Unsupervised matrix and tensor factorization for LIGO glitch identification using auxiliary channels

Rutuja Gurav[1], **Barry C. Barish** [2], **Gabriele Vajente**[3], **Evangelos E. Papalexakis** [1]
[1]Department of Computer Science and Engineering, University of California Riverside.
[2]Department of Physics and Astronomy, University of California Riverside.
[3]LIGO, California Institute of Technology.

Abstract

Glitches are non-Gaussian noise transients that plague the ground-based gravitational wave detectors like the Laser Interferometer Gravitational-Wave Observatory (LIGO). As they appear in the detector's main channel, they can mask or mimic real gravitational wave signals resulting in false alarms in the detection pipelines. Given their high rate of occurrence compared to astrophysical signals, it is vital to examine these glitches and probe their origin in the detector's environment and instruments. LIGO maintains $O(10^5)$ auxiliary electronics channels to monitor the detector's state, some of which may bear witness to these noise transients. Thus, automatically correlating glitches in the main channel with loud transient signals in these auxiliary channels can aid LIGO instrument specialists to find and fix sources of these noise transients and improve the detector. In this paper, we present a *proof-of-concept* application of Non-negative Matrix Factorization (NMF) and CP/PARAFAC tensor decomposition for co-clustering glitches in auxiliary channels space and find candidate channels that witness presence of glitches in the main channel.

1 Introduction

The ground-based gravitational-wave (GW) detectors like advanced LIGO (Aasi et al. 2015) have successfully reached the state-of-the-art sensitivity needed to detect astrophysical signals. Given their exquisite sensitivity, these detectors are plagued by various sources of terrestrial noise transients which affect the searches for GWs. Glitches are non-Gaussian noise transients appearing in the main channel of the detector which measures the amount of strain, $h(t)$, produced by a passing gravitational wave. Their origins are environmental and instrumental in nature. These noise transients stand out from the expected stationary Gaussian noise as transient power excesses which trigger search pipelines leading to false alarms and, along with the stationary noise, limit the detector's sensitivity. Hence, glitch characterization is a crucial problem and has been studied at LIGO in two important ways viz. glitch's morphology in the main channel and coincident glitches across auxiliary channels. Glitches have complex morphologies and vary in duration and frequency. Therefore, LIGO scientists have been studying the glitches based on the morphological differences evident in time-frequency spectrogram representations of these glitches. Several different classes of glitches have been identified (Zevin et al. 2017) and given their diversity and abundance, machine learning approaches have been used to automatically classify glitch spectrograms into distinct groups using image classification techniques (George, Shen, and Huerta 2017).

Apart from the main channel which measures the strain $h(t)$, LIGO maintains $O(10^5)$ auxiliary channels to record the state of the instrument and its environment using a wide variety of sensors, some of which are used in calibration of the detector. Some of these auxiliary channels may witness noise sources which couple to the main channel and produce a transient power excess causing a glitch. Finding correlations between glitches in the main channel and excess power events in the auxiliary channels is therefore useful in determining the origins of these glitches. This paper focuses on the task of *glitch analysis using auxiliary channels*. There have been several efforts at LIGO that use these auxiliary channels for glitch analysis. Notably, iDQ (Essick et al. 2020) is a low-latency glitch prediction pipeline which uses auxiliary channels information to train a binary classifier to compute the probability of presence of glitches in the main channel. Similarly, (Cavaglia, Staats, and Gill 2018) use auxiliary channel information and binary classification as a tool to study 2 sets of well-understood glitches from the first two observing runs of LIGO. They find a set of channels that witness these glitches and infer the probable mechanical couplings which produce them. Our contribution is an unsupervised exploratory analysis of glitches in the auxiliary channels feature space using non-negative matrix factorization (NMF) and CP/PARAFAC tensor factorization.

2 Background

Noise coupling into the instrument's main channel(s) is a typical issue in big scientific experiments that push the limits of technology towards ground-breaking discoveries. LIGO's main data product is the strain, $h(t)$ - the relative change in the interferometer's arm lengths induced by a passing GW. The stretching and squeezing of spacetime by a passing GW

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

physically changes the interferometer's arm length which causes the laser beams to interfere leading to a power excess to occur and a signal is measured at the output photodiode. It is to be noted that transient terrestrial disturbances originating in the environment or the instrument can also cause a brief power excess and result in a signal being registered at the output photo-diode. This coupling of noise into the main channel, $h(t)$, is, in essence, a glitch. Auxiliary channels record the measurements from a wide variety of sensors that are deployed around the interferometer. The $O(10^5)$ recorded auxiliary channels include a lot of redundancy. LIGO instrument specialists use a subset of the auxiliary channels as *safe* channels to monitor and mitigate noise sources. LIGO data analysts also use these auxiliary channels as *veto generators* to remove time segments contaminated with glitches and generate clean segments of $h(t)$ that are used for GW searches. Thus, as described in (Essick, Blackburn, and Katsavounidis 2013) it is useful to determine important auxiliary channels which can act as *veto generators*. Given the large number of auxiliary channels only some of which may witness a glitch, it is conducive to use machine learning based techniques to find correlation between the main and auxiliary channel power excesses.

