Learning Sequential Patterns for Lipreading

Eng-Jon Ong
e.ong@surrey.ac.uk
Richard Bowden
r.bowden@surrey.ac.uk

The Centre for Vision, Speech and Signal Processing,
University of Surrey,
Guildford GU27XH, Surrey, UK

Abstract

This paper proposes a novel machine learning algorithm (SP-Boosting) to tackle the problem of lipreading by building visual sequence classifiers based on sequential patterns. We show that an exhaustive search of optimal sequential patterns is not possible due to the immense search space, and tackle this with a novel, efficient tree-search method with a set of pruning criteria. Crucially, the pruning strategies preserve our ability to locate the optimal sequential pattern. Additionally, the tree-based search method accounts for the training set's boosting weight distribution. This temporal search method is then integrated into the boosting framework resulting in the SP-Boosting algorithm. We also propose a novel constrained set of strong classifiers that further improves recognition accuracy. The resulting learnt classifiers are applied to lipreading by performing multi-class recognition on the OuluVS database. Experimental results show that our method achieves state of the art recognition performane, using only a small set of sequential patterns.

1 Introduction

This paper presents a machine learning approach to Lip Reading and proposes a novel learning technique called sequential pattern boosting that allows us to very efficiently search and combine a set of temporal patterns to form strong spatio-temporal classifiers. Any attempt at automatic lip reading needs to address the demanding challenge that the problem is inherently temporal in nature. It is crucial to model and use spatio-temporal information. There already exists a body of work that deals directly with the task of lip reading, all attempting to model and exploit the dynamics of the lip movements. One popular method is to model the temporal information using Hidden Markov Models (HMM). Matthews et al [5] proposed a method that models lip movements using HMMs on Active Appearance Models (AAMs). This was improved by Lan et al [4], where short term temporal information was included in the feature vector. Another possibility is to use geometric information, for example mouth width, area within the inner lip, etc.. [3]. However, this requires accurate tracking of both the inner and outer lip shape, a non-trivial task. An alternative would be to utilise simpler visual features that can be extracted within a bounding-box area containing the mouth. This is the approach that is taken in this paper, where binary comparison features are used. Recently, Local Binary Patterns (LBPs) have been popular as visual features for lipreading. Zhao et al[8] proposed a method that uses LBPs that span both time and space, effectively modelling

BMVC 2011 http://dx.doi.org/10.5244/C.25.55

local spatio-temporal information. Support Vector Machines were then used to build classifiers of feature vectors from histograms of LBP responses. Our proposed method differs from SVM-based methods in that we attempt explicit temporal feature selection by means of the boosting algorithm. This allows us to build classifiers that are smaller and thus more efficient, whilst providing similar accuracy. The rest of the paper is organised as follows. In the next section, we will describe a novel boosting algorithm that efficiently searches and combines optimal sequential patterns for classifying temporal sequences. We also propose a time-constrained set of strong classifiers. In Section 3, we describe and analyse the experimental results of the proposed methods before concluding in Section 4.

2 Sequential Pattern Boosted Classifiers

In this section, we propose a novel machine learning method for learning strong spatio-temporal classifiers by means of boosting. A boosted classifier consists of a linear combination of a number (S) of selected weak classifiers, and take the form of: $H(I) = \sum_{i=1}^{S} \alpha_i h_i(I)$. The weak classifiers h_i are selected iteratively based on weights formed during training. In order to determine the optimal weak classifier at each Boosting iteration, the common approach is to exhaustively consider the entire set of candidate weak classifiers and finally select the best weak classifier (i.e. that with the lowest weighted error). However, as we will see in Section 2.2, when we are dealing with temporal patterns, the number of weak classifiers becomes too large, making an exhaustive search no longer feasible. To address this, we propose a novel Boosting algorithm called Sequential Pattern Boosting (*SP-Boost*). This method will result in a classifier that combines a set of weak but salient temporal patterns that are useful for recognising a class of temporal sequences.

2.1 Sequential Pattern Weak Classifiers

In order to classify temporal patterns, we start with the representation of sequential patterns frequently used in datamining [2, 7].

Definition 2.1 *Given a binary feature vector $F = (f_i)_{i=1}^{D}$, let $T \subset \{1,\ldots,D\}$ be a set of integers where $\forall t \in T$, $f_t = 1$, that is, T represents all the dimensions of F that have the value of 1. We call T as an* itemset. *Let $\mathbf{T} = (T_i)_{i=1}^{|\mathbf{T}|}$ be a sequence of $|\mathbf{T}|$ itemsets. We denote $\mathbf{T}$ as a* sequential pattern.

Attempting to use sequential patterns for classification requires an operator to determine if a sequential pattern is present within a given feature vector sequence:

Definition 2.2 *Let $\mathbf{T}$ and $\mathbf{I}$ be sequential patterns. We say that the sequential pattern $\mathbf{T}$ is present in $\mathbf{I}$ if there exists a sequence $(\beta_i)_{i=1}^{|\mathbf{T}|}$, where $\beta_i < \beta_j$ when $i < j$ and $\forall i = \{1,\ldots,|\mathbf{T}|\}, T_i \subset I_{\beta_i}$. This relationship is denoted with the $\subset_S$ operator, i.e. $\mathbf{T} \subset_S \mathbf{I}$. Conversely, if the sequence $(\beta_i)_{i=1}^{|\mathbf{T}|}$ does not exist, we denote it as $\mathbf{T} \not\subset_S \mathbf{I}$.*

From this, we can then define a sequential pattern weak classifier as follows: Let $\mathbf{T}$ be a given sequential pattern and $\mathbf{I}$ be an itemset sequence derived from some input binary vector sequence F. A sequential pattern weak classifier or *SP weak classifier*, $h^{\mathbf{T}}(\mathbf{I})$, can be constructed as follows:

$$h^{\mathbf{T}}(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{T} \subset_S \mathbf{I} \\ -1, & \text{if } \mathbf{T} \not\subset_S \mathbf{I} \end{cases} \tag{1}$$

Given a weak classifier $h^{\mathbf{T}}(\mathbf{I})$ defined above, we can now measure its performance on the set of positive and negative example training sequences. We define the positive training set as $\mathscr{I}^+ = \{(\mathbf{I}^+{}_i, w_i^+)\}_{i=1}^{N^+}$, where N^+ is the number of training examples and $\mathbf{I}_i^+$ is a sequential pattern and associated with it is the boosted weight w_i^+. Similarly, the negative training set is defined as $\mathscr{I}^- = \{(\mathbf{I}^-{}_i, w_i^-)\}_{i=1}^{N^-}$, where N^- is the number of training examples and $\mathbf{I}_i^-$ is a single negative training sequential pattern with boosting weight w_i^-. To achieve this, we define the positive training set error, $\varepsilon^{\mathbf{T}+}$, of the weak classifier $h^{\mathbf{T}}(\mathbf{I})$ as: $\varepsilon^{\mathbf{T}+} = \sum_{i=1}^{N^+} w_i^+ (h^T(\mathbf{I}_i^+) \neq 1)$. Similarly, the negative training set error $\varepsilon^{\mathbf{T}+}$, of the weak classifier $h^{\mathbf{T}}(\mathbf{I})$ can be defined as: $\varepsilon^{\mathbf{T}-} = \sum_{i=1}^{N^-} w_i^- (h^T(\mathbf{I}_i^-) \neq -1)$. Finally, the overall error of $h^{\mathbf{T}}(\mathbf{I})$ on the entire training dataset is defined as: $\varepsilon^{\mathbf{T}} = \varepsilon^{\mathbf{T}+} + \varepsilon^{\mathbf{T}-}$. It will be useful in the later sections to note that the errors: $\varepsilon^{\mathbf{T}}, \varepsilon^{\mathbf{T}+}, \varepsilon^{\mathbf{T}-}$ are associated with the pattern $\mathbf{T}$.

2.2 Efficient Tree-Based Search for Optimal Sequential Pattern

We are now faced with the task of selecting the best sequential pattern weak classifier, an extremely challenging task due the vast amount of possible sequential pattern configurations. With D features, sequential patterns up to length N and maximum of K items, the number of weak classifiers is the binomial coefficient polynomial: $\binom{D}{K}^N$. In the experiments performed here ($D = 900, K = 3, N = 7$), the total number of weak classifiers is $5x10^{58}$, making pre-generating the entire set of weak classifiers and evaluating each individually not feasible. We approach the task of selecting the optimal SP weak classifier in terms of a tree-based search. Here, sequential patterns are arranged into a tree-based structure[2].Each tree node corresponds to a particular sequential pattern that increase in complexity as we traverse deeper into the tree. To traverse different nodes on the tree, we introduce two operators for extending a sequential pattern: the first adds a new element to its last set and the second appends another sequential pattern to its end. These correspond to a breadth and depth traversal in the tree structure.

Definition 2.3 *Let P be an itemset and $\mathbf{T} = (T_i)_{i=1}^{|\mathbf{T}|}$ a sequential pattern. We can perform an* itemset extension *to $\mathbf{T}$ using P with the itemset extension operator $\cup_s$: $\mathbf{T} \cup_s P = (T_1, ..., T_{|\mathbf{T}|} \cup P)$. Let $\mathbf{P} = \{P_i\}_{i=1}^{|\mathbf{P}|}$ be a sequential pattern. We can then* sequence extend *$\mathbf{T}$ with $\mathbf{P}$ using the sequence extension operator $+_s$: $\mathbf{T} +_s \mathbf{P} = (T_1, ..., T_{|\mathbf{T}|}, P_1, ..., P_{|\mathbf{P}|})$.*

We find that both operators $\cup_s$ and $+_s$ have important properties in relation to $\subset_s$ that will be crucial to developing suitable tree-pruning criteria later.

Lemma 2.1 *Let $\mathbf{P}$,$\mathbf{T}$ and $\mathbf{R}$ be sequential patterns. If $\mathbf{P} +_s \mathbf{T} \subset_s \mathbf{R}$, then $\mathbf{P} \subset_s \mathbf{R}$, $\mathbf{T} \subset_s \mathbf{R}$.*

Proof Since $\mathbf{P} +_s \mathbf{T} \subset_s \mathbf{R}$, then $\exists(\beta_1, ..., \beta_{|\mathbf{P}|}, \gamma_1, ..., \gamma_{|\mathbf{T}|})$ where $P_i \subset R_{\beta_i}, \forall i = [1, |\mathbf{P}|]$ and $T_i \subset R_{\gamma_i}, \forall i = [1, |\mathbf{T}|]$. Thus, by definition of $\subset_s$, $\mathbf{P} \subset_s \mathbf{R}$ and $\subset_s$, $\mathbf{P} \subset_s \mathbf{R}$.

Lemma 2.2 *Let $\mathbf{P}$ and $\mathbf{R}$ be seq. pats. and T an itemset. If $\mathbf{P} \cup_s T \subset_s \mathbf{R}$, then $\mathbf{P} \subset_s \mathbf{R}$.*

Proof Since $\mathbf{P} \cup_s T \subset_s \mathbf{R}$, then $\exists(\beta_1, ..., \beta_{|\mathbf{P}|})$ where $P_i \subset R_{\beta_i}, \forall i = [1, |\mathbf{P}| - 1]$ and $P_{|\mathbf{P}|} \cup T \subset R_{\beta_{|\mathbf{P}|}}$. This means that $P_{|\mathbf{P}|} \subset R_{\beta_{|\mathbf{P}|}}$, thus $\mathbf{P} \subset_s \mathbf{R}$.

2.3 Tree Pruning Methods

In order to avoid having to search the entire sequential pattern tree, which is equivalent to performing an exhaustive search, a number of tree-pruning strategies are introduced. Importantly, these tree-pruning strategies should not stop us from selecting the optimal weak classifier. This can be achieved by identifying and pruning at sequential patterns that when extended, will never result in weak classifier errors below the current best error. We will denote these sequential patterns as *prune points*. We start by observing that the more complex a sequential pattern becomes (e.g. by itemset or sequence extension), the number of training examples containing it reduces. Specifically:

Lemma 2.3 *Let* $\mathbf{P},\mathbf{T}$ *be sequential patterns, R be an itemset and* $\mathscr{I} = \{(\mathbf{I}_i, w_i)\}_{i=1}^{N}$ *be a set of N training examples in the form of sequential patterns and its associated weights. Let* $\mathscr{I}^P = \{(\mathbf{I}, w) \in \mathscr{I} : \mathbf{P} \subset_s \mathbf{I}\}$*, the set of training examples with sequential pattern P in them. Firstly,* $\mathscr{I}^{\mathbf{P}+_s\mathbf{T}} \subset \mathscr{I}^{\mathbf{P}}$*. Secondly,* $\mathscr{I}^{\mathbf{P}\cup_s\mathbf{R}} \subset \mathscr{I}^{\mathbf{P}}$*. (Proof in Appendix A).*

This has important implications on how errors in the positive dataset ($\mathscr{I}^+$)and negative dataset ($\mathscr{I}^-$) evolve as a weak classifier's associated sequential pattern becomes more complex. In particular, an increase in complexity of a sequential pattern results in monotonically increasing positive errors and monotonically decreasing negative dataset errors:

Lemma 2.4 *Let* $h^{\mathbf{T}}(\mathbf{X})$ *be associated with sequential pattern* $\mathbf{T}$ *with positive and negative training errors of* $\varepsilon^{\mathbf{T}+}$ *and* $\varepsilon^{\mathbf{T}-}$ *respectively. Let* $\mathbf{P}$ *be a sequential pattern and R an itemset, then:* $\varepsilon^{\mathbf{T}+} \leq \varepsilon^{(\mathbf{T}+_s\mathbf{P})+}$ *and* $\varepsilon^{\mathbf{T}+} \leq \varepsilon^{(\mathbf{T}\cup_s R)+}$*. Additionally, we have* $\varepsilon^{\mathbf{T}-} \geq \varepsilon^{(\mathbf{T}+_s\mathbf{P})-}$ *and* $\varepsilon^{\mathbf{T}-} \geq \varepsilon^{(\mathbf{T}\cup_s R)-}$*. (Proof in Appendix B)*

2.3.1 Prune Criteria 1: Error Lower Bound

The first criteria for stopping any further sequence extensions is achieved by considering the lower-bound for the sequential pattern weak classifiers error. Firstly, the error of the weak classifier $h^{\mathbf{T}}(\mathbf{X})$ is $\varepsilon^{\mathbf{T}} = \varepsilon^{\mathbf{T}+} + \varepsilon^{\mathbf{T}-}$. However, Lemma 2.4 shows us that any pattern extensions of $\mathbf{T}$ will lead to monotonically increasing errors of $\varepsilon^{\mathbf{T}+}$, whilst $\varepsilon^{\mathbf{T}-}$ monotonically decreases. From this, it is clear that the error on the positive dataset is the lower bound for the error of future classifiers based on $\mathbf{T}$. Thus, suppose the minimum weak classifier error found thus far is ε_{min}, and we are given a new itemset R or new sequential pattern $\mathbf{P}$ to extend $\mathbf{T}$ by, the error lower bound pruning criteria is introduced:

a) If $\varepsilon^{(\mathbf{T}+_s\mathbf{P})+} \geq \varepsilon_{min}$, then the pattern $T +_s \mathbf{P}$ is a prune point and all its superset patterns need not be given further consideration.

b) Similarly, if $\varepsilon^{(T\cup_s R)+} \geq \varepsilon_{min}$, then the pattern $T \cup_s R$ is a prune point and all its superset patterns need not be given further consideration.

c) If $\varepsilon^{\mathbf{T}-} \leq \varepsilon^{\mathbf{T}+}$, then $\mathbf{T}$ is also a prune point.

2.3.2 Prune Criteria 2: Datamining A-priori Principles

It is also possible to exploit various pruning steps introduced by research in data-mining [1, 2]. These take the form of the following two pruning criteria:

a) We can remove items for consideration during itemset extensions if they were also rejected when considered for sequence extensions. Suppose pattern $\mathbf{T}$ is a sequence extended by (p) where p is an itemset index. It was then found that $\mathbf{T}+_s(p)$ is a prune point (i.e. $\varepsilon^{\mathbf{T}+_s(p)+} \geq \varepsilon_{min}$), then for all possible itemsets R, we have $\varepsilon^{\mathbf{T}\cup_s\{R\cup_s(p)\}+} \geq \varepsilon_{min}$. Consequently, it is no longer necessary to consider p in the itemset extensions of patterns based on T.

b) We can also remove patterns for consideration during sequence extensions if they were also rejected when considered for itemset extensions. Given a pattern $\mathbf{T}$, and an itemset $\{p\}$ for itemset extension. Suppose it was found that $\mathbf{T}\cup_s\{p\}$, is a prune point (i.e. $\varepsilon^{\mathbf{T}\cup_s\{p\}+} \geq \varepsilon_{min}$), then for all possible patterns $\mathbf{R}$, we have $\varepsilon^{\mathbf{T}+_s\mathbf{R}\cup_s\{p\}+} \geq \varepsilon_{min}$. Consequently, it is no longer necessary to consider (p) in the future sequence extensions of patterns based on T.

2.4 Optimal SP Weakclassifier Selection

Using the pruning criteria introduced above, the algorithm for selecting the best sequential pattern weak classifier is given in Algorithm 1. This algorithm works on a depth-first search basis. Additionally, a queue-based algorithm is proposed, thus avoiding the need for recursion. The proposed algorithm works on creating patterns and appending them onto a queue $Q = q_i{}_{i=1}^{|Q|}$, where each element is a tuple with the following form: $q_i = (\mathbf{T}_i, K_i, S_i)$, where $\mathbf{T}_i = (T_{i,j})_{j=1}^{|\mathbf{T}_i|}$ is a sequential pattern, $K_i = \{k_{i,j})_{j=1}^{|\mathbf{T}_i|}\}$ is the set of items to itemset-extend $\mathbf{T}_i$ with, and $S_i = (s_{i,j})_{j=1}^{|\mathbf{T}_i|}$ is the set of items to sequence extend $\mathbf{T}_i$ with.

2.4.1 Weak Classifier Selection Initialisation

Prior to running the algorithm, an initial queue needs to be created. To this end, we initially create the following set of patterns $\mathbf{T}^{init} = \{(1),...,(D)\}$. The errors for all the classifiers associated with these patterns are given in the set $\Upsilon_{init} = \{\varepsilon^{\mathbf{T}_1^{init}},...,\varepsilon^{\mathbf{T}_D^{init}}\}$. We also define the initial set of elements for itemset expansions as: $\{K_i^{init}\}_{i=1}^{D}$ where $K_i^{init} = \{i+1,...,D\}$. Finally, we define the initial set for sequence extension as $S^{init} = \{1,...,D\}$. Using the above, the initial set queue for selecting the best sequential pattern weak classifier is given as: $Q = \{(\mathbf{T}_i^{init}, K_i^{init}, S^{init})\}_{i=1}^{D}$. Additionally, we set the initial best pattern and error as: $\mathbf{T}_{best} = \mathbf{T}_k$, where $k = \arg\min_k \Upsilon$ and $\varepsilon_{best} = min(\Upsilon_{init})$.

2.5 Sequential Pattern Boosted Classifier

Using the selection step above, the SP-Boost Algorithm is then given in Algorithm 2. The algorithm is similar to the AdaBoost algorithm, but with one crucial difference. The selection of the weak classifier is performed using Algorithm 1. This is equivalent to an exhaustive search of possible sequential pattern weak classifiers for the optimal one. The result is a set of sequential pattern weak classifiers that are extracted based on the boosted weights that will be linearly combined to form a strong classifier for a target class.

2.6 Constrained Sequential Pattern Classifiers

In order to improve the performance of trained sequential pattern classifiers, we divide an input sequential pattern into a number (G) of equal sized segments. Following this, G strong

Algorithm 1 SP Weak Classifier Selection Algorithm

```
Perform queue initialisation (Section 2.4.1)
while $Q \neq \emptyset$ do
    Remove the last item of $Q$ and denote it as the tuple $(\mathbf{T}_L, K_L, S_L)$
    if $\varepsilon^{\mathbf{T}+} \geq \varepsilon^{\mathbf{T}-}$ {Prune Criteria 1c} then
        continue;
    end if
    {Perform sequence extensions first}
    if $|\mathbf{T}_L| < S_{max}$ then
        $B = \{\}$
        for $i = \{1 ... |\mathbf{T}_L|\}$ do
            Make the extended sequence, $\mathbf{T}' = \mathbf{T} +_s (s_{L,i})$ {Prune Criteria 2a}
            if $\varepsilon^{(\mathbf{T}+_s(s_{L,i}))+} \geq \varepsilon_{best}$ {Prune Criteria 1a} then
                $B = B \cup \{s_{L_i}\}$
            end if
        end for
        Make the set of items to sequence extend T with: $C = S_L - B$
        for $i = \{1, ..., |C|\}$ do
            $\mathbf{T}' = \mathbf{T} +_s (s_{L,i})$ and $C' = \{c \in C : c > s_{L,i}\}$
            Append $\mathbf{T}'$ into $Q$: $Q = Q \cup (\mathbf{T}', C', C)$
            if $\varepsilon^{\mathbf{T}'} < \varepsilon_{best}$ then
                $\mathbf{T}_{best} = \mathbf{T}', \varepsilon_{best} = \varepsilon^{\mathbf{T}'}$
            end if
        end for
    end if

    { Now, perform itemset extensions }
    Make the set of items to itemset extend T with: $D = K_L - B$ {Prune Criteria 2b}
    $B = \{\}$
    for $i = \{1, ..., |D|\}$ do
        Do itemset extension, $\mathbf{T}' = \mathbf{T} \cup_s (k_{L,i})$
        if $\varepsilon^{T \cup_s (k_{L,i})+} \geq \varepsilon_{best}$ {Prune Criteria 1b} then
            $B = B \cup \{s_{L_i}\}$
        end if
    end for
    $D' = D - B$
    for $i = \{1, ..., |D'|\}$ do
        $\mathbf{T}' = \mathbf{T} \cup_s (k_{L,i}), E = \{d \in D' : d > k_{L,i}\}$
        Append $\mathbf{T}'$ into $Q$: $Q = Q \cup (T', C, E)$
        if $\varepsilon^{\mathbf{T}'+} < \varepsilon_{best}$ then
            $\mathbf{T}_{best} = \mathbf{T}', \varepsilon_{best} = \varepsilon^{\mathbf{T}'}$
        end if
    end for
end while

Return $h^{\mathbf{T}_{best}}, \varepsilon_{best}$
```

Algorithm 2 SP-Boost Algorithm

```
Initialise positive example weights: $\forall w_i^+ \in W^+, w_i^+ = 1/(N^+ + N^-)$
Initialise negative example weights: $\forall w_i^- \in W^-, w_i^- = 1/(N^+ + N^-)$
for $t = 1, ..., M$ do
    Select $(h_t = h^{\mathbf{T}_{best}})$ using Algorithm 1
    Obtain the weight $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_{best}}{\varepsilon_{best}}$ for $h_t$
    Update positive weights: $w_i^+ = w_i^+ \exp(-\alpha_i h_t(I_i^+))$
    Update negative weights: $w_i^- = w_i^- \exp(\alpha_i h_t(I_i^-))$
    Normalise weights: $\sum_{i=1}^{N^+} w_i^+ + \sum_{i=1}^{N^-} w_i^- = 1$
end for
Return the strong classifier $H(I) = \sum_{i=1}^{M} \alpha_i h_i(I)$.
```

classifiers $\{H_g\}_{g=1}^{G}$ are trained using Algorithm 2, each using a particular segment of the training sequences. Given a new input sequence $\mathbf{I}$, it is first divided into G equal segments $\mathbf{I'_g}_{g=1}^{G}$. The output of the constrained classifiers is: $\sum_{g=1}^{G} H_g(I'_g)$. We demonstrate in Section 3 that we obtain better results when the number of segments are increased.

3 Experiments

This section describes the experiments for evaluating the performance of the proposed SP-Boosting algorithm in lip reading. We chose the OuluVS database [8] to allow for comparison against existing state-of-the-art lip reading work [8]. The database contains video sequences of 20 subjects in total (see Figure 1). Each subject read 5 repetitions of 10 different phrases: *'excuse me'(ExM), 'goodbye'(Gb), 'hello'(Hlo), 'nice to see you'(Syu), 'have a nice day'(Nc), 'I'm sorry'(Sry), 'thank you'(Tk), 'have a good time'(Tm) and 'your welcome'(Wl),* resulting in 1000 example sequences. The resolution of the images in all sequences is 720x576 captured at 25fps. The aim of the experiments described in this section is to recognise which of the 10 phrases was spoken.

The visual features selected for the experiments take the form of simple comparative binary features. These features are similar to LBPs in that binary descisions based on relative intensity differences are used. However, our visual features also allow us to capture non-local intensity relations. In order to extract suitable visual features (see Fig. 1b), four points on the mouth of the subjects were first automatically tracked using the linear predictor tracking method [6]. Following this, a bounding box aligned with the orientation of the mouth is extracted. The mean width and height of the mouth bounding box of a subject is then obtained and this determines the size of the final bounding box used throughout all the sequences for a particular subject. A grid of 20x20 points is formed in the bounding box. Next, two random sets of 450 grid point indices are obtained and fixed for all subjects and experiments. A binary feature vector is then formed by performing 450 binary comparisons of the underlying intensities for the two grid point index sets (see Figure 1c). The final feature vector is obtained by concatenating the original binary feature vector with its inverse feature vector, yielding a 900-dimensional binary feature vector for each frame. All the example video sequences were then converted into sequential patterns of length 21, with the middle of the example sequence being the middle of the associated sequential pattern.

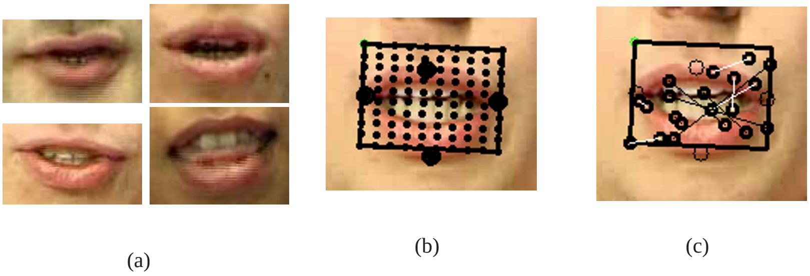

Figure 1: (a) Examples of the mouths for various subjects in the OuluVS database.(b) Visualisation of the grid of points in the mouth bounding box based on four tracked points on the mouth (large black points). (c) Visualisation of the 10 binary comparison features used described in Section 3.

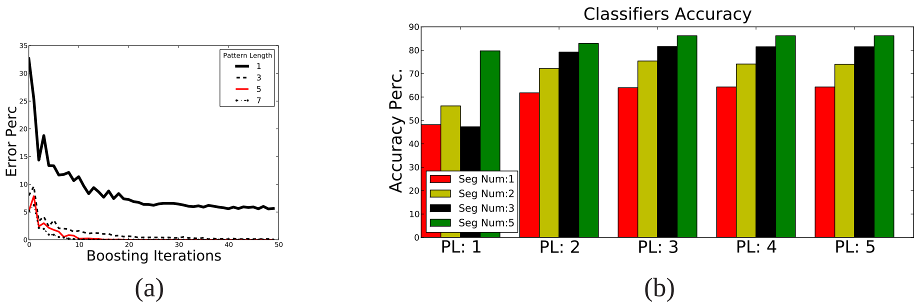

Figure 2: (a)Average of training error curves of different pattern lengths (1,3,5,7). (b)Test data recognition accuracy for classifiers trained with different maximum pattern lengths (PL: 1,2,3,5,7) and segment numbers(1,2,3,5).

3.1 Experimental Setup

The performance of the proposed method is evaluated in a subject-dependent setting using leave-one-out video cross-validation. For each subject, a single video was used as a test sequence. The remaining video sequences for this subject was used as training examples. For each cross validation fold, a set of one-vs-one classifiers (90 in total) using the proposed method described in Section 2 was trained. A voting scheme was then used to obtain a class label for each test sequence. All the classifiers were boosted with a maximum of 50 iterations. In order to analyse the roles of different sequential pattern lengths and constraint segment numbers, we repeated the cross validation experiments with all combinations of classifiers trained with maximum sequential pattern lengths of 1,2,3,5 and 7 and segment numbers of 1,2,3 and 5.

3.2 Experimental Results

It was found that for all one-vs-one classifiers trained, the training errors decreased faster as the maximum length for the sequential patterns was increased. Examples of the error curves for various classifiers trained can be seen in Figure 2a. The recognition accuracy of

	ExM	Gb	Hlo	Hw	Nc	Syu	Sry	Tk	Tm	Wl
ExM	85.3	3.2	3.2	1.1	2.1	1.1	1.1	1.1	2.1	0.0
Gb	0.0	87.4	0.0	1.1	2.1	1.1	0.0	0.0	1.1	7.4
Hlo	0.0	0.0	83.2	0.0	2.1	4.2	2.1	6.3	0.0	2.1
Hw	0.0	0.0	1.1	92.6	1.1	1.1	1.1	1.1	0.0	2.1
Nc	5.3	2.1	0.0	1.1	83.2	3.2	1.1	3.2	0.0	1.1
Syu	1.1	1.1	0.0	1.1	1.1	71.6	1.1	21.1	0.0	2.1
Sry	1.1	1.1	0.0	2.1	1.1	0.0	94.7	0.0	0.0	0.0
Tk	3.2	0.0	3.2	0.0	3.2	7.4	0.0	81.1	0.0	2.1
Tm	1.1	3.2	0.0	1.1	0.0	0.0	1.1	0.0	90.5	3.2
Wl	0.0	3.2	0.0	1.1	1.1	1.1	0.0	0.0	0.0	93.7

Figure 3: Test data confusion matrix.

the proposed method for different parameter configurations can be seen in Figure 2b. Firstly, we note that as the pattern length increased, the recognition accuracy significantly increased as well. An example would be the recognition rate of 64.3% with pattern length 1 increasing to a rate of 86.2% when the maximum pattern length was 7. Additionally, we get another significant increase in performance when the constrained classifier segment number was increased to 5 (i.e. input sequence divided into 5 equal segments). For example, when the maximum pattern length was 5, increasing the segments from 1 to 3 resulted in the accuracy increasing from 79% to 86%. However, we also find that after 3 segments, the accuracy rate converged to 86.2%. The highest average test recognition rate was 86.2%. In comparison, the recognition rate obtained in [8] was 70.2%. The confusion matrix for the best performing classifiers is shown in Figure 3.

