SVM Multivariate Veto

Zhen Wang, Junwei Cao and Xiaoge Wang, Tsingua University
Lindy Blackburn and Erik Katsavounidis, MIT

Abstract

We present the application of Support Vector Machines (SVM) toward identification of noise transients in gravitational-wave analysis. SVM is a multivariate classification method designed to handle a large number of parameters. Non-Gaussian noise transients dominate the background for gravitational-wave transient searches and limit the current sensitivity. SVM techniques allow us to use information from all available auxiliary non-GW channels together to reject gravitational-wave noise transients due to instrumental artifacts. We demonstrate the method and show performance on noise transients in gravitational-wave data from LIGO's fourth science run.

Figure 1. The interferometers of the LIGO/Virgo network are located in Livingston, LA; Hanford, WA; and Cascina, Italy.

1. Vetoes for Gravitational-Wave Searches

Current gravitational-wave searches are dominated by background – random coincidences of noise artifacts in the data. One technique to remove background events during data analysis involves making use of information from auxiliary channels to identify non-GW induced disturbances in the instruments and thus remove them. Support Vector Machines (SVM) provides a technique to handle the hundreds of degrees of freedom this auxiliary information provides and use it to efficiently classify noise events.

2. SVM (Support Vector Machine)

SVM is a statistical classification method which is good at classifying high-dimension sparse samples. In this case we use the SVM to separate noise events caused by instrumental artifacts from simulated signal samples resembling gravitational-wave observation. In general, the SVM works by finding a maximum-margin hyperplane to separate samples in a transformed space, defined by the kernel function, as shown in figure 2. The SVM algorithm has the following steps:

1. Map the non-linearly-separable sample set to a different high-dimension space. The map is defined by the kernel function, and the goal of the map is to make the classification linearly separable (the two classes are separated by a single hyperplane).
2. If the issue is still not linearly separable, slack variables ξ_i are introduced to quantify the misclassification of the data x_i .
3. Find the maximum-margin hyperplane to separate the samples in kernel space.

Formula 1. General form of hyperplane :

$$\vec{w} \bullet \vec{x} - b = 0$$

Formula 2. Find hyperplane and ξ_i which minimizes:

$$\|\vec{w}\|^2 + C\sum_i \vec{\xi}_i$$

Formula 3. subject to the constraint for all *i*:

$$c_i(\vec{w} \bullet \vec{x}_i - b) \geq 1 - \xi_i$$

$$c_i = \begin{cases} +1(\times) \\ -1(\mathrm{o}) \end{cases}, \quad \xi_i \geq 0$$

In our experiments, we use a Gaussian kernel function:

Formula 4. $$K(\vec{x}, \vec{y}) = \exp\{-\frac{|\vec{x} - \vec{y}|^2}{\sigma^2}\}$$

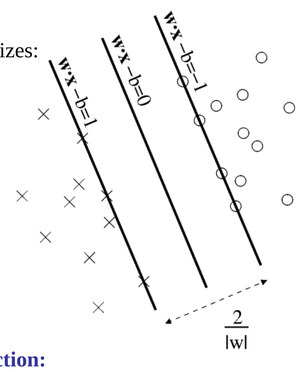
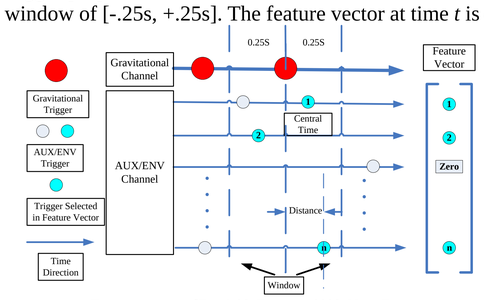


Figure 2. Maximum-margin Hyperplane.

3. Feature vector generated from auxiliary channels

For each noise or signal sample, we generated a feature vector for the SVM classification. Auxiliary channels measure non-GW degrees-of-freedom in the instruments. Noise transients identified in the auxiliary channels using the kleineWelle (KW) algorithm [3] indicate periods in time when there may be an instrumental disturbance. For any specific GPS time (given by an event identified in the gravitational-wave data, or a simulated event), we generate a single feature vector containing the KW trigger properties of the closest noise transients identified in auxiliary channels within a window of [-.25s, +.25s]. The feature vector at time *t* is denoted as *V(t)*.



