always holds with definitions in Equation (5) and (6).*

Proof For each training instance x_i, we consider its training error $E_{t,c,i}$ and the corresponding upper bound $B_{t,c,i}$. It surely falls into one of the following conditions:

(1) $H_{t,c}(x_i) = 1, y_i = c$:
Based on the definition of $H_{t,c}(x_i)$, we have $P_{t,i,c} > \bar{P}_{t,i,c}$.
Since $E_{t,c,i} = 0, B_{t,c,i} = \bar{P}_{t,i,c}/P_{t,i,c} \in [0, 1)$, thus $E_{t,c,i} \leq B_{t,c,i}$.

(2) $H_{t,c}(x_i) = 1, y_i \neq c$:
Based on the definition of $H_{t,c}(x_i)$, we have $P_{t,i,c} > \bar{P}_{t,i,c}$.
Since $E_{t,c,i} = 1, B_{t,c,i} = P_{t,i,c}/\bar{P}_{t,i,c} \in [1, +\infty)$, thus $E_{t,c,i} \leq B_{t,c,i}$.

(3) $H_{t,c}(x_i) \neq 1, y_i = c$:
Based on the definition of $H_{t,c}(x_i)$, we have $P_{t,i,c} \leq \bar{P}_{t,i,c}$.
Since $E_{t,c,i} = 1, B_{t,c,i} = \bar{P}_{t,i,c}/P_{t,i,c} \in [1, +\infty)$, thus $E_{t,c,i} \leq B_{t,c,i}$.

(4) $H_{t,c}(x_i) \neq 1, y_i \neq c$:
Based on the definition of $H_{t,c}(x_i)$, we have $P_{t,i,c} \leq \bar{P}_{t,i,c}$.
Since $E_{t,c,i} = 0, B_{t,c,i} = P_{t,i,c}/\bar{P}_{t,i,c} \in [0,1)$, thus $E_{t,c,i} \leq B_{t,c,i}$.

Therefore, $E_{t,c,i} \leq B_{t,c,i}$ holds for every instance x_i; hence, $E_{t,c} \leq B_{t,c}$ is proved. ■

In the Co-Boosting training process, the weight of each training instance should reflect the difficulty for current weak classifiers to correctly classify it. Hence, instance's weight can be determined by

$$w_i = \frac{\bar{P}_{t,y_i,i}}{P_{t,y_i,i}}. \tag{7}$$

Now we derive the update rule of training instance's weight so as to minimize the error upper bound $B_{t,c}$ through iterations, from a Bayesian perspective.

Based on the definition of $P_{t,c,i}$, we have

$$\begin{aligned}
P_{t,c,i} &= P\left(z_i = c | h_{1,1}, h_{1,2}, \ldots, h_{t,1}, h_{t,2}\right) \\
&= \frac{P\left(z_i = c, h_{1,1}, h_{1,2}, \ldots, h_{t,1}, h_{t,2}\right)}{P\left(h_{1,1}, h_{1,2}, \ldots, h_{t,1}, h_{t,2}\right)} \\
&= \frac{P\left(z_i = c, h_{1,1}, h_{1,2}, \ldots, h_{t-1,1}, h_{t-1,2}\right)}{P\left(h_{1,1}, h_{1,2}, \ldots, h_{t-1,1}, h_{t-1,2}\right)} \frac{P\left(h_{t,1} | z_i = c\right) P\left(h_{t,2} | z_i = c\right)}{P\left(h_{t,1}, h_{t,2} | h_{1,1}, h_{1,2}, \ldots, h_{t-1,1}, h_{t-1,2}\right)} \\
&= P_{t-1,c,i} \cdot \frac{P\left(h_{t,1} | z_i = c\right) P\left(h_{t,2} | z_i = c\right)}{P\left(h_{t,1}, h_{t,2} | h_{1,1}, h_{1,2}, \ldots, h_{t-1,1}, h_{t-1,2}\right)},
\end{aligned}$$

in which $h_{*,*} = h_{*,*}(x_i)$ is the predicted class label given by the weak classifier.

Similarly, we can derive the update equation for $\bar{P}_{t,c,i}$

$$\bar{P}_{t,c,i} = \bar{P}_{t-1,c,i} \cdot \frac{P\left(h_{t,1} | z_i \neq c\right) P\left(h_{t,2} | z_i \neq c\right)}{P\left(h_{t,1}, h_{t,2} | h_{1,1}, h_{1,2}, \ldots, h_{t-1,1}, h_{t-1,2}\right)}.$$

Therefore, the ratio between $\bar{P}_{t,c,i}$ and $P_{t,c,i}$ can be rewritten as

$$\frac{\bar{P}_{t,c,i}}{P_{t,c,i}} = \frac{\bar{P}_{t-1,c,i} \cdot P\left(h_{t,1} | z_i \neq c\right) P\left(h_{t,2} | z_i \neq c\right)}{P_{t-1,c,i} \cdot P\left(h_{t,1} | z_i = c\right) P\left(h_{t,2} | z_i = c\right)}. \tag{8}$$

In order to simplify the following theoretical derivation, we define these symbols

$$\begin{aligned}
&P_{c,1} = P\left(h_{t,1} = c | z_i = c\right), P_{c,2} = P\left(h_{t,1} = c | z_i \neq c\right) \\
&P_{c,3} = P\left(h_{t,1} \neq c | z_i = c\right), P_{c,4} = P\left(h_{t,1} \neq c | z_i \neq c\right) \\
&Q_{c,1} = P\left(h_{t,2} = c | z_i = c\right), Q_{c,2} = P\left(h_{t,2} = c | z_i \neq c\right) \\
&Q_{c,3} = P\left(h_{t,2} \neq c | z_i = c\right), Q_{c,4} = P\left(h_{t,2} \neq c | z_i \neq c\right).
\end{aligned} \tag{9}$$

For each instance x_i, considering whether its ground-truth label y_i and predicted label $h_{t,1}, h_{t,2}$ is equal to c or not, we can assign it into one of the following subsets

$$\begin{aligned}
&\mathcal{D}_1 = \{x_i | h_{t,1} = c, h_{t,2} = c, y_i = c\}, \mathcal{D}_2 = \{x_i | h_{t,1} = c, h_{t,2} = c, y_i \neq c\} \\
&\mathcal{D}_3 = \{x_i | h_{t,1} = c, h_{t,2} \neq c, y_i = c\}, \mathcal{D}_4 = \{x_i | h_{t,1} = c, h_{t,2} \neq c, y_i \neq c\} \\
&\mathcal{D}_5 = \{x_i | h_{t,1} \neq c, h_{t,2} = c, y_i = c\}, \mathcal{D}_6 = \{x_i | h_{t,1} \neq c, h_{t,2} = c, y_i \neq c\} \\
&\mathcal{D}_7 = \{x_i | h_{t,1} \neq c, h_{t,2} \neq c, y_i = c\}, \mathcal{D}_8 = \{x_i | h_{t,1} \neq c, h_{t,2} \neq c, y_i \neq c\}.
\end{aligned} \tag{10}$$

On the basis of the above data partitioning, $B_{t,c}$ can be expanded as

$$\begin{aligned}
B_{t,c} &= \sum_{i:y_i=c} \frac{\bar{P}_{t,c,i}}{P_{t,c,i}} + \sum_{i:y_i\neq c} \frac{P_{t,c,i}}{\bar{P}_{t,c,i}} \\
&= \sum_{i:x_i\in\mathcal{D}_1} \frac{\bar{P}_{t-1,c,i}P_{c,2}Q_{c,2}}{P_{t-1,c,i}P_{c,1}Q_{c,1}} + \sum_{i:x_i\in\mathcal{D}_2} \frac{P_{t-1,c,i}P_{c,1}Q_{c,1}}{\bar{P}_{t-1,c,i}P_{c,2}Q_{c,2}} \\
&\quad + \sum_{i:x_i\in\mathcal{D}_3} \frac{\bar{P}_{t-1,c,i}P_{c,2}Q_{c,4}}{P_{t-1,c,i}P_{c,1}Q_{c,3}} + \sum_{i:x_i\in\mathcal{D}_4} \frac{P_{t-1,c,i}P_{c,1}Q_{c,3}}{\bar{P}_{t-1,c,i}P_{c,2}Q_{c,4}} \\
&\quad + \sum_{i:x_i\in\mathcal{D}_5} \frac{\bar{P}_{t-1,c,i}P_{c,4}Q_{c,2}}{P_{t-1,c,i}P_{c,3}Q_{c,1}} + \sum_{i:x_i\in\mathcal{D}_6} \frac{P_{t-1,c,i}P_{c,3}Q_{c,1}}{\bar{P}_{t-1,c,i}P_{c,4}Q_{c,2}} \\
&\quad + \sum_{i:x_i\in\mathcal{D}_7} \frac{\bar{P}_{t-1,c,i}P_{c,4}Q_{c,4}}{P_{t-1,c,i}P_{c,3}Q_{c,3}} + \sum_{i:x_i\in\mathcal{D}_8} \frac{P_{t-1,c,i}P_{c,3}Q_{c,3}}{\bar{P}_{t-1,c,i}P_{c,4}Q_{c,4}}.
\end{aligned}$$

To simplify the expression, we define

$$\alpha_1 = \frac{P_{c,1}}{P_{c,2}}, \alpha_2 = \frac{P_{c,3}}{P_{c,4}}, \alpha_3 = \frac{Q_{c,1}}{Q_{c,2}}, \alpha_4 = \frac{Q_{c,3}}{Q_{c,4}}, \tag{11}$$

$$\begin{aligned}
&S_1 = \sum_{i:x_i\in\mathcal{D}_1} \frac{\bar{P}_{t-1,c,i}}{P_{t-1,c,i}},\ S_2 = \sum_{i:x_i\in\mathcal{D}_2} \frac{P_{t-1,c,i}}{\bar{P}_{t-1,c,i}},\ S_3 = \sum_{i:x_i\in\mathcal{D}_3} \frac{\bar{P}_{t-1,c,i}}{P_{t-1,c,i}},\ S_4 = \sum_{i:x_i\in\mathcal{D}_4} \frac{P_{t-1,c,i}}{\bar{P}_{t-1,c,i}} \\
&S_5 = \sum_{i:x_i\in\mathcal{D}_5} \frac{\bar{P}_{t-1,c,i}}{P_{t-1,c,i}},\ S_6 = \sum_{i:x_i\in\mathcal{D}_6} \frac{P_{t-1,c,i}}{\bar{P}_{t-1,c,i}},\ S_7 = \sum_{i:x_i\in\mathcal{D}_7} \frac{\bar{P}_{t-1,c,i}}{P_{t-1,c,i}},\ S_8 = \sum_{i:x_i\in\mathcal{D}_8} \frac{P_{t-1,c,i}}{\bar{P}_{t-1,c,i}}.
\end{aligned} \tag{12}$$

where $\alpha_k, k = 1,\dots 4$ are unknown variables and $S_k, k = 1,\dots,8$ can be computed with weak classifier's prediction. Then we rewrite $B_{t,c}$ as

$$\begin{aligned}
B_{t,c} &= \frac{S_1}{\alpha_1\alpha_3} + S_2\cdot\alpha_1\alpha_3 + \frac{S_3}{\alpha_1\alpha_4} + S_4\cdot\alpha_1\alpha_4 \\
&\quad + \frac{S_5}{\alpha_2\alpha_3} + S_6\cdot\alpha_2\alpha_3 + \frac{S_7}{\alpha_2\alpha_4} + S_8\cdot\alpha_2\alpha_4.
\end{aligned}$$