3 Analysis

The task at hand is to cluster glitches in a chosen time interval as per their auxiliary channels characteristics. In this paper, we formulate this task as a soft co-clustering which is discussed in brief in Section 3.3. As a result of co-clustering, we can obtain a small subset of important channels that are indicative of presence of glitches in $h(t)$.

$h(t)$	the main channel sensitive to GWs
A	*safe* auxiliary channels selected for analysis
$a_i(t)$	i^{th} auxiliary channel $\in A$
G	glitches in $h(t)$
t_g	time of a $h(t)$ glitch $\in G$
F	number of frequency bins
S	selected channels that witness glitches
$[(t_{start}), (t_{end})]$	time interval chosen for analysis

Table 1: Terminology

3.1 Data Collection

A trigger refers to a power excess in $h(t)$ and any $a(t)$ detected by an event trigger generator (ETG) like Omicron (Robinet 2016) which runs continually in real-time at LIGO during its operational runs. Ergo, glitches are simply loud triggers in $h(t)$ that are not astrophysical in origin. Omicron also infers properties of the generated triggers viz. GPS timestamp, signal-to-noise ratio (SNR), peak frequency, bandwidth, amplitude, duration, phase, etc. Thus, each trigger is represented by a list of its properties.

In this analysis, the threshold for loud triggers in $h(t)$ is set at peak SNR $\geq$ 7.5 following the example of Gravity Spy which only registers triggers with peak SNR $\geq$ 7.5 as glitches in their catalog. We use a list of safe auxiliary channels maintained by LIGO's Detector Characterization group.

We can do a systematic trigger collection inspired by (Cavaglia, Staats, and Gill 2018) as follows:

1. Obtain a set, G, of loud triggers, i.e. trigger with peak SNR $>$ 7.5, that occurred in $h(t)$ between t_{start} and t_{end}.
2. For all $t_g \in G$, find triggers from a set, A, of safe auxiliary channels in a window around each t_g.
   $window_{start} = t_g - (\alpha * \Delta t_g) - (\beta * \Delta t_g)$
   $window_{end} = t_g + ((1-\alpha) * \Delta t_g) + (\beta * \Delta t_g)$
   where, $t_g \in G$ is the peak SNR time of the glitch trigger. Δt_g is duration in seconds for which the glitch trigger lasted in $h(t)$. α, β are window parameters to center the glitch trigger around its peak SNR. In this analysis $\alpha = 0.5, \beta = 0$.

If no trigger is present in the window, a default null trigger is stored for that channel corresponding to that t_g. In case of multiple triggers in the window, we pick the loudest trigger. Note that we do not impose any SNR threshold on the auxiliary channels trigger selection, therefore the lower bound on SNR threshold is 5.5 which is the default provided by Omicron.

3.2 Feature Engineering

Matrix Construction: A feature matrix $M \in \mathbb{R}^{|G| \times |A|}$ encodes the presence or absence of a loud trigger in all auxiliary channels for every glitch in G weighted by peak SNR. Each $|A|$-length row of M corresponds to a glitch and each element $M_{i,j}$ is the peak SNR of i_{th} trigger in j_{th} auxiliary channel if trigger present, zero otherwise.
Tensor Construction: A tensor in this context is a multidimensional array. The tensor encodes - 1) presence or absence of a loud trigger in all auxiliary channel for every glitch in G weighted by peak SNR, 2) if present, the peak frequency of the trigger in an auxiliary channel. In order to encode these 2 pieces of information, we create a 3-mode tensor $\mathcal{X} \in \mathbb{R}^{|G| \times |A| \times |F|}$. Since peak frequency is a continuous quantity, we discretize it into frequency bins. Thus, each element $\mathcal{X}_{i,j,k}$ is the peak SNR of i_{th} trigger in j_{th} auxiliary channel having peak frequency that falls in the k_{th} frequency bin if trigger present, zero otherwise.

