MicroDrift with Bayesian Covertrees

Sven Cattell

CAMLIS, 2021

About Me

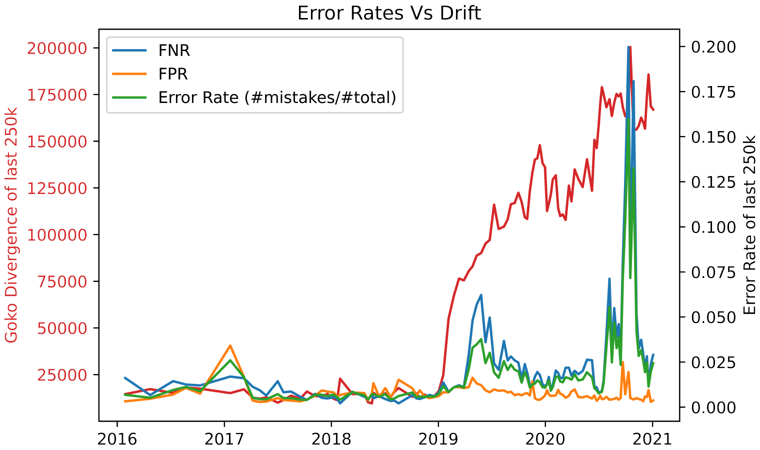
- Ph.D. in Algebraic Topology from JHU
- Very involved in the AI Village
- Formerly at Endgame / Elastic

Table of Contents

Table of Contents

Problem Formulation

Previous Work: Chronological Drift



Work done at Elastic, published at ICLR

Problems With Previous Work

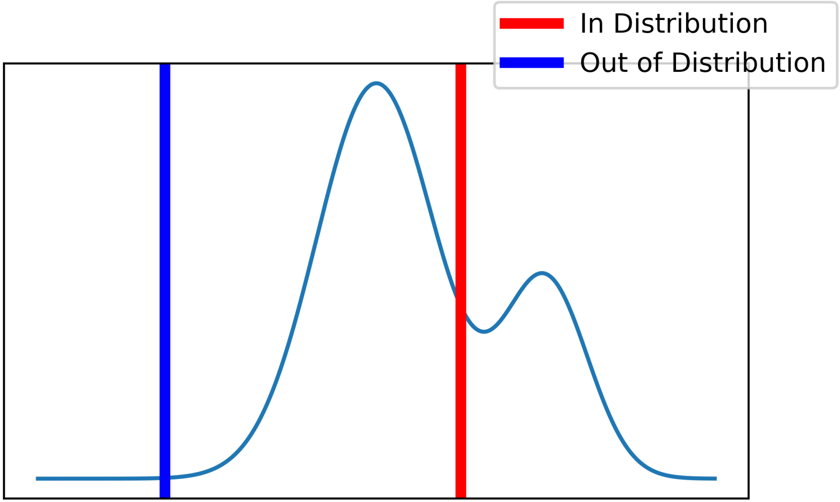
- Doesn't model the efficacy metrics well
- Not that actionable, just "Retrain when KL-Div exceeds X"
- There's way more detail than just a single metric in the method

Objective of This Talk

Tell me where there's a problem in my dataset, not just that there's a problem.

Where am I being attacked/bypassed?
Where is that new malware family?
Where is that new popular spam technique?

Types of Bypass



What I'm Actually Doing

- We have a dataset, and model.
- Queries stream in from anonymous users.
- One user has an in-distribution "bypass" they are repeating.
  - Building an attack with ZOO, or HopSkipJump.
  - Spamming their spam everywhere.
- The bad user's queries only account for a small percentage of total traffic.
- *We want to isolate that user's queries as best as possible.*

Table of Contents

Definition

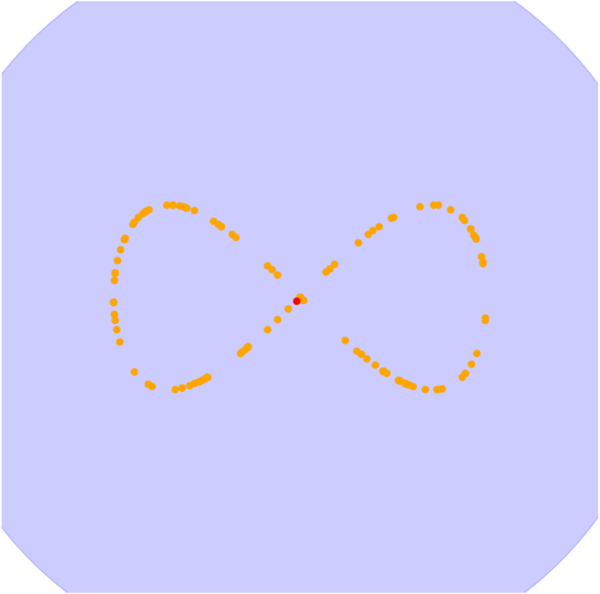
A *covertree* over a dataset $X = \{x_1, \dots x_n\}$ is a filtration of a dataset into *m-layers*, with a scale base of S

