Semi-Supervised Tree Support Vector Machine for Online Cough Recognition

Huynh Thai Hoa, Tran Vu An, Tran Huy Dat

Human Language Technology Department
A-STAR * Institute for Infocomm Research, Singapore

Abstract
Pneumonia and asthma are among the top causes of death worldwide with 300 million people suffered. In the year 2005, 255,000 people died only because of asthma [1]. Good controlling requires both proper medication and continual monitoring over days and nights. In this paper, we introduce a novel classifier, namely Semi-Supervised Tree Support Vector Machine, to target the problem of cough detection and monitoring. It will adaptively analyze the distribution of samples’ confidence metrics, automatically select the most informative samples and re-train the core Tree SVM classifier inside accordingly. Besides, we also introduce a new way to build Tree SVM, based on Fisher Linear Discriminant (FLD) analytic. All are meant to improve final system performance, and our proposed classifier has really demonstrated good improvement over conventional method; validated on a database consists of comprehensive body-sounds, recorded with wearable contact microphone. Index Terms: cough recognition, cough classification, semi-supervised, tree SVM, semi-supervised tree SVM

1. Introduction
Pneumonia and asthma exhibit many symptoms such as chest tightness, coughing, wheezing and shortness of breath, among which coughing is the most frequent and obvious. Traditionally, in diagnosis, a physician performs auscultation using stethoscope, listening for internal lung sound and analyzing patient’s condition. Computerized auscultation is widely researched and developed; is anticipated to soon overcome the subjectivity of those traditional methods.

State-of-the-art cough detection systems utilize audio signals together with various ASR-modified recognition techniques such as Hidden Markov Models (HMM), Artificial Neural Networks (ANN) and have reportedly achieved good performance [2, 3, 4]. However, they share the following three common limitations:

- Requirement of huge and well-prepared labeled database before being put into operation, which would consume time and effort and are not suitable for practical application.
- Heavy recognizer such as HMM or ANN are not suitable for implementation in portable devices.
- Audio signals are easily affected by environmental noises and therefore not reliable in realistic conditions.

In this paper, we introduce a novel method namely Semi-Supervised Tree SVM together with a contact microphone to address the issues above. Our system will be more suitable for portable devices, due to the following salient features: (1) Active learning system based on Semi-Supervised Tree SVM for higher accuracy with less training effort and less computational power required (2) Wearable contact microphone for clearer signals, less environmental noises.

Semi-supervision is well-known to help eliminate the need for huge time and effort spent on collecting and labelling data. However, how to determine a reliable sample to be added to current training dataset is still an open question. Derived from traditional one-against-one (OAO) multiclass SVM, our proposed confidence metric is targeted at this aspect. It scans all pairwise classifications in which one class is the decided class, compares their corresponding sample-to-hyperplane distances and picks up the smallest as confidence metric. We understand that smaller distance-to-hyperplane means higher chance for classification errors to occur and reinforce. However, we also understand that smaller distance-to-hyperplane means more new useful information to be collected. We implement a variable threshold, in order to keep this trade-off in control. Finally, to further enhance the system performance, we incorporate this Semi-Supervised learning strategy with Tree Structure to construct the novel method Semi-Supervised Tree SVM, an active learning system with improved discrimination capability. This method has been validated in a comprehensive database collected from ten subjects and has shown good improvement over conventional methods.

The organization of this paper is as follows: Section 2 introduces characteristics of cough sound and system overview. Section 3 describes Semi-supervised SVM and Semi-supervised Tree SVM. Section 4 presents experimental results. Section 5 concludes this work.

2. Cough sound characterization and signal processing

2.1. Acoustic model of coughing
A cough is defined as a reaction mechanism of the human body for clearing central airways from secretions and foreign bodies. A cough production system, like speech, can be modelled as a linear system of a glottal excitation and a transfer function (vocal tract or bone conduction). However, cough is different from speech in the following aspects: (1) Glottal excitation of cough is the main source of information as it is generated depending on the physiological condition of human; it could be a simple impulse for a normal cough or more complicated modulated impulse in the case of wheezing or asthma (2) Transfer function (vocal tract or bone conduction) is less varying as compared to speech and does not carry information on physiological state.
A cough model can be described by Figure 1 and the following equations.

In time domain:
\[ x(t) = e(t) \ast h(t) \]

In frequency domain:
\[ X(f) = E(f)H(f) \]

After taking the logs of powers and then DCT, the cepstral coefficients obtained are:
\[ c_X = c_E + c_H \]

Therefore, the cepstral analysis, which has been successfully applied in speech technology, is expected to be useful in this cough sound recognition.

2.2. System overview

An online cough detector must be able to capture sound events through a microphone and then perform some manipulation, feature extraction together with classification to determine the sound event that has been detected. A diagram of such a system is shown in Figure 3.

2.3. Signal processing models

As described by Figure 3 above, simple signal processing is performed on the signal first before the actual signal analysis is involved. This cannot be too complicated; otherwise it could become too time consuming for the system and result in slower than real-time operation. A few examples are given below: (1) DC Removal. This is applied when the signal has a DC offset which should first be removed; (2) Pre-emphasis. This can be in the form of a difference equation between successive signal samples, or some other signal emphasis designed to increase the signal-to-noise ratio; (3) Noise Reduction. This requires the most processing, and could take the form of a spectral filter, which measures the spectral distribution of background noise and then subtracts it from the signal. Windowing splits up the continuous signal into discrete frames, and is often used in conjunction with a predefined windowing function, such as the Hamming window. This improves the frequency distortion when the discrete Fourier transform of the frame is taken during event detection and feature extraction.

2.4. Detection

A sound event detector is used to detect the start and end points of a sound event in the audio signal. A common method involves calculating the power in the frequency spectrum in each window, and when a large change in power occurs, it corresponds to the start of the sound event.

3. Semi-supervised Tree SVM

3.1. Semi-supervised SVM and the confidence metric

Initially, one conventional SVM is constructed from a small set of training data. The idea is to use this SVM, while classifying unlabeled samples, at the same time, collect some informative samples and add them to the current training dataset. One iteration consists of 3 phases: (1) training SVM (2) classifying unlabeled samples and (3) adding chosen samples, to prepare for re-training in next round. However, uncontrolled classification errors and letting those errors reinforce themselves after iterations. A confidence metric is employed to limit selection to reliable samples only and ignore the rest. We choose to experiment and restrict our discussion to one-against-one, because of its superior performance over one-against-all. In OAO, all possible pair-wise classifications among the input classes are considered and final output is decided by the maxwins voting strategy. Our confidence metric is built upon this OAO method. Among all pair-wise classifications in which one class is the decided class, we consider their corresponding sample-to-hyperplane distances and choose the smallest as the confidence metric. We reason that smaller distance means the samples are nearer to separating hyperplanes, which may potentially help to bring in more useful information. However, smaller distance also mean higher chances of collecting classification errors and letting those errors reinforce themselves after iterations. We implement a variable threshold, in order to keep this trade-off in control. An example in Figure 4 will help to illustrate the idea. Note that in this case, Cough is the decided class and the distance to Cough/Artifact hyperplane is chosen as confidence metric.
3.2. Semi-supervised Tree SVM

In SVM multi-class problem, given \( n \) as the number of classes, OAO requires up to \( C(n, 2) = \frac{n(n-1)}{2} \) hyperplanes. The Tree structure, which requires only \( (n-1) \) for training and \( \log_2 n \) for testing [5], is a good candidate for reducing that heavy computation cost of OAO while still manage to maintain comparable performance. We propose a new method to construct Tree SVM, making use of FLD, a discriminating ability measurement between two classes [6]. We reason that it would enhance system classification performance (a higher FLD means that a particular grouping is highly discriminative). Such a design algorithm is described below.

Algorithm 1 Tree design

Require: Training data \( \{(x, y)\} \)

\[\text{if } \text{number of classes} = 2 \text{ then} \]

\[\text{Build the simple tree, nodes from those 2 classes} \]

\[\text{else} \]

1. Search all binary grouping possibilities at this layer
2. Compute FLD for each groupings based on training data \( (x, y) \)
3. Choose the grouping with highest FLD
4. Construct the left node and right node with the chosen binary grouping
5. Recursively go to left node
6. Recursively go to right node

end if

return \( f = \) the tree SVM classifier

In this particular case, the 3 classes will generate 3 combinations, listed as in Table 1.

Table 1: Fisher Linear Discriminant results.

<table>
<thead>
<tr>
<th>Class</th>
<th>Artifact/Non-artifact</th>
<th>Cough/Non-cough</th>
<th>Speech/Non-speech</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0238</td>
<td>0.0465</td>
<td>0.0269</td>
</tr>
</tbody>
</table>

Figure 5 shows the full picture of our Tree SVM, after being built by the proposed method. It consists of two sub-classifiers: (1) to separate Artifact and Non-Artifact samples (2) to separate Cough and Speech samples. In comparison with OAO SVM, the number of bi-class classifications it has to deal with is lesser, reduced from \( C(3, 2) = 3 \) to \( (n-1) = 2 \). Thus, computational power and processing time can be saved, especially when system is dealing with long feature vectors and large training set.

We then replace OAO SVM of the conventional Semi-supervised strategy by this Tree SVM. In each iteration, this Tree SVM also sequentially goes through 3 phases: (1) training (2) performing classification (3) adding unlabeled data. We name this Semi-supervised Tree SVM.

Figure 6: Semi-supervised Tree SVM.

Selection of reliable samples is governed by the same mechanism: using confidence metric to filter out those unwanted samples. Because of the difference in the classifier’s structure here compared with OAO SVM, choice of confidence metric is slightly modified as below.

Algorithm 2 Confidence metric decision

Require: testing data point \( (x, y) \)

\[\text{if } y = \text{Artifact} \text{ then} \]

\[d = \text{distance to Artifact/Non-Artifact hyperplane} \]

\[\text{else} \]

\[d = \text{distance to Cough/Speech hyperplane} \]

end if

return \( d \)

3.3. Prototype

A user-friendly application is implemented in Visual C# with two modes of operation: on-line and off-line. In on-line mode, with input signal is fed directly from microphone, events are detected and classified continuously. In off-line mode, system loads a pre-stored audio file, detects events stored inside and classifies those found events. In each mode, program can run with either learning strategies (1) active learning - by Semi-supervised Tree SVM (2) passive learning - by conventional OAO SVM.

Figure 7: Screenshot of Cough Detector application.

4. Experiments

4.1. Databases

Our database is recorded on ten subjects. Each subject is asked to produce at least 150 samples including: 50 coughs, 50 artifacts and 50 speeches. This would ensure that most variations of signals are well captured. Recording was done in relatively silent environments: lab, home, office. One contact-microphone is attached on the outer skin of subject’s trachea.
where coughing sound is the most audible. The database consists of three single classes: Cough, Artifact and Speech, simulating the real working environment of system. All cough sounds are voluntary coughs. Artifact consists of sound generated from user’s daily activities including: drinking, throat clearing, laughing. Speech are user’s pronunciations of number {1,2,3,4,5}, alphabet characters {a,b,c,d,e,x,y,z}. All are recorded and sampled at the frequency of 11.025 KHz, stored in WAV format. Training samples are selected randomly from the testing database; the size of training database is kept small, only at a fraction of total testing data. The purpose is to provide system with little initial information; leaving space for it to demonstrate the self-learning ability through the discussed semi-supervised mechanism.

<table>
<thead>
<tr>
<th>Class</th>
<th>Training database</th>
<th>Testing database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cough</td>
<td>80</td>
<td>1233</td>
</tr>
<tr>
<td>Artifact</td>
<td>100</td>
<td>745</td>
</tr>
<tr>
<td>Speech</td>
<td>80</td>
<td>1011</td>
</tr>
</tbody>
</table>

**4.2. Signal processing details**

In this experiment, system is set up with a linear kernel SVM and the following MFCC configuration, as in Table 3. Files are pre-segmented beforehand. MFCC configuration is optimized through experiments.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of MFCC coefficients</td>
<td>20</td>
</tr>
<tr>
<td>Number of filters</td>
<td>54</td>
</tr>
<tr>
<td>Window length</td>
<td>256</td>
</tr>
<tr>
<td>Window increment</td>
<td>128</td>
</tr>
<tr>
<td>Extra coefficients</td>
<td>zeroth + energy + delta</td>
</tr>
<tr>
<td>Feature vector dimension</td>
<td>44</td>
</tr>
</tbody>
</table>

**5. Conclusions**

In this paper, we have proposed a novel method combining Semi-supervised SVM and Tree SVM to detect and classify coughs among other acoustic signals. With Semi-supervised Tree SVM, it’s now more feasible to design a new home-based cough monitoring system that has self-adaptability and requires lesser computational power.

**6. References**