3.3 Co-clustering glitches and channels

Consider the traditional clustering problem where, a data matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ has m data-points (or observations) each of which has n features. Clustering methods, like *K-means* partition the data-points to discover subsets such that data-points in a cluster are *similar* to each other and *distinct* from data-points in other subsets. Co-clustering refers to simultaneous clustering of multiple modes of the data. In case of the 2 mode data matrix $\mathbf{D}$ described above, co-clustering partitions along the rows as well as the columns to find subsets. Co-clustering can be formulated as a multi-linear decomposition as described in (Papalexakis and Sidiropoulos 2011) such that $\mathbf{D}^{m \times n} \approx [\mathbf{R}^{m \times k}\mathbf{C}^{k \times n}]$ where k is referred to as the *rank* of the decomposition. In this paper, to co-cluster the 2-mode data matrix $\mathbf{M} \in \mathbb{R}^{|G| \times |A|}$, we use Non-negative Matrix Factorization and for the 3-mode data tensor $\mathcal{X} \in \mathbb{R}^{|G| \times |A| \times |F|}$, we use non-negative CP/PARAFAC factorization. The choice of NMF and non-negative CP/PARAFAC is dictated by the non-negative values (the SNRs of the triggers) in $\mathbf{M}$ and $\mathfrak{X}$. Imposing the non-negativity constraint can potentially yield more interpretable factors that prove useful in selected specific channels.

Non-negative Matrix Factorization (NMF): The formulation of NMF is stated as follows. Given $\mathbf{M} \in \mathbb{R}_{+}^{|G|\times|A|}$ and desired number of components (*rank*) $k << min(|G|, |A|)$, find $\mathbf{W} \in \mathbb{R}_{+}^{|G|\times k}$ and $\mathbf{H} \in \mathbb{R}_{+}^{k\times|A|}$ such that $\mathbf{M} \approx \mathbf{WH}$. In our case, each row of $\mathbf{W}$ and $\mathbf{H}^T$ is a k-length latent space representation of triggers and channels respectively. For computation of NMF, we used the implementation from Scikit-Learn library available for Python.

Non-negative Tensor Factorization: CP/PARAFAC tensor factorization of $\mathfrak{X} \in \mathbb{R}^{|G|\times|A|\times|F|}$ is expressed as a sum of outer products of R rank-1 tensors called components or factors. These components can be arranged as 3 factor matrices viz. $\mathbf{U}_{|G|\times r}$, $\mathbf{V}_{|A|\times r}$, and $\mathbf{W}_{|F|\times r}$ corresponding to each mode of $\mathfrak{X}$, each with r columns. Formally stated as, $\mathfrak{X} \approx \sum_{r=1}^{R} \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$, where $\mathbf{u}_r$, $\mathbf{v}_r$ and $\mathbf{w}_r$ are the r^{th} column in factor matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ respectively. The true rank R of the tensor is defined as the minimum number of rank-1 components required to exactly reconstruct the original tensor. However, a low-rank approximation of the tensor is of interest to our analysis since a low-rank approximation can capture latent patterns across the modes of the tensor. In our case, the factor matrix $\mathbf{U}_{|G|\times r}$ and $\mathbf{V}_{|G|\times r}$ hold the r-length latent space representations of the glitch triggers and channels respectively. For the computation of the CP/PARAFAC decomposition we used the implementation of the Alternating Least Squares method in Tensorly (Kossaifi et al. 2019). Additionally, we impose non-negativity constraint on the factors.

4 Results

Witnessing glitches in the main channel: For this analysis, we chose time intervals in the second half of the third observing run of LIGO (referred to as O3b by the community) (Tse et al. 2019). Specifically, the *training* interval spans from November 1 to November 4, 2019 and the *validation* interval spans from November 5 to November 6, 2019. There are 987 loud triggers in the *training* interval and 382 loud triggers in the *validation* interval at the LIGO Hanford observatory. We used 845 safe auxiliary channels obtained from a list maintained by the Detector Characterization group at LIGO. We perform NMF on $\mathbf{M}$ and CP/PARAFAC on $\mathfrak{X}$ constructed using loud triggers in the *training* interval and examine the co-clusters obtained. The latent space representations of the channels ($\mathbf{H}^T$ in NMF and $\mathbf{V}$ in CP/PARAFAC) are then used to obtain a small set of channels, S, to test as *witnesses* of occurrence of glitches during the *validation* interval. More specifically, each $|A|$-length vector in $\mathbf{H}^T$ or $\mathbf{V}$ encodes the *contribution* of subset of channels to the corresponding subset of glitches. We pick the channel corresponding to the maximum magnitude value in the vector and add it to S. For each channel $s_i \in S$, if there is a glitch in $h(t)$ and there is a corresponding trigger in s_i in a window around the glitch, we say s_i *witnessed* the glitch. For each channel $s_i \in S$, we count the number of glitches in $h(t)$ that it *witnessed* during the *validation* interval. Tables 2 and 3 show the percentage of glitches witnessed in the *validation* interval by channels selected using NMF and CP/PARAFAC with rank 20 respectively. In both NMF and CP/PARAFAC models across different ranks, the top channels witnessed ∼80% of the glitches that occurred in the *validation* interval.

