# Detecting structured signals in radio telescope data using RKHS

**Russell Tsuchida**
Data61-CSIRO
Canberra, 2601 Australia
russell.tsuchida@data61.csiro.au

**Suk Yee Yong**
CSIRO Space & Astronomy
NSW, 2122 Australia
sukyee.yong@mq.edu.au

## Abstract

Astrophysical sources emit radio emissions that are detectable by radio telescopes. Due to the volume of data produced by radio telescopes, efficient computational methods for automatically detecting signals of interest are required. The most basic of these methods involves fitting a physical model of frequency dispersion to the observed signal, and flagging a detection if the dedispersed signal has high power. This method can successfully detect single pulses, but might miss detecting other interesting astronomical signals. We propose a method for dedispersion that does not use a physical model but instead uses a flexible element of a reproducing kernel Hilbert space (RKHS). Our method can outperform classical dedispersion on a benchmark of real and synthetic data consisting of signals that are pulse-like, physical, and non-physical origins.

## 1 Background

**Time-domain discoveries in astronomy** As radio waves travel from a source to a radio telescope, low frequency components of the signal experience a relative delay due to electron interference between the source and telescope. In a certain idealised setting, the time delay $\hat{t}(\nu)$ experienced by component with frequency $\nu$ is proportional to the line integral of the electron density $n_e$ between the source and observer divided by $\nu^2$ [Lorimer and Kramer, 2012, Equation 4.4]. That is,

$$\hat{t}(\nu) = K\nu^{-2} \underbrace{\int_L n_e(x)\, dx}_{\text{DM}}, \tag{1}$$

where $K$ is a physical constant, $L$ is the line connecting source to observer, and DM is called the dispersion measure. Assuming a step response at the source, this physical model can be used to describe the left side of Figure 1, but not the origin of the signal itself. One type of pulse signal of this nature that does not have an explanation are called fast radio bursts (FRBs) [Lorimer et al., 2007].

**Reproducing Kernel Hilbert Spaces** Given a set of points and a hypothesis class $\mathcal{H}$ of functions, a commonly encountered problem throughout quantitative sciences is to fit a function $f^* \in \mathcal{H}$ to the set of points. In machine learning, this is often performed with the hope that the fitted function generalises to a set of unseen points. Two popular choices for the hypothesis class $\mathcal{H}$ are neural networks and reproducing kernel Hilbert spaces (RKHS). An RKHS is a rich class of (infinite dimensional) functions with special structure that allow them to be manipulated using (finitely many) computer operations. To every RKHS $\mathcal{H}_k$ is associated a unique positive semidefinite kernel $k$ (and conversely), which describes the smoothness of the functions in the RKHS. Unlike neural networks, kernel methods can be easy to fit when appropriately regularised since they admit a representation of a simple form. We use a special case of Schölkopf et al. Theorem 1.

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

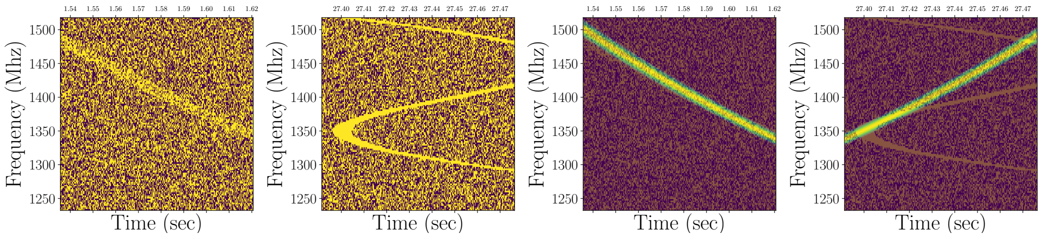

Figure 1: (Left) Two chunks $Y_c$ of simulated data. There are $n = 128$ discrete frequency bins ranging from $1.2$ GHz to $1.5$ GHz. The data represents a duration of $x$ seconds with $T = 100$ discrete time steps. Each element $y_{ij}$ of this matrix is either $0$ (purple) or $1$ (yellow). (Right) Overlay of best-fitting FRB using the physical model (1)–(3).

**Theorem 1.** *Let $\lambda > 0$ be a regularisation parameter, $\mathcal{X}$ a set and $L : (\mathcal{X} \times \mathbb{R}^2)^N \to \mathbb{R} \bigcup \{\infty\}$ an arbitrary loss function. Let $\mathcal{H}_k$ be an RKHS with kernel $k$. Then each minimiser $f \in \mathcal{H}_k$ of*

$$\sum_{i=1}^{N} L\Big(\big(x_i, y_i, f(x_i)\big)\Big) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_k}$$

*admits a representation of the form $f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$.*

When coupled with (strong) convexity of the regularised loss, such problems over RKHS can be solved by application of convex optimisers to find (unique) optimal representer coefficients $\{\alpha_i\}_{i=1}^{N}$.