Trigger property=[Freq/100, Energy*10000, NormEnergy, Npts, Significance, Distance]

The feature vector for one sample contains properties of all the nearby auxiliary channel triggers such as:

1) **dt**, the time difference between the sample time and the auxiliary channel trigger time
2) **frequency** of the auxiliary channel trigger
3) **signal strength** and statistical significance of the auxiliary channel trigger

Figure 3. Sample feature vectors for noise events are generated by collecting auxiliary channel trigger properties near the time of the GW noise trigger.

4. SVM training and testing samples

The SVM is used to classify events identified in the gravitational-wave data in a single instrument as either "noise" or "signal". To generate the appropriate separating hyperplane, a set of training samples for both populations is needed. Noise samples are produced by generating feature vectors about real transients identified in gravitational-wave data (the overwhelming majority of non-coincidence gravitational-wave channel triggers are noise). Signal samples are produced by generating feature vectors about random times, simulating a random population of true signals. A separate set of testing samples is generated for measuring the performance of the trained SVM.

Noise Samples:

Noise samples generated by running kleineWelle, a wavelet-based excess power method, on the gravitational-wave data from a single instrument. It is assumed that the overwhelming majority of these non-coincident triggers are noise fluctuations and not true gravitational waves.

Signal Samples:

Signal samples are meant to resemble true gravitational-waves. Because we only use the single trigger central time from the gravitational-wave data to build our samples, we approximately model the signal population as a uniformly distributed set of random times within our analysis intervals. Separate training and evaluation sample feature vectors are built from auxiliary channel triggers about these times, with the exception that for training purposes, a feature vector must not be zero (must contain at least one nearby auxiliary channel noise trigger). Zero feature vectors are automatically classified as signal, and provide no useful training information.

5. Experimental data and configuration

We use the LIGO S4 data from GPS 793152013s to 793756813s (first 7 days beginning Feb 23, 2005) from each of the three interferometric detectors: H1, H2, and L1. We use a KW significance threshold of 35 for all GW triggers to select moderately loud noise transients. Additional thresholds, shown below, are placed on the auxiliary channel triggers to ensure no hardware injections are vetoed. This is a safety requirement to make sure the auxiliary triggers we use for vetoes do not correspond to true gravitational waves.

s4_h1_asac	15
s4_h1_michctrl	15
s4_h1_pobi	28
s4_h1_pobq	11
s4_h1_prcctrl	25
s4_h1_radiolvea	12
s4_h1_refli	0
s4_h1_reflq	20
s4_h1_spobmon	0
s4_h1_spobi	0
s4_h1_wfs1qp	20
s4_h1_wfs2ip	11
s4_h1_wfs2qp	14
s4_h1_wfs3ip	0
s4_h1_wfs4ip	12
s4_h1_wfs5ip	12
s4_h1_bsp	0
s4_h1_bsv	0

Figure 4.a. H1 auxiliary channels and significance threshold

s4_h2_asac	25
s4_h2_michctrl	12
s4_h2_pobi	13
s4_h2_pobq	12
s4_h2_prcctrl	12
s4_h2_radiolvea	12
s4_h2_refli	20
s4_h2_reflq	0
s4_h2_wfs1qp	25
s4_h2_wfs2ip	0
s4_h2_wfs2qp	75
s4_h2_wfs3ip	11
s4_h2_wfs4ip	18
s4_h2_wfs5ip	14
s4_h2_bsp	0
s4_h2_bsy	11

Figure 4.b. H2 auxiliary channels and significance threshold

s4_l1_asac	0
s4_l1_michctrl	25
s4_l1_pobi	12
s4_l1_pobq	12
s4_l1_prcctrl	0
s4_l1_refli	14
s4_l1_reflq	12
s4_l1_spobmon	24
s4_l1_spobi	24
s4_l1_wfs1qp	20
s4_l1_wfs2ip	0
s4_l1_wfs2qp	25
s4_l1_wfs3ip	17
s4_l1_wfs4ip	12
s4_l1_wfs5ip	0
s4_l1_bsp	0
s4_l1_bsy	0

Figure 4.c.L1 auxiliary channels and significance threshold

We use 400 signal samples and 400 noise samples to train the SVM classifier and over 1000 samples of each to test performance of the SVM. The dimensions of the feature vectors for H1, H2 and L1 are 108, 96 and 102 respectively.

