

Proposal For Final Project:

Machine Learning Detectors

Matt Daily

For my midterm project, I applied a simple SVM to the problem of detecting echos in sonar signals. The results were very promising on the simulated signals I generated. I am interested in expanding this work and comparing the performance of Support Vector Machines (SVMs), Artificial Neural Nets (ANNs), and Random Forests (RFs). In all cases I will compare the performance with the SVM trained according to the optimal values found in the first project: a radial kernel with parameter 0.1 and a cost of 100.

It is my end objective to apply these models, trained on simulated data, to real data of some sort. In order to achieve this, several efforts need to be undertaken as described here.

More Realistic Signal and Noise Model

I intend to develop a model that is a little more representative of a real world sonar. Specifically, this would involve:

1. Addition of a realistic beam pattern on transmit
2. Addition of representative surface and bottom reverberation processes, taken from published papers in the area of underwater acoustics
3. Development of a heuristic clutter model to account for the random noises one gets in a real environment. This would involve random clicks, false alarms, and transient changes in the background noise level due to things like waves
4. Development of a more realistic target model, that is, a target with multiple highlights

A lot of this work is already done as part of the simulation previously developed but needs to be exercised and checked before going forward.

More Advanced Detector Algorithm

Looking forward to applying this to a real world data set, a slightly more advanced detector needs to be implemented. Put another way, more features need to be made available to the model to compete with expertly developed systems. Thus I will augment the prior work as follows.

More Frequency Resolution

First I will increase the number of spectral values (bins) from 32 to 128 to make it more representative of a real world sonar. This will quadruple the size of the original SVM, potentially causing training run time problems. If this becomes the case, I will reduce the problem by training the original SVM across specific overlapping 32 bin expanses, in essence generating 7 different SVMs.

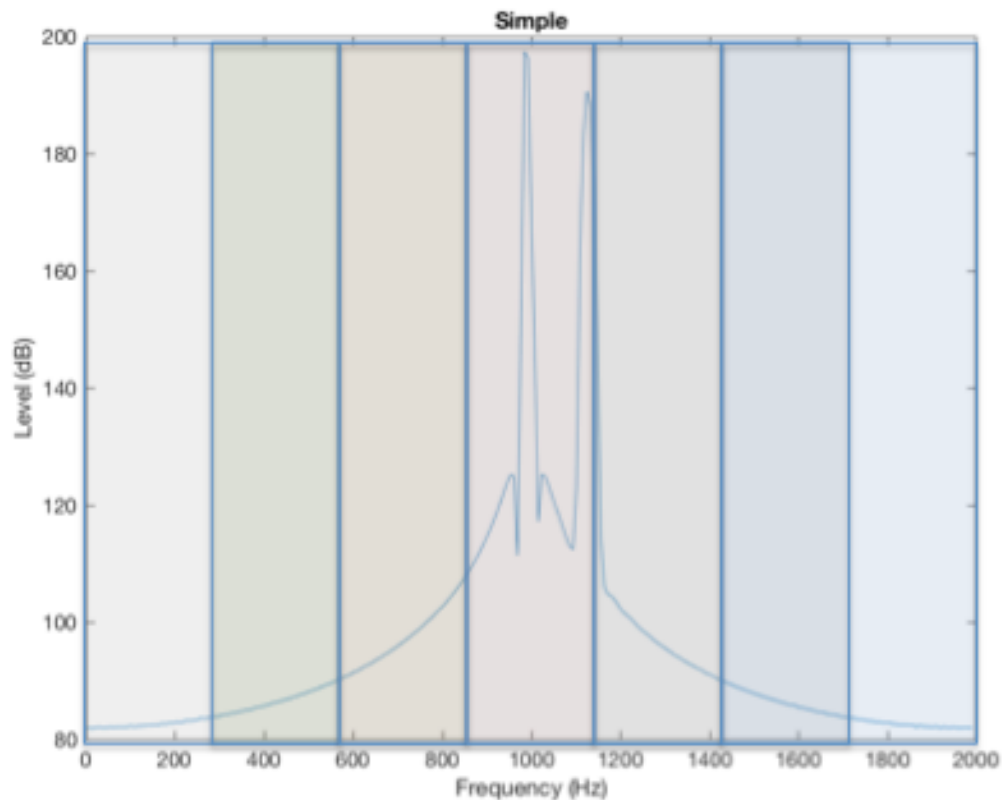


Figure 1: Picture of Frequency Extent of Templates

Improved Temporal Knowledge

In the prior work, the detection was based entirely on the spectrum in a single frame. Put another way, the only information available to the SVM was signals that came in *at the same time but at different frequencies*. Thus the SVM has no ability to leverage temporal changes in the signal from frame to frame. Specifically, it has no way to leverage the *transient nature* of the echo.

In this work, we will supply not just the current frame but both prior and ensuing frames to the ML algorithms. The number of surrounding frames will be explored. This represents the second dimension of the *template*.

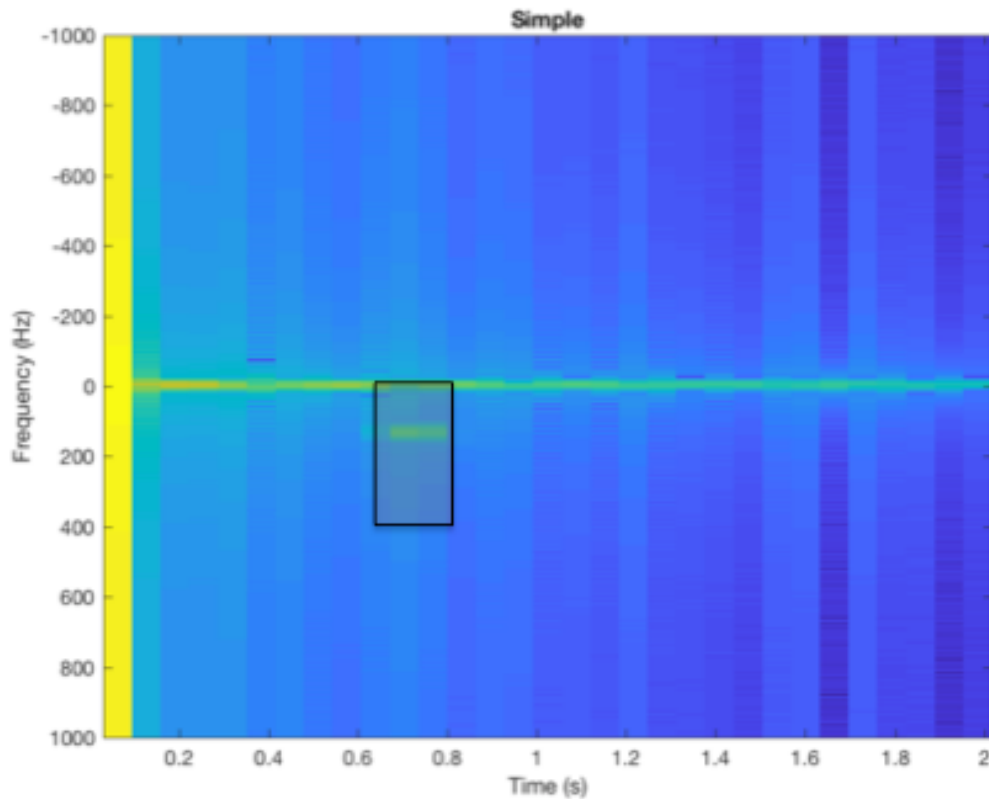


Figure 2: Representation of Detection Template

Generally speaking, I will use the term *template* to describe the exact data points used to train any specific algorithm. This is to avoid the term *kernel* which seems horribly overloaded at this time, though conceptually the template is similar to the kernel of a convolutional neural net, except that it is trained differently for each frequency sub-band.

The size of this template, along with some other detector specifics, will be a focus of this study. Specifically, the following table would be explored (run time permitted)

Feature	Parameter Limits
Template Width Time	3, 5, 7 frames
Template Width Frequency	8, 16, 32
Annotation List	{Range}, {Range, Speed}, {Range, Speed, Depth, Height}

Different Machine Learning Approaches

For each set of the parameters above, in addition to the original single-frame SVM, I would train at least three other models:

- An SVM using the full template,
- An ANN using the full template and
- A Random Forest using the full template

For each of these types, I would start with the 3x8 template size and do an exploration of various model parameters.

SVM – as before I would assess linear and radial kernels at different costs

ANN – The last layer is by definition a signal neuron, and the first is by definition the size of the template. I would try 2 and 3 layer implementations with hidden layer sizes equal to the input size, the input size/2, and the input size/8

RF – the number of estimators and the number of features (tree depth)

Based upon this initial exploration, the settings for each of the model types will be fixed for an exploration of the other detector parameters above.

Real World Application (As Time and Access Permits)

The parameter used in this work will reflect a real world sonar that I have experience with. It is my intention to use the best resulting models trained from this simulated data and apply them to real world data. How much of these results I will be able to present is a matter I am currently pursuing.