$$\{x_r\} = C_k \subset C_{k-1} \subset \cdots \subset C_{k-m} = X,$$

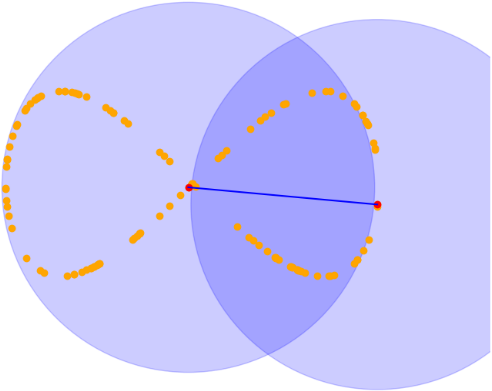
which satisfies the following properties:

1. *Covering Layer*: For each $x_j \in X$ and $i \in \{k, \dots k-m\}$, there exists $p \in C_i$ such that $d(x_i, p) < s^i$.
2. *Covering Tree*: For each $p \in C_{i-1}$ there exists $q \in C_i$ such that $d(p, q) < s^i$.
3. *Separation*: For all $p, q \in C_i$, $d(p, q) > s^i$.

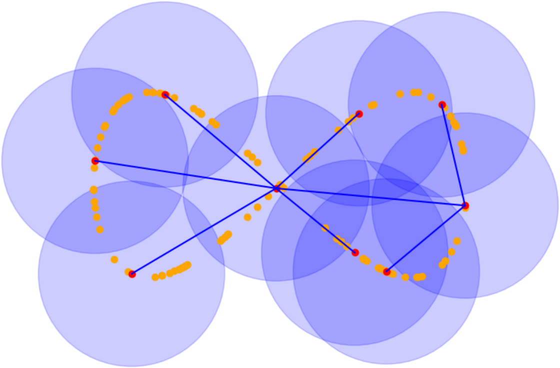
Lets's build one, Level 1



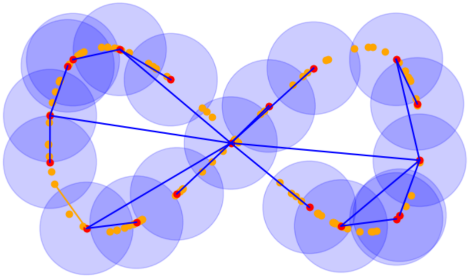
Lets's build one, Level 0



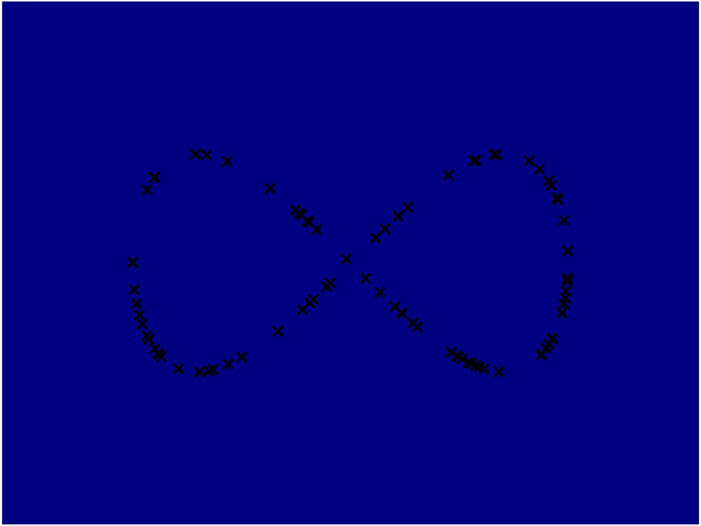
Lets's build one, Level -1



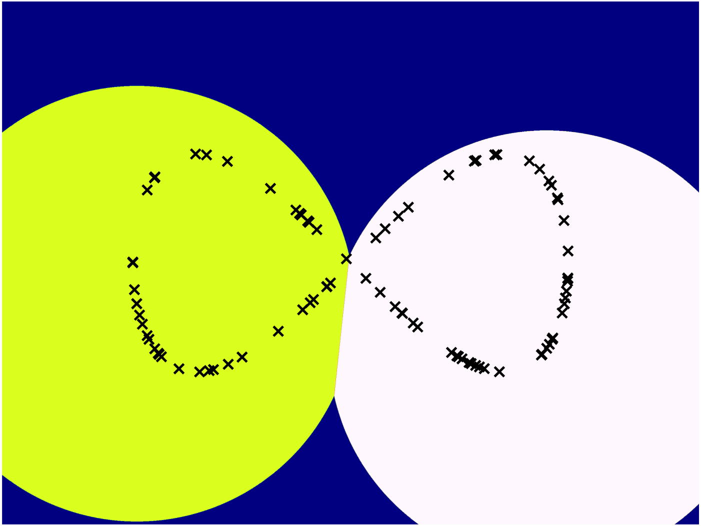
Lets's build one, Level -2



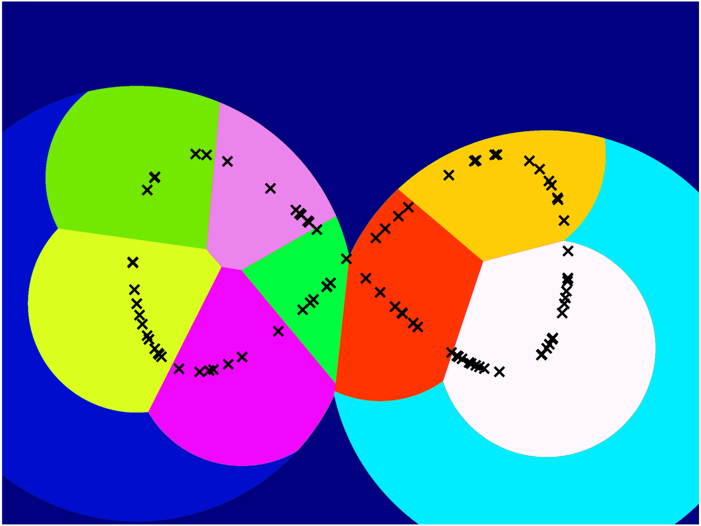
How A Covertree Partitions Space, Level 1



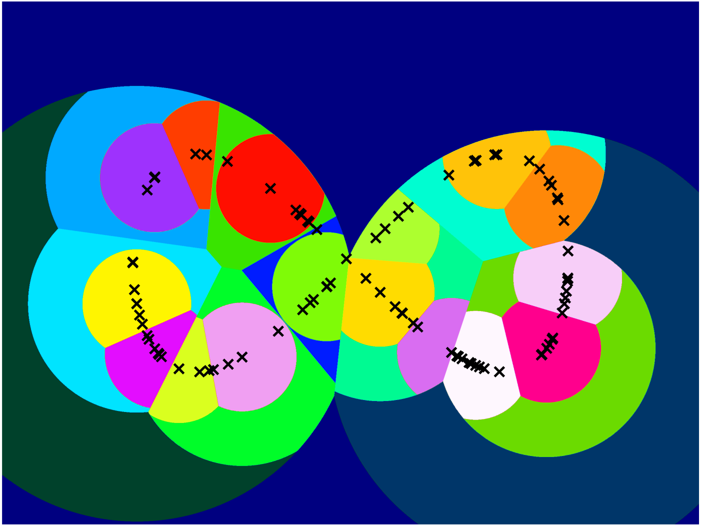
How A Covertree Partitions Space, Level 0



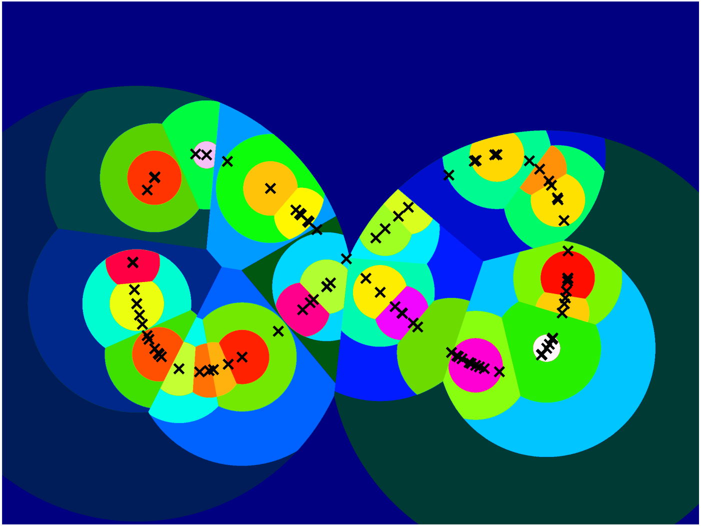
How A Covertree Partitions Space, Level -1



How A Covertree Partitions Space, Level -2



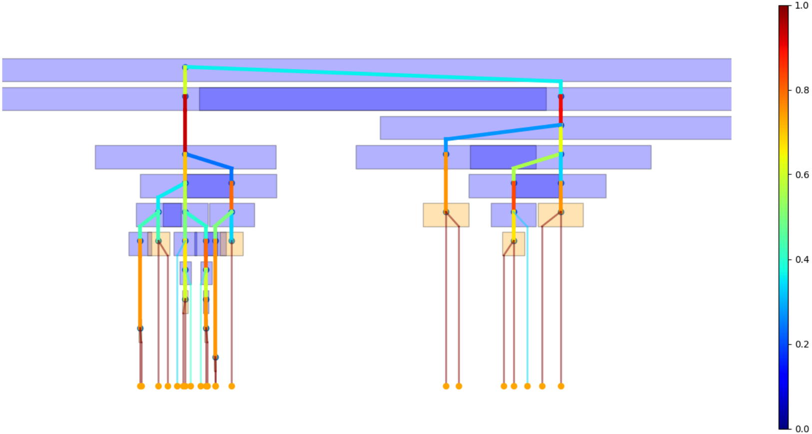
How A Covertree Partitions Space, Level -3



A simple approximation of the true distribution

- Each node covers N elements of the tree.
- The node's children cover $(N_1, N_2, \ldots N_k)$
- Therefore the probability of a point associated to the parent node, is associated to the ith child node is $\frac{N_i}{N}$

Approximating the Probability Distribution From a Covertree



Oops, The Estimate was Wrong

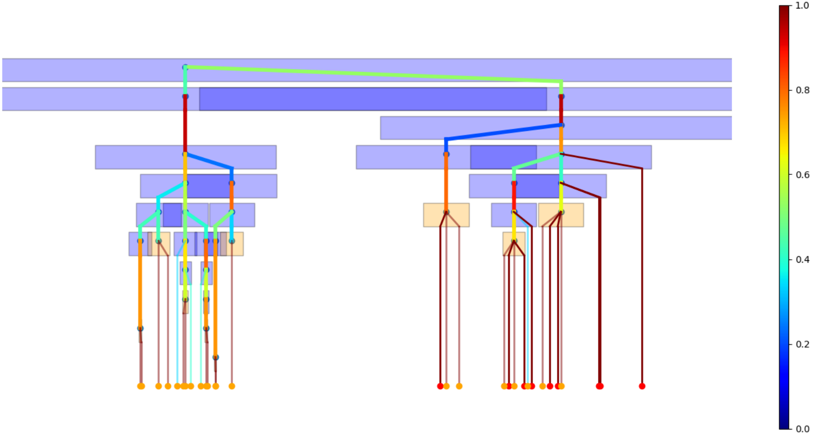


Table of Contents

Let's be Bayesian about this

- We know a lot about the root of the tree, lots of observations.
- We know little about the leaves of the tree, few observations.
- Therefore, model the distribution of distributions, using a Dirichlet distribution.

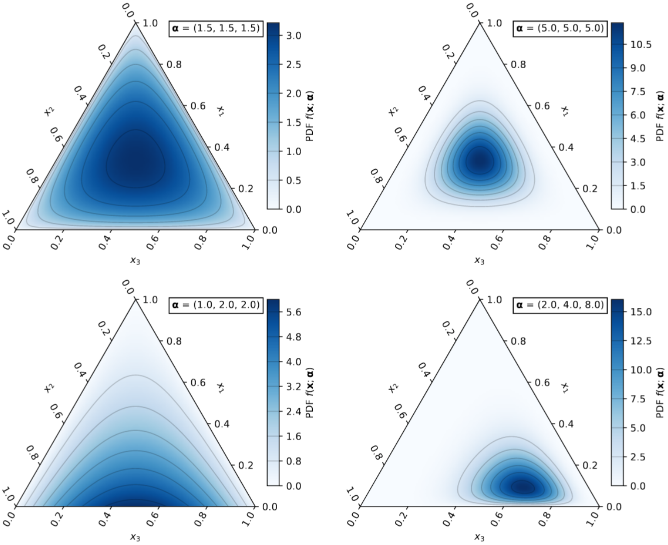
A Node's Dirichlet Distribution

For node covering N_0, with children covering $\alpha = (N_1, \ldots, N_k)$, we associate a Dirichlet Distribution $\mathrm{Dir}(\alpha)$. The probability density function for this is:

$$f(x_1, \ldots, x_k; N_1, \ldots, N_k) = \frac{\prod_{i=1}^{k} \Gamma(N_i)}{\Gamma(N_0)} \prod x_i^{N_i - 1}$$

Can also do this with all nodes for the "overall distribution"

A Dirichlet Visualization [1]



[1] Source: Wikipedia

Prior VS Posterior

The *prior* associated to a node is $\text{Dir}((1, \ldots, 1))$. The training posterior is

$$P_A = \text{Dir}((N_1 + 1, \ldots, N_k + 1)).$$

If there are O_i points in the test set whose paths pass through the ith child, then the test-posterior is:

$$Q_A = \text{Dir}((N_1 + O_1 + 1, \ldots, N_k + O_k + 1)).$$

Drift Metrics: Kullback–Leibler divergence [2]

$$\mathrm{KL}(Q_A||P_A) = \log\Gamma(N_0) - \log\Gamma(N_0 + O_0) + \sum_{i=1}^{k}\{\Gamma(N_i + O_i) - \Gamma(N_i) + O_i(\psi(N_i) - \psi(N_0))\} \quad (1)$$

[2] Source: https://bariskurt.com/kullback-leibler-divergence-between-two-dirichlet-and-beta-distributions/

Marginal Log Likelihood of Test, Given Observations

Model the distributions of multinomial distributions with O samples instead of categorical, then calculate the ln of the marginal distribution:

$$\mathrm{MLL}(O|N) = \log\Gamma(N_0) + \log\Gamma(O_0+1) - \log\Gamma(N_0+O_0) + \sum_{i=1}^{k}\{\Gamma(N_i+O_i) - \Gamma(N_i) - \Gamma(O_i+1)\} \quad (2)$$

Visualization Of KL Div VS MLL

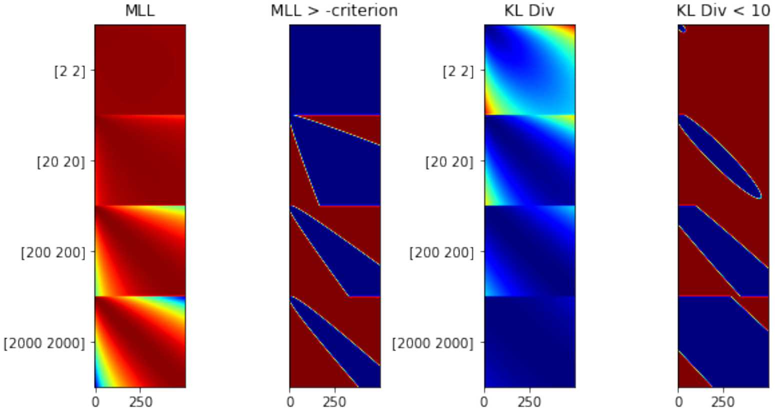
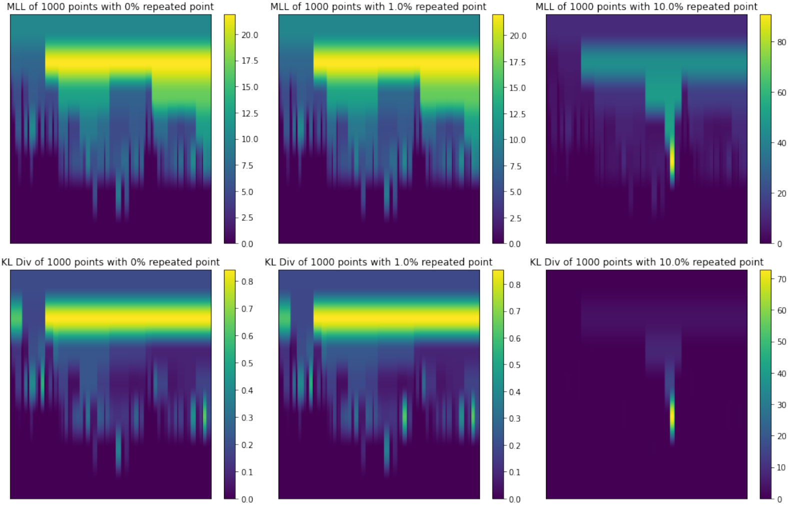