channel	% glitches witnessed
H1-PEM-EY_MAG_EBAY_SEIRACK_X_DQ	85.86
H1-IMC-PZT_YAW_OUT_DQ	81.15
H1-PEM-EX_ADC_0_17_OUT_DQ	79.31
H1-IMC-MC2_YAW_OUT_DQ	78.53
H1-PEM-EX_MAG_EBAY_SEIRACK_X_DQ	77.22
H1-PEM-EX_ADC_0_19_OUT_DQ	76.96
H1-PEM-VAULT_MAG_1030X195Y_COIL_Y_DQ	74.86
H1-PEM-VAULT_MAG_1030X195Y_COIL_X_DQ	71.98
H1-PEM-CS_MAG_EBAY_SUSRACK_Y_DQ	60.73
H1-PEM-EX_MAG_EBAY_SUSRACK_Y_DQ	59.16

Table 2: Percentage of $h(t)$ glitches witnessed by a set of unique channels selected using NMF with rank 20 *(only channels that witnessed more than half the actual glitches are shown here)*

channel	% glitches witnessed
H1-PEM-EY_MAG_EBAY_SEIRACK_Z_DQ	87.43
H1-PEM-EY_MAG_EBAY_SEIRACK_X_DQ	85.86
H1-IMC-PZT_YAW_OUT_DQ	81.15
H1-IMC-PZT_PIT_OUT_DQ	80.62
H1-PEM-EX_ADC_0_17_OUT_DQ	79.31
H1-IMC-MC2_YAW_OUT_DQ	78.53
H1-PEM-EX_MAG_EBAY_SEIRACK_X_DQ	77.22
H1-PEM-VAULT_MAG_1030X195Y_COIL_Y_DQ	74.86
H1-PEM-VAULT_MAG_1030X195Y_COIL_X_DQ	71.98
H1-LSC-SRCL_OUT_DQ	64.13
H1-PEM-CS_MAG_EBAY_SUSRACK_Y_DQ	60.73

Table 3: Percentage of $h(t)$ glitches witnessed by a set of unique channels selected using CP/PARAFAC with rank 20 *(only channels that witnessed more than half the actual glitches are shown here)*

Clustering glitches: Since, the $|G|$-length vectors in $\mathbf{W}$ and $\mathbf{U}$ are latent representations of glitches and indicators of glitch clusters, given some ground-truth labels like the Gravity Spy catalog, we can examine whether we find homogeneous clusters for the different Gravity Spy classes in the auxiliary channels space. More specifically, we quantify *homogeneity* of the clusters found by NMF and CP/PARAFAC with respect to the Gravity Spy classes. To that effect, we count the number of occurrences of glitches belonging to each Gravity Spy class in the top k values of each $|G|$-length vector in $\mathbf{W}$ and $\mathbf{U}$ and aggregate this quantity across all factors. For example, homogeneity of a co-clustering instance of NMF with rank r is defined as $\frac{1}{r}\sum_{i=1}^{r}\frac{1}{N_i}$ where N_i is the number of unique labels in top k of ith factor i.e. column in $\mathbf{W}$. Here, k is adaptively chosen to be 90% of the norm of the factor. Thus if we find highly *homogeneous* auxiliary channels space clusters for the Gravity Spy classes, this quantity is close to 1.

Figures 1 shows the average *homogeneity* across different factorization ranks using NMF and non-negative CP/PARAFAC respectively. This preliminary analysis suggests that the overall low homogeneity is probably suggestive of the fact that not all Gravity Spy classes have a simple,

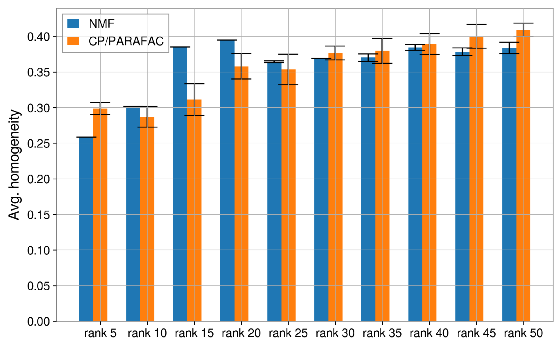

Figure 1: Average *homogeneity* across 10 runs w.r.t. Gravity Spy classes in *clusters* found by NMF and CP/PARAFAC across different ranks using *training* interval data.

consistent structure in the auxiliary channels space despite the visual shape similarities in $h(t)$ signal.