## 2 Signal processing for astronomy data

**Dataset** The Single-dish PARKES for finding the uneXpected [SPARKESX; Yong et al., 2022] dataset is a publicly available[1] compilation of real and simulated high-time resolution observations of the Parkes (Murriyang) radio telescope. SPARKESX is designed as a data challenge to test different search methods and pipelines. SPARKESX labels and benchmark results with the standard pulsar search software, PRESTO, are provided for comparison with other methods.

SPARKESX consists of a range of artificially injected signals, including expected astrophysical signals (e.g., pulsars, FRBs, and stellar flares) and unexpected events (e.g., negatively dispersed pulses, splines, and steganography images). For further details on the types of signals generated, see Yong et al. [2022]. We consider the following event groups from SPARKESX: `simplepulse`, `known+rfi`, `unknown+rfi`, `combo+rfi`, and `real+combo`.

Let $Y$ denote an $n \times m$ binary matrix representing a datastream of 1-bit data. Here $n$ is the number of discrete frequency bins and $m$ is the number of discrete timesteps, which may be infinite for streaming data. We denote the $ij$th element of $Y$ by $y_{ij}$. We use $x_{ij} = (\nu_i, t_j)$ to denote the corresponding discrete frequency-time pair. Let $T$ denote the length of some time window and define the $c$th chunk $Y_c$ of $Y$ as the $n \times T$ submatrix $Y_c = Y_{\{:, Tc:T(c+1)\}}$. Representative data is shown in Figure 1. For our experiments, we only use the 1-bit single beam of the multibeam survey from SPARKESX. This single-bit is used as data of the form $Y$, and each chunk $Y_c$ is associated with a binary value $z_c$ indicating whether a signal of interest is present in the chunk. Each signal is also associated with attributes such as the amplitude $A$, width $W$ and dispersion measure DM (in the case of a pulse). Each dataset contains 51,200 seconds $\approx 14$ hours of data, except for `real+combo` which contains twice this duration. In total, this represents about 3.6 days worth of data, with a total memory footprint of 14.36GB. Roughly 10 % of the data contains an event.

**Dedispersion using physical models** For a signal that has undergone dispersion, pixels that lie close to the parametric curve $\big(\hat{t}(\nu) - t_0, \nu\big)$ in the $t - \nu$ plane are more likely to be 1 than 0, where $t_0$ is some offset indicating the start datum of the FRB. This may be modelled by setting the probability $p_{ij}$ that the $ij$th pixel is 1 to

$$p_{ij} = \max\left\{aw\left(t_j - \hat{t}(\nu_i)\right), b\right\}, \tag{2}$$

[1] `https://doi.org/10.25919/fd4f-0g20`.

where $0 < b < a < 1$ are hyperparameters and $w$ is some window function, upper bounded by 1 and decaying to zero as it's argument goes to $\pm\infty$. For example, Gaussian and rectangular windows are $w\left(z\right) = \exp(-z^2/\ell^2)$ and $w\left(z\right) = \Theta\left(|z| - \ell\right)$, where $0 < \ell < \infty$ and $\Theta$ is the step function.

Given a candidate DM and $t_0$ with fixed hyperparameters $\ell$, $a$ and $b$, the process of dedispersion is to adjust each frequency band in the signal for the time delay $\hat{t}(\nu)$ experienced by that frequency band. After dedespersion, the total power of the signal at the source can be computed by summing the squared amplitudes of the signals in each frequency band. If this power is large, it indicates that the observed signal can be explained by a powerful pulse with the candidate DM and datum $t_0$. The power of the signal can be written in terms of an optimisation problem,

$$P_c = \max_{\text{DM},t_0} \sum_{i=1}^{n} \sum_{j=Tc}^{T(c+1)} \big(y_{ij} p_{ij}\big)^2. \tag{3}$$

The right two plots in Figure 1 shows $p_{ij}$ using the optimal DM and $t_0$ corresponding with the solution of (3). To solve the optimisaton problem, a fine-grained grid search can be used over the space of DM and $t_0$. Software such as PRESTO can perform this search, as reported by Yong et al..