Table of Contents

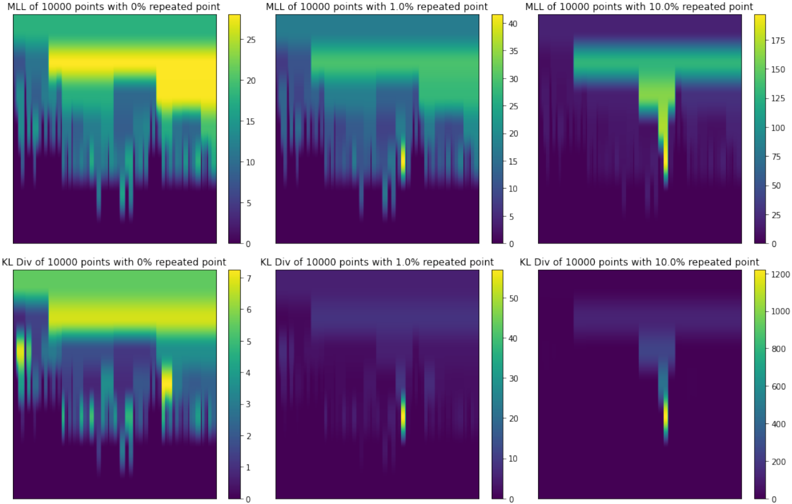
Let's build some intuition

1. Our training set will be 10000 points from a 2D daussian.
2. Or test sets will be 1000, and 10000 points sampled from the same gaussian.
3. We'll sample the attack point from the same gaussian.
4. We'll replace 0%, 1% and 10% of the test set with the attack point, these are the attack rates.

Visualization Of Gaussian Toy



Visualization Of Gaussian Toy



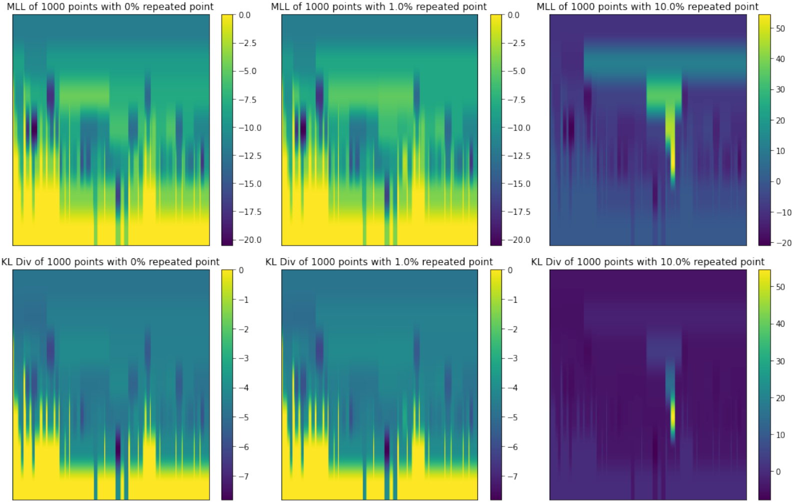
How to do Classification

Take a baseline, B, run some sequences through the covertree's tracker and calculate the per-node maximum, and standard deviation.

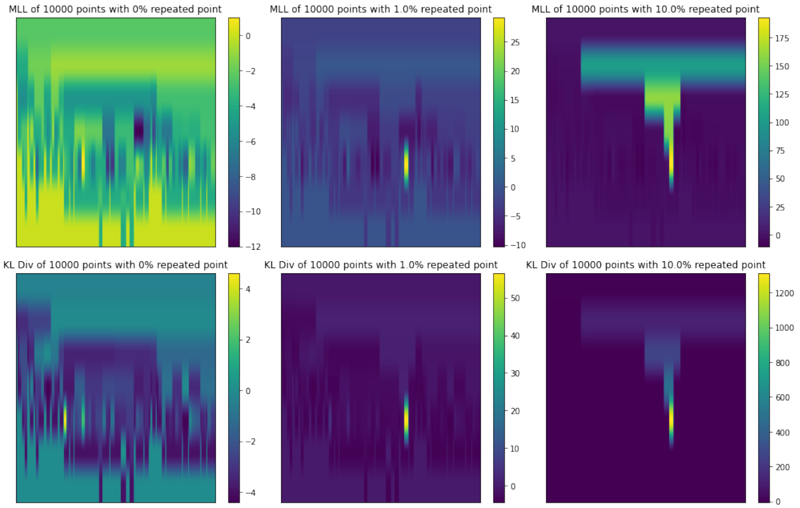
$$\widehat{\mathrm{KL}}_B(Q_a||P_a) = \mathrm{KL}(Q_a||P_a) - \max_{B} \mathrm{KL}(Q_a||P_a) - S_{\mathrm{KL}}\sigma_{\mathrm{KL}} - C_{\mathrm{KL}}$$

$$\widehat{\mathrm{MLL}}_B(O||N) = \mathrm{MLL}(O||N) - \max_{b \in B} \mathrm{MLL}(O_b||N_b) - S_{\mathrm{MLL}}\sigma_{\mathrm{MLL}} - C_{\mathrm{MLL}}$$

Visualization Of Gaussian Toy



Visualization Of Gaussian Toy



Definition of Detection

A "detection" is performed in 2 passes, the first is the address of the node with the maximal positive $\widehat{\mathrm{KL}}_B(Q_a||P_a)$.
If $\widehat{\mathrm{KL}}_B(Q_a||P_a)$ is everywhere non-positive, the address of the node with maximal positive $\widehat{\mathrm{MLL}}_B(O||N)$.
If both terms are non-positive for all nodes, nothing is detected.

Table of Contents

Overall KL Divergence of SOREL's test set

	Window size					
	1000		10000		100000	
Attack Rate	μ	σ	μ	σ	μ	σ
0.0	0.0	1.0	0.0	1.0	0.0	1.0
0.0001	8e-5	1.0	3e-5	1.0	2e-5	0.999
0.001	0.0001	0.99	0.0003	1.0	0.0004	1.0
0.01	0.007	1.03	0.009	1.06	0.014	1.095
0.10	0.293	4.025	0.299	4.167	0.329	4.122
1.00	10.172	55.40	7.379	41.376	5.987	36.260

Overall Marginal Log Likelihood of SOREL's test set

	Window size					
	1000		10000		100000	
Attack Rate	μ	σ	μ	σ	μ	σ
0.0	0.0	1.0	0.0	1.0	0.0	1.0
0.0001	-0.0006	1.0	-0.0014	1.0	-0.0053	1.00
0.001	-0.004	0.99	-0.04	1.0	-0.16	1.00
0.01	-0.18	1.03	-0.92	1.08	-2.70	1.095
0.10	-3.78	1.66	-13.45	1.75	-32.26	1.38
1.00	-53.91	4.382	-160.87	2.78	-407.52	1.456

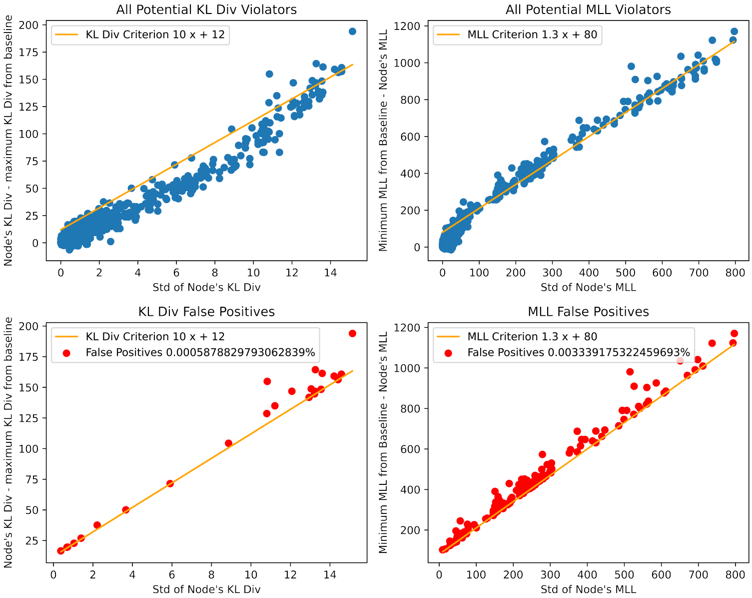
SOREL Baseline Adjustment

Took a baseline, with a validation set. Did leave one out cross validation and adjusted the 4 hyperparameters until the following saw next to no FPS. There's an extra term ω called the *margin of safety*. I used 1.5.

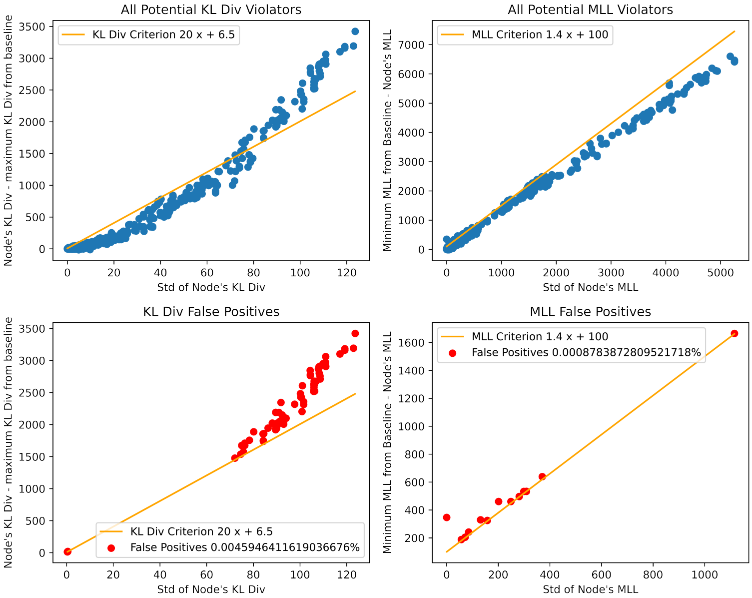
$$\widehat{\mathrm{KL}}_B(Q_a||P_a) = \omega \mathrm{KL}(Q_a||P_a) - \max_B \mathrm{KL}(Q_a||P_a) - S_{\mathrm{KL}}\sigma_{\mathrm{KL}} - C_{\mathrm{KL}}$$

$$\widehat{\mathrm{MLL}}_B(O||N) = \omega \mathrm{MLL}(O||N) - \max_{b \in B} \mathrm{MLL}(O_b||N_b) - S_{\mathrm{MLL}}\sigma_{\mathrm{MLL}} - C_{\mathrm{MLL}}$$

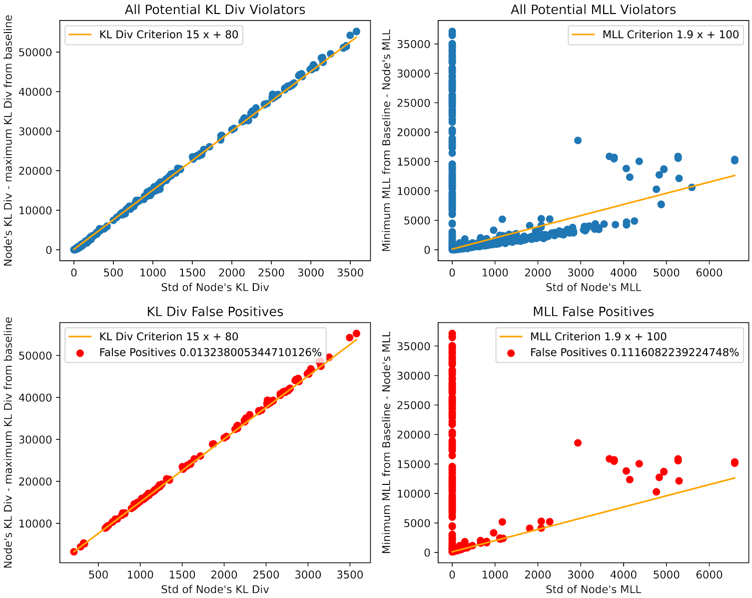
Visualization Of SOREL Baseline Adjustment for 1000



Visualization Of SOREL Baseline Adjustment for 10000



Visualization Of SOREL Baseline Adjustment for 100000



Safe Baseline Hyperparameter Results

With a safety margin of 2.

Window Size	S_{KL}	C_{KL}	S_{ML}	C_{ML}
1000	10	12	1.3	80
10000	20	6.5	1.4	100
100000	15	80	1.9	100

Safe Test Set Results

		Attack Rates					
Window Size		0%	0.01%	0.1%	1%	10%	100%
1000	TPR	0	0	0	0.7	88	100
	FPR	0	0	0	0	0	0
	MDR	-	-	-	96	87	93
10000	TPR	0	0	0.7	63.7	99.95	100
	FPR	0	0	0	0	0	0
	MDR	-	-	96	93	93	91
100000	TPR	0	0.1	22.7	98.4	100	100
	FPR	0.4	0.3	0	0	0	0
	MDR	-	85	94	93	92	88

Mean Depth Rate - Detection depth of attack over the final depth. All values in percentages. Averaged over 1972 runs with 48 different trees.

Not So Safe Baseline Hyperparameter Results

With a safety margin of 1.3.

Window Size	S_{KL}	C_{KL}	S_{ML}	C_{ML}
1000	8	7	1.3	20
10000	10	6.5	1.3	20
100000	10	40	1.7	50

Not So Safe Test Set Attack Results for SOREL

		Attack Rates					
Window Size		0%	0.01%	0.1%	1%	10%	100%
1000	TPR	0	0	0	16.6	96	100
	FPR	0	0	0	0	0	0
	MDR	-	-	-	94	89	93
10000	TPR	0	0	5	81	99.95	100
	FPR	0	0	0	0	0	0
	MDR	-	-	94	91	93	91
100000	TPR	0	0.2	44.5	98.4	100	100
	FPR	0.1	0.9	0.6	0	0	0
	MDR	-	84	94	94	92	88

Mean Depth Rate - Detection depth of attack over the final depth. All values in percentages. Averaged over 1972 runs with 48 different trees.