4 Conclusions

This paper proposed a novel machine learning algorithm (SP-Boosting) to tackle the problem of lipreading by building visual sequence classifiers that are based on salient sequential patterns. Since an exhaustive search of optimal sequential patterns is not possible due to the immense search space, a novel efficient tree-based search strategy with pruning conditions for reducing the required search space was proposed. Crucially, the pruning strategies preserves our ability to locate the optimal sequential pattern. The search method is then integrated into the boosting framework resulting in the SP-Boosting algorithm. The learnt sequential-pattern strong classifiers were applied to lipreading by performing multi-class recognition on the OuluVS database. Experimental results show that our method achieves state of the art recognition performance, using only a small set of sequential patterns.

A Proof of Lemma 2.3

For the first case, let $(\mathbf{I},w) \in \mathscr{I}^{\mathbf{P}+_s\mathbf{T}}$ and thus $\mathbf{P} +_s \mathbf{T} \subset_s \mathbf{I}$. From Lemma 2.1, this means that $\mathbf{P} \subset_s \mathbf{I}$ and thus $(\mathbf{I},w) \in \mathscr{I}^{\mathbf{P}}$. Thus, $\mathscr{I}^{\mathbf{P}+_s\mathbf{T}} \subset \mathscr{I}^{\mathbf{P}}$. For the second case, let $(\mathbf{I},w) \in \mathscr{I}^{\mathbf{P}\cup_s\mathbf{R}}$. From Lemma 2.2, this means that $\mathbf{P} \subset_s \mathbf{I}$ and thus $(\mathbf{I},w) \in \mathscr{I}^{\mathbf{P}}$. Thus, $\mathscr{I}^{\mathbf{P}\cup_s\mathbf{R}} \subset \mathscr{I}^{\mathbf{P}}$.

B Proof of Lemma 2.4

Let $W^+ = \sum_{(\mathbf{I}^+,w^+)\in\mathscr{I}^+} w^+$. It can be seen that $\varepsilon^{\mathbf{T}+} = W^+ - \sum w^+, \forall(\mathbf{I},w) \in \mathscr{I}^{\mathbf{T}+}$. Thus, from Lemma 2.3, since $\mathscr{I}^{\mathbf{T}+_s\mathbf{P}+} \subset \mathscr{I}^{\mathbf{T}+}$ and $\mathscr{I}^{\mathbf{T}\cup_s R+} \subset \mathscr{I}^{\mathbf{T}+}$, then $\varepsilon^{\mathbf{T}+} \leq \varepsilon^{(\mathbf{T}+_s\mathbf{P})+}$ and $\varepsilon^{\mathbf{T}+} \leq \varepsilon^{(\mathbf{T}\cup_s R)+}$. Next, it can be seen that $\varepsilon^{\mathbf{T}-} = \sum w^-, \forall(\mathbf{I},w^-) \in \mathscr{I}^{\mathbf{T}-}$. From Lemma 2.3, we have $\mathscr{I}^{\mathbf{T}+_s\mathbf{P}-} \subset \mathscr{I}^{\mathbf{T}-}$ and $\mathscr{I}^{\mathbf{T}\cup_s R-} \subset \mathscr{I}^{\mathbf{T}-}$, thus $\varepsilon^{\mathbf{T}-} \geq \varepsilon^{(\mathbf{T}+_s\mathbf{P})-}$ and $\varepsilon^{\mathbf{T}-} \geq \varepsilon^{(\mathbf{T}\cup_s R)-}$.

Acknowledgement

The research leading to these results has received funding from the UK Home Office and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231135 - DictaSign.

References

[1] R. Agrawal, T. Imielinkski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using bitmaps. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2002.

[3] N. Brooke. Using the visual component in automatic speech recognition. In *Proceedings of International Conference on Spoken Language*, 1996.

[4] Y. Lan, B. Theobald, E. Ong, and R. Harvey. Improving visual features for lip-reading. In *Proceedings of International Conference on Auditory-Visual Speech Processing*, 2010.

[5] I. Matthews, T. Cootes, J. Bangham, S. Cox, and R. Harvey. Extraction of visual featurs for lipreading. *IEEE Transactions on Pattern Analysis and Machine Learning*, 24(2), 2002.

[6] E. Ong, Y. Lan, B. Theobald, R. Harvey, and R. Bowden. Robust facial feature tracking using selected multi-resolution linear predictors. In *Proceedings of the 12th International Conference on Computer Vision*, 2009.

[7] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. Technical report, IBM Almaden Research Center, December 1995.

[8] G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7), 2009.