**Detecting signals using an RKHS** The physical model (1)–(3) is a useful inductive bias to search for dispersed pulses, such as FRBs. However, more complicated signals, such as that depicted in the second column of Figure 1, cannot be easily detected using this model. We propose to alter the process of dedispersion by using an element of an RKHS instead of the hypothesis class of dispersed pulses. The process of fitting the element $f^*$ of the RKHS $\mathcal{H}_k$ and then computing the power $P_c$ is described in terms of an optimisaton problem,

$$f^* = \operatorname*{argmin}_{f\in\mathcal{H}_k} \sum_{i=1}^{n} \sum_{j=Tc}^{T(c+1)} -\log p\big(y_{ij} \mid f(x_{ij})\big) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_k}, \qquad P_c = \sum_{i=1}^{n} \sum_{j=Tc}^{T(c+1)} \big(y_{ij} f^*(x_{ij})\big)^2,$$

where $p\big(y_{ij} \mid f(x_{ij})\big)$ is the evaluation of the likelihood of a minimal exponential family [Deisenroth et al., 2020, § 6.6] with canonical parameter $f(x_{ij})$, as considered by others Canu and Smola [2006]. In our setting, we use a Bernoulli likelihood, yielding kernel logistic regression [Zhu and Hastie, 2001]. This objective is strongly convex [Wainwright et al., 2008, Proposition 3.1], and so admits a unique global minima in the representer coefficients $\alpha_i$ [Wright and Recht, 2022, Theorem 2.8].

## 3 Experiments and results

We deploy our method[2] on all datasets in the SPARKESX data challenge, and compare our results with those obtained using PRESTO as reported in Yong et al. [2022]. We note that Yong et al. [2022] use nominal PRESTO parameters, and high values of DM were not searched for. We use a Bernoulli likelihood. We set $\lambda = 0.1$. We use a squared exponential kernel for $k$, with a lengthscale of 1. We find the representer coefficients $\alpha_i$ using Newton's method initialised at a vector of zeros. We set the length $T$ of each chunk to 50 discrete steps, where 4096 steps measures 1.024 seconds. For each dataset, we run 64 parallel processes across different chunks $c$ on a 64-core (CPU-based) Dell PowerEdge C6525 Server. This results in a computation time roughly 36 times slower than real-time.

Due to the highly imbalanced nature of the data (interesting events are rare), accuracy is not the most meaningful metric to report results. Instead, true positive rate (TPR) might apply to the hypothetical scenario where a computer produces candidate events for a scientist to manually inspect with a fixed time budget, and it is desirable that as many true events are selected. We set this fixed budget to be the top 10% most likely anomalous events. To even further break down our analysis, we view TPR as a function of three attributes of simulated data: DM, amplitude $A$ and signal width $W$. This helps us see where our method performs well in the parameter-space of simulated events. See Figure 2.

We observe that on more complicated signals beyond idealised `simplepulse`, our method outperforms PRESTO. This is expected behaviour, since the strong inductive bias is completely aligned with the idealised signals in `simplepulse`. However, for `known+rfi` dataset, which includes more realistic pulses, long duration pulses, flares and radio frequency interference (RFI), our method usually outperforms PRESTO. For non-physical signals, real unknown data, and mixed settings, our method almost always performs as well as or better than PRESTO, sometimes by a large margin.