5 Conclusions and Future Directions

In this paper, we provide *proof-of-concept* LIGO glitch analysis using auxiliary channels information. We show that non-negative matrix factorization and CP/PARAFAC tensor factorization are able co-clusters glitches from a *training* time interval in the auxiliary channels space and produce a small subset of *safe* auxiliary channels some of which witnessed ∼80% of the glitches in a *validation* time interval.

Although this preliminary analysis shows the selected channels witness ∼80% of the glitches, as we mention in Section 3.1, we do not impose any threshold on auxiliary channels SNR. It is highly possible that there are spurious correlations between glitches in main channel and relatively low SNR triggers in the selected auxiliary channels. Although, this paper does not define them as such, using these selected channels as standalone *veto generators* results in high number of false positives at low SNR thresholds. This is crucial when determining whether a given channel is a good *veto generator* or not. For every trigger present in a selected *veto generator* channel candidate, we predict that there is a glitch in $h(t)$ and remove a segment of $h(t)$ to eliminate the glitch. The characteristics of a good *veto generator* channel are that it has high *efficiency* -number of actual $h(t)$ glitches predicted- and low *dead-time* -amount of data removed from $h(t)$. High number of false positives results in a large amount of *dead-time*.

We reserve further investigation on the selection of appropriate SNR thresholds for auxiliary channels to decrease the false positives for extensions of this work in the future.

6 Acknowledgements

The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. Research at UCR was supported by the National Science Foundation CDS&E Grant no. OAC-1808591 and by a UCR Regents Faculty Development Award.

References

[Aasi et al. 2015] Aasi, J.; Abbott, B.; Abbott, R.; Abbott, T.; Abernathy, M.; Ackley, K.; Adams, C.; Adams, T.; Addesso, P.; Adhikari, R.; et al. 2015. Advanced ligo. *Classical and quantum gravity* 32(7):074001.

[Cavaglia, Staats, and Gill 2018] Cavaglia, M.; Staats, K.; and Gill, T. 2018. Finding the origin of noise transients in ligo data with machine learning. *arXiv preprint arXiv:1812.05225*.

[Essick et al. 2020] Essick, R.; Godwin, P.; Hanna, C.; Blackburn, L.; and Katsavounidis, E. 2020. idq: Statistical inference of non-gaussian noise with auxiliary degrees of freedom in gravitational-wave detectors. *arXiv preprint arXiv:2005.12761*.

[Essick, Blackburn, and Katsavounidis 2013] Essick, R.; Blackburn, L.; and Katsavounidis, E. 2013. Optimizing vetoes for gravitational-wave transient searches. *Classical and Quantum Gravity* 30(15):155010.

[George, Shen, and Huerta 2017] George, D.; Shen, H.; and Huerta, E. 2017. Deep transfer learning: A new deep learning glitch classification method for advanced ligo. *arXiv preprint arXiv:1706.07446*.

[Kossaifi et al. 2019] Kossaifi, J.; Panagakis, Y.; Anandkumar, A.; and Pantic, M. 2019. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research* 20(1):925–930.

[Papalexakis and Sidiropoulos 2011] Papalexakis, E. E., and Sidiropoulos, N. D. 2011. Co-clustering as multilinear decomposition with sparse latent factors. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2064–2067. IEEE.

[Robinet 2016] Robinet, F. 2016. Omicron: an algorithm to detect and characterize transient events in gravitational-wave detectors.

[Tse et al. 2019] Tse, M.; Yu, H.; Kijbunchoo, N.; Fernandez-Galiana, A.; Dupej, P.; Barsotti, L.; Blair, C.; Brown, D.; Dwyer, S.; Effler, A.; et al. 2019. Quantum-enhanced advanced ligo detectors in the era of gravitational-wave astronomy. *Physical Review Letters* 123(23):231107.

[Zevin et al. 2017] Zevin, M.; Coughlin, S.; Bahaadini, S.; Besler, E.; Rohani, N.; Allen, S.; Cabero, M.; Crowston, K.; Katsaggelos, A. K.; Larson, S. L.; et al. 2017. Gravity spy: integrating advanced ligo detector characterization, machine learning, and citizen science. *Classical and Quantum Gravity* 34(6):064003.