6. Veto performance metrics

Efficiency: fraction (%) of GW triggers rejected.

For the SVM classifier, veto efficiency is measured by the fraction the noise samples (generated from GW data) which are correctly classified as noise.

Dead-time: fraction (%) of live-time lost due to veto application.

Because the SVM does not construct explicit veto intervals, we use an alternative definition of dead-time: the probability of dismissing a real event as noise. For the SVM, dead-time is measured as the fraction of random times (signal samples) which are misclassified as noise.

7. Performance of the SVM classifier

We can modify the weight in the object function (described in formula 2) to give a greater emphasis to positive noise identifications (increase efficiency) or minimize false rejection of simulated signals (reduce dead-time). The efficiency vs. dead-time measurements at many different tunings are shown below.

Figure 5.a. Veto efficiency vs. dead-time for H1 noise events

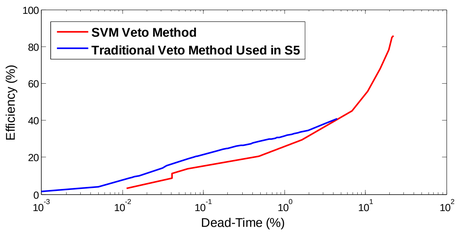


Figure 5.b. Veto efficiency vs. dead-time for H2 noise events

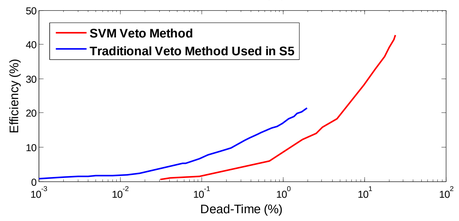
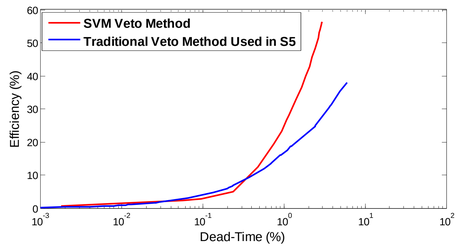


Figure 5.c. Veto efficiency vs. dead-time for L1 noise events



Shown here is a comparison with a classical fixed-window veto approach using the same auxiliary channel triggers and a hierarchal veto choice optimization scheme similar to that used by the LIGO S5 burst search [4]. The traditional single-channel fixed-window approach performs well at low dead-time, while the SVM has better maximum veto efficiency (where there are simply no more statistically significant single-channel vetoes to apply), indicating that the SVM may be particularly good at making use of information from many weak AUX channel triggers.

8. Future work

We will continue to mature the algorithm, testing different feature vector parameters, training samples, and kernel functions, and we plan to apply new multivariate veto classification schemes such as the SVM to current S6 data from the LIGO/Virgo network.

9. Acknowledgements

This work is supported by National Science Foundation of China (grant No. 60803017) and Ministry of Science and Technology of China under the national 863 high-tech R&D program (grants No. 2008AA01Z118 and No. 2008BAH32B03).

LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation and operates under cooperative agreement PHY-0107417. This paper has LIGO Document Number LIGO-G0901072.

10. Notes and references

[1] *A Tutorial on Support Vector Machines for Pattern Recognition, Christopher J. C. Burges. Kluwer Academic Publishers, Boston, 1998.*

[2] *Learning and Soft Computing — Support Vector Machines, Neural Networks, Fuzzy Logic Systems, Vojislav Kecman, The MIT Press, Cambridge, MA, 2001.*

[3] KleineWelle technical documentation, *http://www.ligo.caltech.edu/docs/T/T060221-00.pdf*

[4] LIGO Scientific Collaboration, *Search for gravitational-wave bursts in the first year of the fifth LIGO science run*, Phys.Rev.D80:102001 (2009)

Software:

LIBSVM: is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM).