Detection of Whistler Waves

Evan Jeng

December 15, 2010

1 Introduction

A whistler wave is a very low frequency electromagnetic wave generated by lightning discharges. As the electromagnetic wave travels through the plasma environments of the ionosphere and magnetosphere, it undergoes dispersion of a several thousand kilohertz. Because the propagation velocity decreases with frequency, the lower frequency components of the wave arrive later than the higher frequency components. Thus, a radio station picking up a whistler wave would detect a signal that has decreasing frequency in time. Since the frequency range of the whistler waves lies within the human audible range, it would be perceived as a descending tone if converted into audio, hence its name. A whistler wave bounces back and forth across opposite sides of the planet, and the particular hop number and direction is indicated by a number and a sign. A 0+ whistler indicates a wave on its first upward path, whereas a 1- whistler indicates a wave on its downward path after its first reflection. In this project, we are interested in the automatic detection of the presence of 0+ whistler waves from radio data collected by the DEMETER (Detection of Electro-Magnetic Emissions Transmitted from Earthquake Regions) satellite.

2 Satellite Data

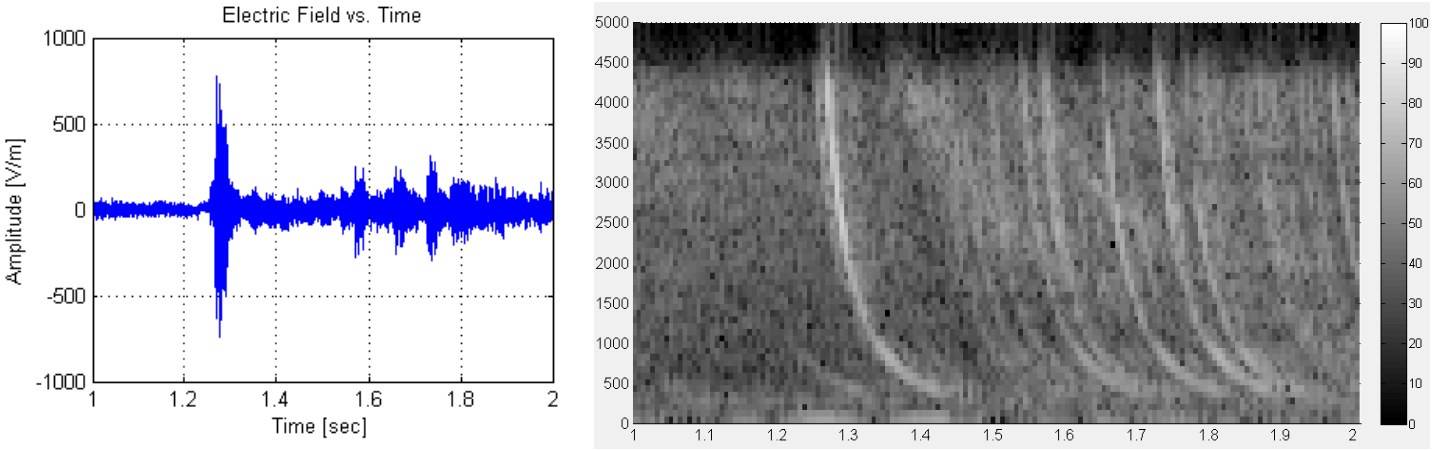

Figure 1. A one second interval of the electric field readings a) in time domain. b) as a spectrogram

The raw data gathered by DEMETER consists of time series of scalar-valued electric field measurements taken from various time spans during the satellite's orbit. The sampling frequency is 10kHz, which sets the Nyquist frequency just above the maximum whistler frequency. A one second sample of the time-domain electric field signal is shown in Figure 1a. The time-domain signal gives us amplitude as a function of time, which is not too helpful. Instead, we take a spectrogram of the data, which reveals the frequency content of the signal as a function of time. An FFT length of 128 with an overlap of 64 points

between frames was used to generate the spectrogram shown in Figure 1b. As we can see, the whistler waves are now easily identified as the white "L-shaped" curves in the image.

While identifying whistler waves in the spectrogram is a rather straightforward task for a human, how we would teach a machine to automatically detect whistlers is not immediately obvious. If we take an image recognition approach, major challenges to overcome seem to be the lack of a smooth gradient field that occur in normal images that would facilitate edge detection, poor signal to noise ratio, blurred boundaries for closely spaced whistlers, disappearance and reappearance of curve segments, as well as interference by extraneous signals.

3 Previous Attempts

Prior to this work, an attempt at automatic whistler detection by Stanford's VLF lab was done under the strong assumption that the background noises (i.e. non-signal regions) are independently and identically distributed. Given that the shape and duration of the whistlers don't vary much from whistler to whistler, they created a 2D filter shown in Figure 2 with $+\alpha$ for the white grids and $-\beta$ for the black grids, such that the sum of the values in the filter equals zero. The idea is that the expected filter output is zero when applied to a frame consisting of only noise. For a frame containing a whistler wave, the filter output would have a large positive value since it would lie in the white region of the frame. Classification would then involve either a constant thresholding or adaptive thresholding.

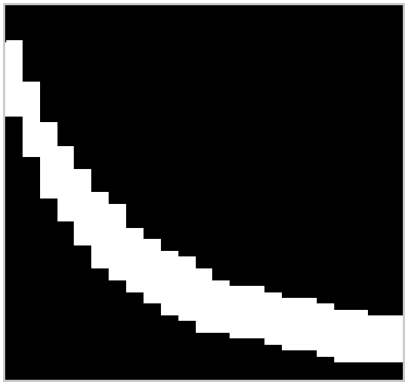

Figure 2. Filter for convolution with spectrogram.

In practice, this approach does not work very well for a number of reasons. First, the assumption that the background noise has zero mean is largely invalid. If we look at the spectrogram in Figure 1b, we see plenty of stray signals and other whistlers in the black 'non-signal' region of the filter. Unless a whistler exists in isolation, the presence of extraneous signals will result in destructive interference, leading to false negatives. On the other hand, strong localized non-whistlerlike pulse passing through the white grids in the filter can lead to large outputs values and false positives, making this approach a poor selector for shape.

4 Our Approach

In this work, we tackle the problem using support vector machines. Our goal was to improve the accuracy of classification by focusing on 'shape' that characterizes the whistlers, which is not well captured by the work described in the previous section.

4.1 Training Data

We were able to hand label around 240 whistlers in the available data. We observed that the whistlers have approximately the same duration of 0.2 seconds. A spectrogram 'frame' containing a whistler would then be around 65 x 24 points (spectrogram frequency bins x time steps), which corresponds to 1560 points per frame. Given that we only have around 240 whistlers in the data, we will run into serious issues

with high variance if we use every point in the frame as a feature. Thus, some preprocessing of the data is necessary before we start training our support vector machine.

4.2 Preprocessing/Filtering

Our first step in reducing the number of features is discarding the data in the 'non-signal' region of the frame. This is done simply by applying a binary mask with 1 for points in the signal region and 0 for those in the non-signal regions to each frame. We will discuss how we determine 'signal' and 'non-signal' regions in the next section, but for now suppose the allowed signal region at each time step spans 10 frequency bins. This would already take us to 32 x 10 = 320 features, down from 1560. While this is a significant improvement, we can do even better if we only kept the peak frequency bin within the signal region at each time-step, which ideally yields the trajectory of the whistler curve. This way, we have only one value per time step, yielding a basic feature set of around 32.

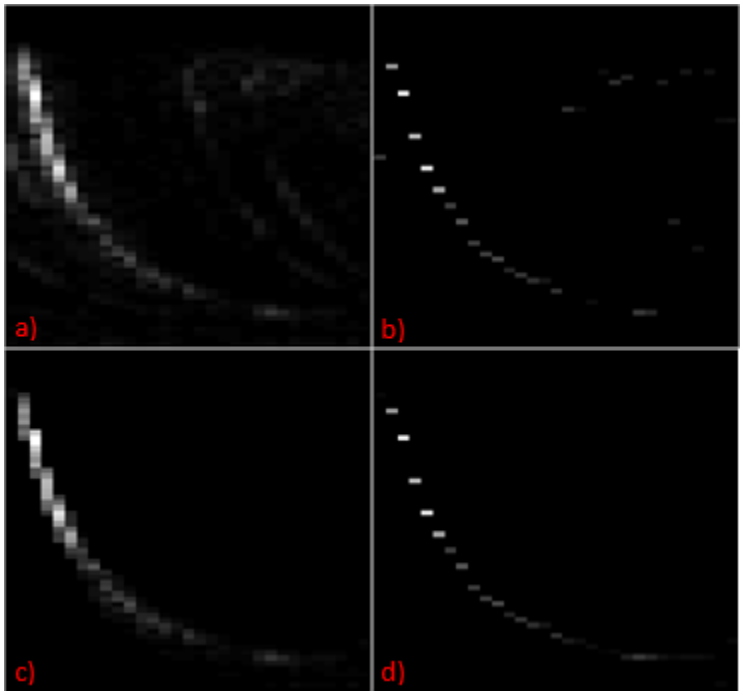

Figure 3. A spectrogram frame containing a whistler. a) raw spectrogram image. b) keeping only the peak intensity frequency bin in each column. c) applying binary mask. d) keeping the maximum frequency in each column after applying binary mask.

As can be seen in Figure 3, this preprocessing and feature reduction approach has the potential to work very well. By using the index of the frequency bin as a feature, we have the added benefit of invariance to intensity scaling. Since the signal strength of whistlers vary, we want the support vector machine to discriminate more on the structural characteristics of the trajectory, and not the absolute intensities of individual points. While this idea is good in principle, our filter does not always yield perfectly nice curves like the example in Figure 3. Sometimes extraneous signals within the defined 'signal' region dominate resulting in sharp discontinuities in the curve. However, we press forward with this approach, hoping that given enough 'good' data points within each training sample, the support vector machine will be able to make the correct decision.

4.3 Defining the Signal Region

Let N_f and N_t be the number of frequency bins and time steps, respectively, in a frame. Let $f^{*(i)} \in \mathbb{R}^{N_t}$ be a vector containing the indices of peak frequency bins within the chosen signal region for a given frame i. We want to find $\hat{f}, \Delta\hat{f} \in \mathbb{R}^{N_x}$ such that $\hat{f} \pm \Delta\hat{f}$ set a good upper and lower frequency bound on the signal region at each time step. Let $A(f', t)$be the intensity of frequency bin f' at time offset t. Our algorithm is as follows:

Initialize $\hat{f}, \Delta\hat{f}$ arbitrarily with rough approximation

Repeat until convergence:

$$\forall i \in \{1, \dots, M\}, \forall t \in \{1, \dots, N_t\} \qquad f_t^{*(i)} \coloneqq \arg\max_{f' \in \{\hat{f}-\Delta\hat{f}, \dots, \hat{f}+\Delta\hat{f}\}} A(f', t)^{(i)}$$

$$\forall i \in \{1, \dots, \mathrm{M}\} \qquad \hat{f}_t \coloneqq \frac{1}{M}\sum_i^M f_t^{*(i)}$$

$$\forall i \in \{1, \dots, \mathrm{M}\} \qquad \Delta\hat{f} \coloneqq \left\lceil k \cdot \sqrt{\frac{1}{M}\sum_i^M \left(f_t^{*(i)} - \hat{f}_t\right)^2} \right\rceil$$

, where M is the number of whistler-containing samples, and the constant k in the last line of the algorithm was a parameter to be optimized over in an outer loop given a particular feature set.

4.4 Feature Selection, Training, and Testing

We experimented with many combinations of features, including taking the first-order and second-order differences to capture the slope and curvature information. While each whistler may last around 32 time steps, it may not be optimal to include all 32 as features. For example, the tail end of the curve tends to have lower signal strength and is often dominated by noise or extraneous signals. We used Matlab and Liblinear to iterate over many combinations of features, using cross-validation to empirically determine a good feature set for our support vector machine.

5 Results

We settled on using 25 time steps and the absolute values of the first and second order differences of the peak indices as our feature set. Table 1 compares the performance of different features sets.

Feature Set	# of Features	False Positives	False Negatives	Overall Error
Stanford VLF	1	47.1%	47.5%	47.3%
$d0 + d1$	49	38.8%	24.2%	31.5%
$d1$	24	40.0%	24.6%	32.3%
$\lvert d1 \rvert$	24	30.4%	27.5%	29.0%
$\lvert d1 \rvert + d2$	47	40.0%	23.8%	31.9%
$\lvert d1 \rvert + \lvert d2 \rvert$	47	19.2%	14.2%	16.7%
$\lvert d2 \rvert$	23	12.9%	64.6%	38.8%

Table 1. Performance comparison of different feature sets. d0, d1, and d2 refer to 0th, 1st, and 2nd order differences of peak frequency bin indices.

As we can see, the best feature set used the absolute value of the first order and second order differences for a total of 47 features, achieving an overall error rate of 16.7%

6 Conclusion

Our support vector machine approach performs much better than the approach previously taken by the Stanford's VLF lab. In fact, on this data set, the previous approach does not do much better than random guessing, yielding a 47.3% error using the best thresholding value. Although we have achieved a pretty good error rate of 16.7%, we believe more training data is necessary, since using 480 training samples for 47 features is not a very good ratio. However, these initial trials results are promising. With more training data, future work could involve experimenting with more complicated features, such as local gradients around the signal region.