[2] `https://github.com/yongsukyee/sparkesXML_klr/`

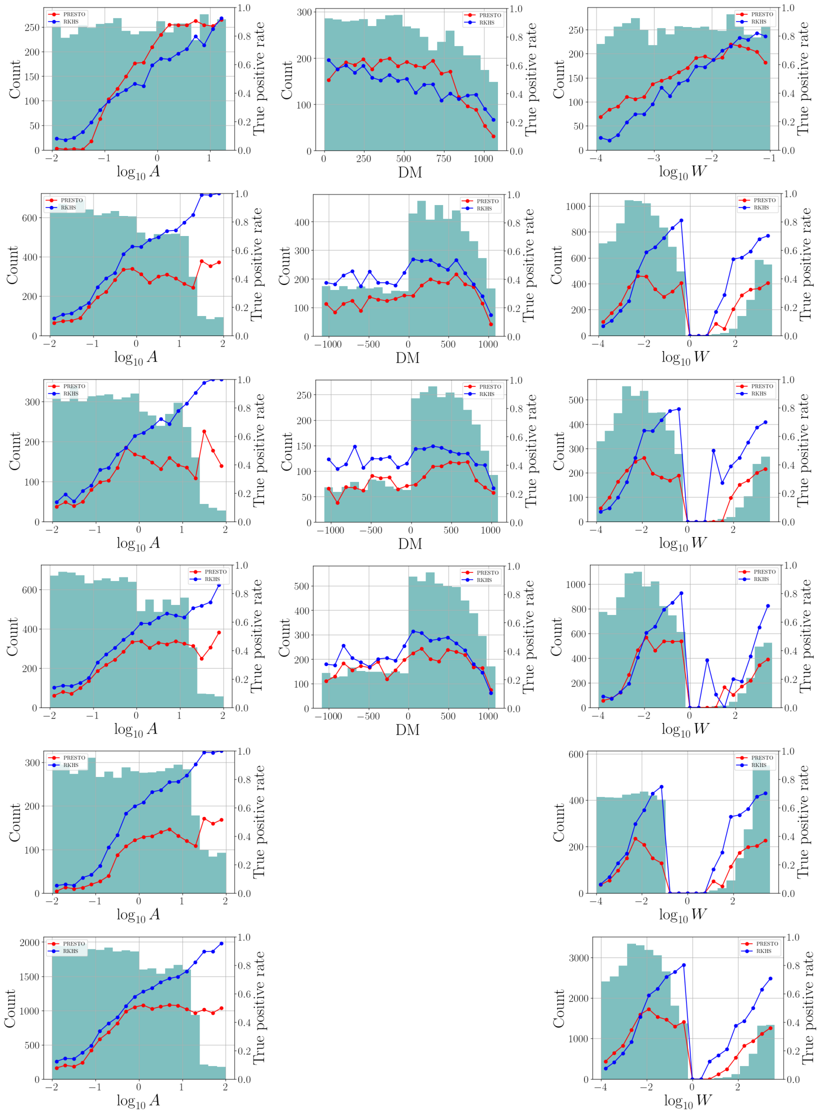

Figure 2: TPR as a function of attributes $\log A$, DM and $\log W$ overlayed on top of histograms showing frequency of attributes for each dataset. Our proposed RKHS method can outperform the matched filter using PRESTO in some settings. (Top to Bottom) `simplepulse`; `known+rfi`; `combo+rfi`; `real+combo`; `unknown+rfi`; All data. Some datasets do not have a DM parameter.

## 4 Conclusion, limitations and future work

Using an RKHS to detect structured signals is promising, both for detecting idealistic simple pulses and for detecting new types of interesting signals. Our current method on our hardware operates at about 36 times slower than real-time, but individual chunks may be processed in parallel. Increasing the speed will allow for real-time deployment of our system. We use a single length-scale to fit our RKHS element, but in future we may simultaneously fit in parallel multiple RKHS elements each using a different lengthscale. The resulting collection of powers could be used inside another ML model such as a one-class support vector machine to detect whether the signal is noise or interesting. Features obtained as solutions to optimisation problems, such as those used here, can be differentiated without differentiating through the solver using the implicit function theorem Gould et al. [2022], so that deep learning extensions might also be considered. Finally, non-binary data could be considered by using kernel binomial regression instead of kernel logistic regression.

## Acknowledgments and Disclosure of Funding

We thank George Hobbs and Lars Petersson for a helpful discussion. We gratefully acknowledge support from the Machine Learning and Artificial Intelligence Future Science Platform.

## References

Duncan Ross Lorimer and Michael Kramer. *Handbook of pulsar astronomy*. Cambridge University Press, 2012.

Duncan R Lorimer, Matthew Bailes, Maura Ann McLaughlin, David J Narkevic, and Froney Crawford. A bright millisecond radio burst of extragalactic origin. *Science*, 318(5851):777–780, 2007.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

Suk Yee Yong, George Hobbs, Minh T. Huynh, Vivien Rolland, Lars Petersson, Ray P. Norris, Shi Dai, Rui Luo, and Andrew Zic. SPARKESX: Single-dish PARKES data sets for finding the uneXpected - a data challenge. *MNRAS*, 516(4):5832–5848, November 2022. doi: 10.1093/mnras/stac2558.

Marc Peter Deisenroth, A Aldo Faisal, and Cheng-Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

Stéphane Canu and Alex Smola. Kernel methods and the exponential family. *Neurocomputing*, 69 (7-9):714–720, 2006.

Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. *Advances in neural information processing systems*, 14, 2001.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Stephen J. Wright and Benjamin Recht. *Foundations of Smooth Optimization*, page 15–25. Cambridge University Press, 2022. doi: 10.1017/9781009004282.003.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks, Aug 2022.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 4
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work does not have any direct negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See footnote 2.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 3
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Ours was not a randomised algorithm and trained in an unsupervised setting without data splits.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See footnote 2
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See footnote 2.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]