The partial derivatives of $B_{t,c}$ for the unknown variables $\alpha_{1:4}$ are

$$\begin{aligned}
\frac{\partial B_{t,c}}{\partial\alpha_1} &= -\frac{S_1}{\alpha_1^2\alpha_3} + S_2\cdot\alpha_3 - \frac{S_3}{\alpha_1^2\alpha_4} + S_4\cdot\alpha_4 \\
\frac{\partial B_{t,c}}{\partial\alpha_2} &= -\frac{S_5}{\alpha_2^2\alpha_3} + S_6\cdot\alpha_3 - \frac{S_7}{\alpha_2^2\alpha_4} + S_8\cdot\alpha_4 \\
\frac{\partial B_{t,c}}{\partial\alpha_3} &= -\frac{S_1}{\alpha_1\alpha_3^2} + S_2\cdot\alpha_1 - \frac{S_5}{\alpha_2\alpha_3^2} + S_6\cdot\alpha_2 \\
\frac{\partial B_{t,c}}{\partial\alpha_4} &= -\frac{S_3}{\alpha_1\alpha_4^2} + S_4\cdot\alpha_1 - \frac{S_7}{\alpha_2\alpha_4^2} + S_8\cdot\alpha_2.
\end{aligned} \tag{13}$$

The optimal values of α_k should ensure that all partial derivatives in Equation (13) are equal to 0. Therefore, we obtain the following equations

$$\begin{aligned}\alpha_1 &= \sqrt{\frac{S_1/\alpha_3 + S_3/\alpha_4}{S_2 \cdot \alpha_3 + S_4 \cdot \alpha_4}}, \alpha_2 = \sqrt{\frac{S_5/\alpha_3 + S_7/\alpha_4}{S_6 \cdot \alpha_3 + S_8 \cdot \alpha_4}} \\ \alpha_3 &= \sqrt{\frac{S_1/\alpha_1 + S_5/\alpha_2}{S_2 \cdot \alpha_1 + S_6 \cdot \alpha_2}}, \alpha_4 = \sqrt{\frac{S_3/\alpha_1 + S_7/\alpha_2}{S_4 \cdot \alpha_1 + S_8 \cdot \alpha_2}},\end{aligned} \tag{14}$$

and α_k can be solved within a few iterations (less than 10 rounds for most conditions, according to our experimental results).

Based on the definitions in Equation (9), it is obvious that

$$\begin{aligned}P_{c,1} + P_{c,3} = 1, P_{c,2} + P_{c,4} = 1 \\ Q_{c,1} + Q_{c,3} = 1, Q_{c,2} + Q_{c,4} = 1,\end{aligned} \tag{15}$$

and these eight variables can be solved after all α_k are obtained.

Based on the above analysis for training error minimization, the detailed algorithm for multiple weak classifiers training is concluded in Algorithm 3.

Algorithm 3 Instance's Weight Updating

Input: training instances $\{x_i\}$
Input: instances' weight $\{w_{i,t-1}\}$
Input: weak classifiers $h_{t,1}(x_i), h_{t,2}(x_i)$
Output: updated instances' weight $\{w_{i,t}\}$

```
for c = 1, ..., C do
    assign instances into D_k according to Equation (10)
    compute S_k according to Equation (12)
    compute α_k according to Equation (14)
    compute P_{c,k}, Q_{c,k} according to Equation (11) and (15)
    for instance x_i in the c-th class do
        compute P_{t,c,i}, P̄_{t,c,i} according to Equation (8)
        compute w_{i,t} according to Equation (7)
    end for
end for
```

3.2 Class Label Inference

In our multi-modal gesture recognition system, the predicted class label of unclassified instance is determined by the voting result of all weak classifiers.

For the optimal weak classifier $h^*_{t,v}(x_i)$ with training error $\varepsilon^*_{t,v}$, the classifier weight is defined as

$$\alpha^*_{t,v} = \log \frac{1 - \varepsilon^*_{t,v}}{\varepsilon^*_{t,v}},$$

where the training error is calculated by

$$\varepsilon_{t,v}^{*}=\sum_{c=1}^{C}\sum_{i:y_i=c}w_i\cdot\mathbb{1}\left\{h_{t,v}^{*}\left(x_i\right)\neq c\right\}.$$

The final prediction of instance's class label is determined by

$$H\left(x_i\right)=\arg\max_{c}\sum_{t=1}^{T}\sum_{v=1}^{2}\alpha_{t,v}^{*}\mathbb{1}\{h_{t,v}^{*}\left(x_i\right)=c\}.$$

4. Experimental Results

In this section, experiments are carried out on two multi-modal gesture recognition data sets, to prove the effectiveness of our proposed Bayesian Co-Boosting training framework. On the basis of comparative results of different training algorithms, the main contributing elements to our improvement on classification accuracy are also analyzed.

4.1 Baseline Methods Description

The training framework we propose in this paper is a general model, and some state-of-the-art methods can be considered as the special cases of our framework. The key parameters controlling the complexity of training process are T (number of iterations), V (number of modalities), and M_v (number of feature subset candidates). Various approaches can be obtained with different combinations of these three parameters.

If we set $T=1$, then model is trained without boosting learning. Many approaches using a single HMM to model instances from one gesture class can be categorized into this case.

If we set $V=1$, then the classifier is actually trained with only one feature modality. During iterations, feature selection procedure remains unchanged, but the update rule of instance's weight no longer applies. In this case, an instance's weight can be updated in a similar way as described in Viola and Jones (2004).

If we set $M_v=1$ for each modality, the feature selection procedure is removed from training process. In this case, there is no need to generate feature subset, since it may cause unnecessary information loss. All feature dimensions are used during training.

Now we define 7 baseline approaches listed as follows, each of which is a special case of our framework. Through this comparison, we can discover which part of the framework is really contributing to the improvement in classification accuracy.

(1) M1: training a classifier with the 1st modality:
Parameters setup: $T=1, V=1, M_1=1$.
Classifier: $H\left(x_i\right)=\arg\max_c P\left(x_i|\theta_{1,c}\right)$.
x_i is the unlabeled instance, and $\theta_{1,c}$ are the parameters of hidden Markov model for instances in the c-th class, trained on the 1st modality.

(2) M2: training a classifier with the 2nd modality:
Parameters setup: $T=1, V=1, M_2=1$.
Classifier: $H\left(x_i\right)=\arg\max_c P\left(x_i|\theta_{2,c}\right)$.

x_i is the unlabeled instance, and $\theta_{2,c}$ are the parameters of hidden Markov model for instances in the c-th class, trained on the 2nd modality.

(3) M1+M2: training classifiers with the 1st and 2nd modality:
Parameters setup: $T = 1, V = 2, M_1 = M_2 = 1$.
Classifier: $H(x_i) = \arg\max_c [\alpha P(x_i|\theta_{1,c}) + (1-\alpha) P(x_i|\theta_{2,c})]$.
x_i is the unlabeled instance, and $\theta_{1,c}$ and $\theta_{2,c}$ are respectively the parameters of hidden Markov model for instances in the c-th class, trained on the 1st and 2nd modality.

(4) Boost.M1: training boosted classifiers with the 1st modality:
Parameters setup: $T > 1, V = 1, M_1 = 1$.
Classifier: $H(x_i) = \arg\max_c \sum_{t=1}^{T} \alpha_{t,1} \mathbb{1}\{h_{t,1}(x_i) = c\}$.
$h_{t,1}(x_i) = \arg\max_c P(x_i|\theta_{t,1,c})$ is the weak classifier learnt at the t-th boosting iteration, and $\alpha_{t,1}$ is the corresponding classifier's weight.

(5) Boost.M2: training boosted classifiers with the 2nd modality:
Parameters setup: $T > 1, V = 1, M_2 = 1$.
Classifier: $H(x_i) = \arg\max_c \sum_{t=1}^{T} \alpha_{t,2} \mathbb{1}\{h_{t,2}(x_i) = c\}$.
$h_{t,2}(x_i) = \arg\max_c P(x_i|\theta_{t,2,c})$ is the weak classifier learnt at the t-th boosting iteration, and $\alpha_{t,2}$ is the corresponding classifier's weight.

(6) Boost.Sel.M1: training boosted classifiers with selected features of the 1st modality:
Parameters setup: $T > 1, V = 1, M_1 > 1$.
Classifier: $H(x_i) = \arg\max_c \sum_{t=1}^{T} \alpha_{t,1} \mathbb{1}\{h_{t,1}(x_i) = c\}$.
$h_{t,1}(x_i) = \arg\max_c P(x_i|\theta_{t,1,c})$ is the weak classifier learnt at the t-th boosting iteration, and $\alpha_{t,1}$ is the corresponding classifier's weight. Unlike "Boost.M1", feature selection is performed in the training process of weak classifier $h_{t,1}(x_i)$.

(7) Boost.Sel.M2: training boosted classifiers with selected features of the 2nd modality:
Parameters setup: $T > 1, V = 1, M_2 > 1$.
Classifier: $H(x_i) = \arg\max_c \sum_{t=1}^{T} \alpha_{t,2} \mathbb{1}\{h_{t,2}(x_i) = c\}$.
$h_{t,2}(x_i) = \arg\max_c P(x_i|\theta_{t,2,c})$ is the weak classifier learnt at the t-th boosting iteration, and $\alpha_{t,2}$ is the corresponding classifier's weight. Unlike "Boost.M2", feature selection is performed in the training process of weak classifier $h_{t,2}(x_i)$.

For convenience, we denote our proposed approach as "BayCoBoost". Its corresponding parameter setup is $T > 1, V = 2, M_1 > 1, M_2 > 1$.

"M1" and "M2" are two naive methods for single-modal gesture recognition, and many HMM-based recognizers can be categorized into one of these. "M1+M2" is the late fusion result of "M1" and "M2". Considering the weight coefficient α, we evaluate 11 candidate values from 0 to 1 with equal step length on the training set using cross validation, and select the optimal α which reaches the minimal error. The approach used in Wu et al. (2013) can be regarded as a variation of the "M1+M2" method.

In "Boost.M1" and "Boost.M2", boosting learning is applied to enhance the recognition performance. Multiple HMM-based weak classifiers are trained through iterations. Foo et al. (2004); Zhang et al. (2005) respectively used this type of approach for the recognition of visual speech element and sign language. "Boost.Sel.M1" and "Boost.Sel.M2" are similar to them, but feature selection is embedded into the training process of each weak classifier.

Finally, our proposed method "BayCoBoost" integrates both modalities under the Bayesian Co-Boosting framework.

4.2 Experiment 1: ChaLearn MMGR data set

In 2013, ChaLearn organized a challenge on multi-modal gesture recognition with motion data captured by the Kinect™ sensor. This challenge provides a benchmark data set on the topic of multi-modal gesture recognition. Detailed information about this data set can be found in Escalera et al. (2013).

This data set contains 20 gesture categories, each of which is an Italian cultural or anthropological sign. Gestures in the data set are performed with one or two hands by 27 users, along with the corresponding word/phase spoken out. Data modalities provided in this data set include color image, depth image, skeletal model, user mask, and audio data.

The data set has been divided into three subsets already, namely *Development*, *Validation*, and *Evaluation*. In our experiment, *Development* and *Validation* subsets are used respectively for model training and testing. Based on the labeled data, we can segment out $7,205$ gesture instances from *Development* subset and $3,280$ instances from *Validation*. These two numbers are slightly smaller than the amount ($7,754$ and $3,362$) announced in Escalera et al. (2013), since we filter out those gesture instances which contain invalid skeleton data (when Kinect™ fails to track the skeleton and outputs all-zero skeleton data).

Among all feature modalities offered in this data set, we choose audio and skeleton feature to perform our proposed Bayesian Co-Boosting training process. We extract 39-dimension MFCC feature (Martin et al., 2001) from audio data stream and denote it as the first feature modality. The second modality is the 138-dimension skeleton feature extracted from 3D coordinates of 20 tracked joint points. The detailed extraction process of skeleton feature is described in the appendix.

In this experiment, parameters in Algorithm 1 are chosen as follows: $T = 20, V = 2, M_1 = 5$, and $M_2 = 10$. For MFCC feature, the size of feature subset is set to be 50% of all feature dimensions. The skeleton feature subset consists of 15% dimensions from the original feature space. Therefore, the number of feature dimensions used to train weak classifiers is respectively 20 for audio and 21 for skeleton. The number of iterations to estimate parameters of hidden Markov models for weak classifiers is set to 20. All these parameters are selected roughly using a grid search based on the cross validation result on the training subset.

We report the recognition accuracy of each gesture category in Figure 2. Also, several statistics are computed to provide a quantitative comparison between different methods' average recognition performance across all categories, which are reported in Table 1. The recognition accuracy is defined as the ratio of the number of correctly classified gestures against the number of all existing gestures in each class.

4.3 Experiment 2: ChAirGest data set

In Ruffieux et al. (2013), a multi-modal data set was collected to provide a benchmark for the development and evaluation of gesture recognition methods. This data set is captured with a Kinect™ sensor and four Xsens inertial motion units. Three data streams are provided by the Kinect™ sensor: color image, depth image, and 3D positions of upper-body joint points.

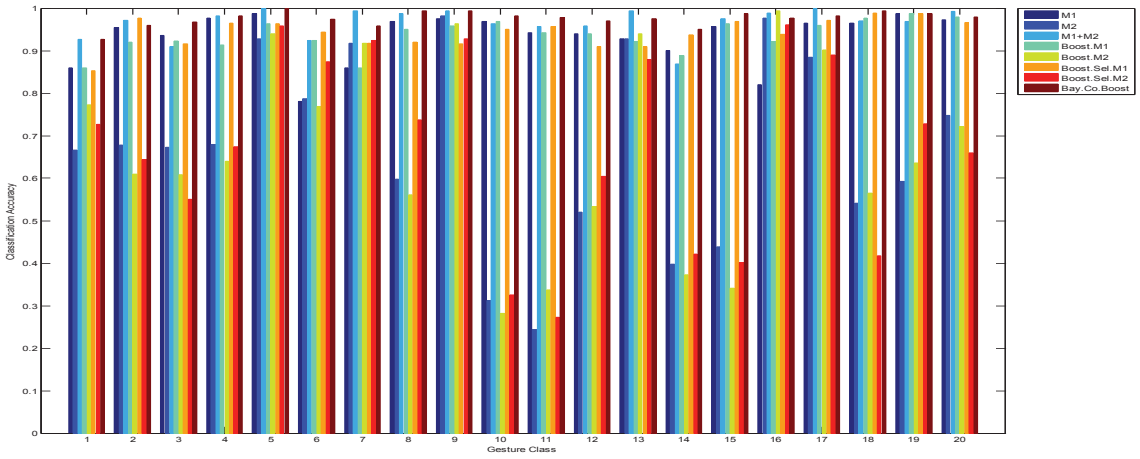

Figure 2: Recognition accuracy of each gesture category on ChaLearn MMGR data sets.

Method	Mean	Std	Conf	[Mean-Conf, Mean+Conf]
M1	0.9326	0.0584	0.0273	[0.9052, 0.9599]
M2	0.6749	0.2223	0.1040	[0.5709, 0.7790]
M1+M2	0.9666	0.0345	0.0162	[0.9504, 0.9827]
Boost.M1	0.9364	0.0366	0.0171	[0.9192, 0.9535]
Boost.M2	0.6705	0.2276	0.1065	[0.5640, 0.7770]
Boost.Sel.M1	0.9432	0.0334	0.0156	[0.9275, 0.9588]
Boost.Sel.M2	0.6793	0.2219	0.1038	[0.5754, 0.7831]
BayCoBoost	**0.9763**	**0.0173**	**0.0081**	**[0.9682, 0.9844]**

Table 1: Recognition accuracy on ChaLearn MMGR data sets.

Each Xsens IMU sensor can provide linear acceleration, angular acceleration, magnetometer, Euler orientation, orientation quaternion, and barometer data with a frequency of 50Hz.

This data set contains a vocabulary of 10 one-hand gestures commonly used in close human-computer interaction. Gestures are performed by 10 subjects, and each gesture is repeated 12 times, including 2 lighting conditions and 3 resting postures. The total number of gesture instances is 1200.

Similar to the previous experiment, two feature modalities are chosen to perform our Bayesian Co-Boosting training process. The first feature modality is based on the data captured by Xsens sensors. We use the raw data collected by four Xsens sensors as feature vector, which is of 68-dimension. Skeleton data captured by the Kinect™ is used as the second modality, and a 120-dimension feature vector is extracted per frame (see the appendix for details). The number of skeleton feature dimensions is smaller than the previous one, because the position of two joint points (hip-center and spine) cannot be tracked since all users were performing gestures while sitting.

The parameters in this experiment are almost identical with previous experiment. In Algorithm 1, parameters are: $T = 20, V = 2$, and $M_1 = M_2 = 10$. The feature selection ratio of Xsens and skeleton are respectively 20% and 15%. Under this setup, the feature

dimension of Xsens data for weak classifier training is 14, and this number is 18 for skeleton feature. The number of iterations for weak classifier training is also set to 20. Similar to the previous experiment, these parameters are also determined by cross-validation.

Since no division of training and testing subset is specified in this data set, we perform leave-one-out cross validation. In each round, gesture instances of one subject are used for model evaluation, and other instances are used to train the model. We compute the average recognition accuracy for each gesture class and report them in Figure 3 and Table 2.

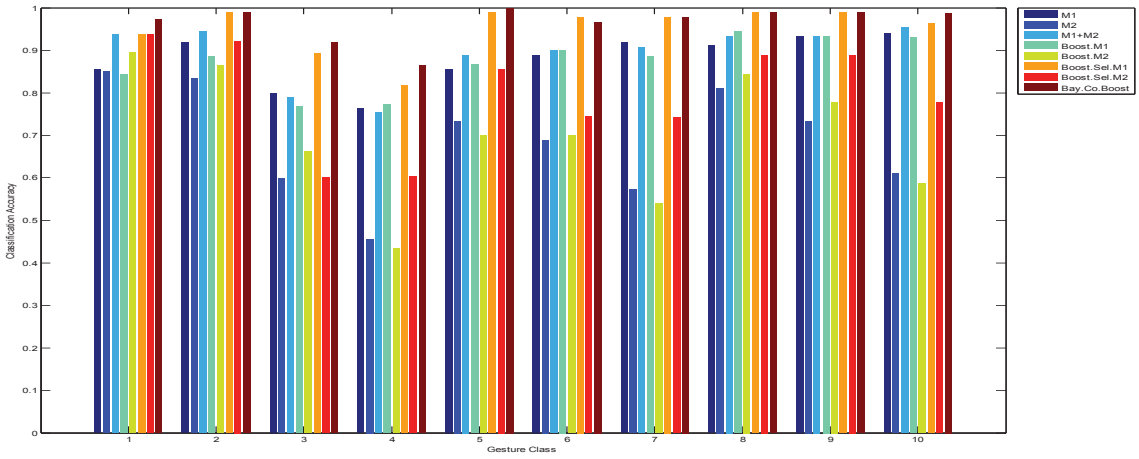

Figure 3: Recognition accuracy of each gesture category on ChAirGest data sets.

Method	Mean	Std	Conf	[Mean-Conf, Mean+Conf]
M1	0.8782	0.0598	0.0427	[0.8355, 0.9210]
M2	0.6884	0.1283	0.0918	[0.5966, 0.7801]
M1+M2	0.8940	0.0685	0.0490	[0.8450, 0.9430]
Boost.M1	0.8728	0.0623	0.0445	[0.8283, 0.9174]
Boost.M2	0.7003	0.1501	0.1074	[0.5929, 0.8077]
Boost.Sel.M1	0.9522	0.0564	0.0403	[0.9119, 0.9925]
Boost.Sel.M2	0.7958	0.1242	0.0889	[0.7070, 0.8847]
BayCoBoost	**0.9653**	**0.0420**	**0.0300**	**[0.9353, 0.9953]**

Table 2: Recognition accuracy on ChAirGest data sets.

4.4 Result Analysis

From the above experimental results, it is obvious that our proposed Bayesian Co-Boosting training algorithm achieves the best recognition accuracy in both data sets. Our approach's recognition accuracy ranks first in 14 out of 20 classes on ChaLearn MMGR data set and 9 out of 10 classes on ChAirGest data set. The average recognition accuracy of our method is also superior to any other baseline methods, as shown in Table 1 and Table 2. This improvement of our method mainly benefits from two aspects: multi-modal fusion under Bayesian Co-Boosting framework, and boosting learning with feature selection.

The improvement brought by multi-modal fusion is inevitable, since different modalities surely can provide complementary information for each other. "M1+M2" implements late fusion using a weight coefficient α, which requires more training time to determine its optimal value through cross-validation. On the other hand, in our approach, each classifier's weight is determined during boosting process, which avoids extra parameter tuning and is more reasonable and explainable based on the above theoretical analysis.

Comparing the result of "M1", "M2", "Boost.M1", and "Boost.M2", we can see that boosting learning could not necessarily improve the recognition accuracy. This may due to the overfitting caused by the small amount of available training instances. The overfitting problem of boosting methods has been discussed in several literatures (Zhang and Yu, 2005; Reyzin and Schapire, 2006; Vezhnevets and Barinova, 2007; Yao and Doretto, 2010). Considering the high feature dimension of instances, the weak classifier may be too complex to be well trained on such few instances.

Based on the above observation, we tackle the overfitting problem from two aspects. Firstly, feature selection is used to reduce the number of feature dimensions while preserving enough discriminative information, which alleviates overfitting brought by the small size sample problem. Secondly, Bayesian Co-Boosting is employed to combine two weak classifiers together with collaborative training strategy, and each modality can provide complementary information for the other modality. Therefore, the amount of available training information for classifiers is actually increased to avoid overfitting problem to some extent.

As demonstrated in Table 1 and Table 2, "Boost.Sel.M1" and "Boost.Sel.M2" outperform their corresponding training methods without feature selection. On this basis, after applying Co-Boosting method to fuse two modalities, our proposed "BayCoBoost" achieves superior recognition accuracy than all baseline methods.

As for the computation complexity, we compare the average classification time for each method. It takes around 0.31s/0.11s for our proposed "BayCoBoost" method to label an instance in ChaLearn MMGR and ChAirGest data set, respectively. Although non-boosting methods can operate at higher speed (for "M1+M2", the time is about 0.037s/0.013s), we think it is worthy to spend more time since our method's performance is superior to these methods, especially for the second data set. Another remarkable comparison is that by using feature selection strategy, "Boost.Sel.M1" and "Boost.Sel.M2" not only run twice as fast as "Boost.M1" and "Boost.M2", due to the lower classifier's complexity, but also outperform them in the classification performance. This also proves that the effectiveness of the feature selection strategy in our "BayCoBoost" method.

5. Conclusion

In this paper, a novel Bayesian Co-Boosting training framework for multi-modal gesture recognition is proposed. The merits of our work are three-fold: first, the collaborative training between multiple modalities provides complementary information for each modality; second, the boosting learning combines weak classifiers to construct a strong classifier of higher accuracy; third, the Bayesian perspective theoretically ensures that the training error of our method is minimized through iterations. Feature selection and multi-modal fusion are naturally embedded into the training process, which bring significant improvement to the recognition accuracy. Experimental results on two multi-modal gesture recognition

data sets prove the effectiveness of our proposed approach. Moreover, our proposed framework can be easily extended to other related tasks in multi-modal scenarios, such as object detection and tracking.

Acknowledgments

This work was supported in part by 973 Program (Grant No. 2010CB327905), National Natural Science Foundation of China (Grant No. 61332016, 61202325), and Key Project of Chinese Academy of Sciences (Grant No. KGZD-EW-103-5).

Appendix A. Skeleton Feature Extraction

The Kinect™ sensor is able to provide 3D position information for 20 joint points of human body. We denote the original 3D coordinates of these joints as $(x_i, y_i, z_i), i = 1, \ldots, 20$.

In order to extract the skeleton feature which is invariant to user's position, orientation, and body size, we perform the following transformations:

1. Select one joint point as the origin of the normalized coordinate system.
   Translate all joint points to move the selected point to the origin.
2. Select three joint points to construct the reference plane.
   Rotate the reference plane so that it is orthogonal to the z-axis.
3. Calculate the distance sum of 19 directly connected joint pairs.
   Normalize all coordinates so that the sum is equal to 1.

After above transformations, we can obtain the normalized 3D coordinates (x_i^*, y_i^*, z_i^*), which are invariant to the user's position, orientation, and body size.

Since most gestures are performed with upper body, and the lower body's movement may interfere the recognition of gestures, we only select joint points in the upper body for feature extraction. The final feature vector consists of four parts:

1. Absolute 3D position of joint points.
2. Relative 3D position of joint points, defined on directly connected joint pairs.
3. First order difference in time of part 1 in the feature vector.
4. First order difference in time of part 2 in the feature vector.

References

I. Bayer and T. Silbermann. A multi modal approach to gesture recognition from audio and video data. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 461–466, 2013.

K.K. Biswas and S.K. Basu. Gesture recognition using Microsoft Kinect. In *the 5th International Conference on Automation, Robotics and Applications*, pages 100–103, 2011.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

A. Boyali and M. Kavakli. A robust gesture recognition algorithm based on sparse representation, random projections and compressed sensing. In *IEEE Conference on Industrial Electronics and Applications*, pages 243–249, 2012.

G.S. Chambers, S. Venkatesh, G.A.W. West, and H.H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 1082–1085, 2002.

X. Chen and M. Koskela. Online RGB-D gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 467–474, 2013.

S. Eickeler, A. Kosmala, and G. Rigoll. Hidden Markov model based continuous online gesture recognition. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 2, pages 1206–1208, 1998.

M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis. A hidden Markov model-based continuous gesture recognition system for hand motion trajectory. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante. Multi-modal gesture recognition challenge 2013: Dataset and results. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 445–452, 2013.

Yoav F. and Robert E.S. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

S.W. Foo, Y. Lian, and L. Dong. Recognition of visual speech elements using adaptively boosted hidden Markov models. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):693–705, 2004.

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, 1995.

Y.F.A. Gaus, F. Wong, K. Teo, R. Chin, R.R. Porle, L.P. Yi, and A. Chekima. Comparison study of hidden Markov model gesture recognition using fixed state and variable state. In *IEEE International Conference on Signal and Image Processing Applications*, pages 150–155, 2013.

D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H.J. Escalante. ChaLearn gesture challenge: Design and first results. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, 2012.

R.L. Kashyap. Optimal feature selection and decision rules in classification problems with time series. *IEEE Transactions on Information Theory*, 24(3):281–288, 1978.

J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

Y. M. Lui. A least squares regression framework on manifolds and its application to gesture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 13–18, 2012a.

Y.M. Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13(1):3297–3321, 2012b.

Y.M. Lui, J.R. Beveridge, and M. Kirby. Action classification on product manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–839, 2010.

S. Malassiotis, N. Aifanti, and M.G. Strintzis. A gesture recognition system using 3D data. In *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 190–193, 2002.

M.R. Malgireddy, I. Inwogu, and V. Govindaraju. A temporal Bayesian model for classifying, detecting and localizing activities in video sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 43–48, 2012.

V. Mantyla, J. Mantyjarvi, T. Seppanen, and E. Tuulari. Hand gesture recognition of a mobile device user. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 281–284, 2000.

A. Martin, D. Charlet, and L. Mauuary. Robust speech/non-speech detection using LDA applied to MFCC. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 237–240, 2001.

S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

F. Mörchen. Time series feature extraction for data mining using DWT and DFT. Technical report, Philipps-University Marburg, 2003.

K.P. Murphy. *Machine learning: A probabilistic perspective*. MIT Press, 2012.

K. Nandakumar, K.W. Wan, S.M.A. Chan, W.Z.T. Ng, J.G. Wang, and W.Y. Yau. A multi-modal gesture recognition system using audio, video, and skeletal joint data. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 475–482, 2013.

C. Oz and M.C. Leu. American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7):1204–1213, 2011.

T. Pylvänäinen. Accelerometer based gesture recognition using continuous HMMs. In *Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science*, pages 639–646. Springer Berlin Heidelberg, 2005.

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

J.L. Raheja, R. Shyam, U. Kumar, and P.B. Prasad. Real-time robotic hand control using hand gestures. In *the 2nd International Conference on Machine Learning and Computing*, pages 12–16, 2010.

L. Reyzin and R.E. Schapire. How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 753–760, 2006.

M. Roccetti, G. Marfia, and A. Semeraro. A fast and robust gesture recognition system for exhibit gaming scenarios. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 343–350, 2011.

S. Ruffieux, D. Lalanne, and E. Mugellini. ChAirGest - a challenge for multimodal mid-air gesture recognition for close HCI. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 483–488, 2013.

J. Suarez and R.R. Murphy. Hand gesture recognition with depth images: A review. In *IEEE RO-MAN*, pages 411–417, 2012.

A. Vezhnevets and O. Barinova. Avoiding boosting overfitting by removing confusing samples. In *Proceedings of the 18th European Conference on Machine Learning*, volume 4701 of *Lecture Notes in Computer Science*, pages 430–441. Springer Berlin Heidelberg, 2007.

P. Viola and M. Jones. Robust real-time face detection. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, volume 2, pages 747–747, 2001.

P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma. Towards a practical face recognition system: Robust registration and illumination by sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 597–604, 2009.

D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

D. Wu, F. Zhu, and L. Shao. One shot learning gesture recognition from RGBD images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12, 2012.

J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 453–460, 2013.

M. Yang and N. Ahuja. Recognizing hand gestures using motion trajectories. In *Face Detection and Gesture Recognition for Human-Computer Interaction*, volume 1 of *The International Series in Video Computing*, pages 53–81. Springer US, 2001.

Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1855–1862, 2010.

M. Yeasin and S. Chaudhuri. Visual understanding of dynamic hand gestures. *Pattern Recognition*, 33(11):1805–1817, 2000.

Y. Yin and R. Davis. Gesture spotting and recognition using salience detection and concatenated hidden Markov models. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 489–494, 2013.

H. Yoon, K. Yang, and C. Shahabi. Feature subset selection and feature ranking for multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 17(9): 1186–1198, 2005.

S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R.B. Rao. Bayesian co-training. In *Advances in Neural Information Processing Systems*, volume 20, pages 1665–1672. MIT Press, 2008.

S. Yu, B. Krishnapuram, R. Rosales, and R.B. Rao. Bayesian co-training. *Journal of Machine Learning Research*, 12:2649–2680, 2011.

Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the Kinect. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, pages 279–286, 2011.

L. Zhang, X. Chen, C. Wang, Y. Chen, and W. Gao. Recognition of sign language subwords based on boosted hidden Markov models. In *Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 282–287, 2005.

T. Zhang and B. Yu. Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, pages 1538–1579, 2005.

Y. Zhang, W. Liang, J. Tan, Y. Li, and Z. Zeng. PCA & HMM based arm gesture recognition using inertial measurement unit. In *Proceedings of the 8th International Conference on Body Area Networks*, pages 193–196, 2013.

F. Zheng, G. Zhang, and Z. Song. Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, 16(6):582–589, 2001.