ICML
International Conference
On Machine Learning

POLITECNICO
MILANO 1863

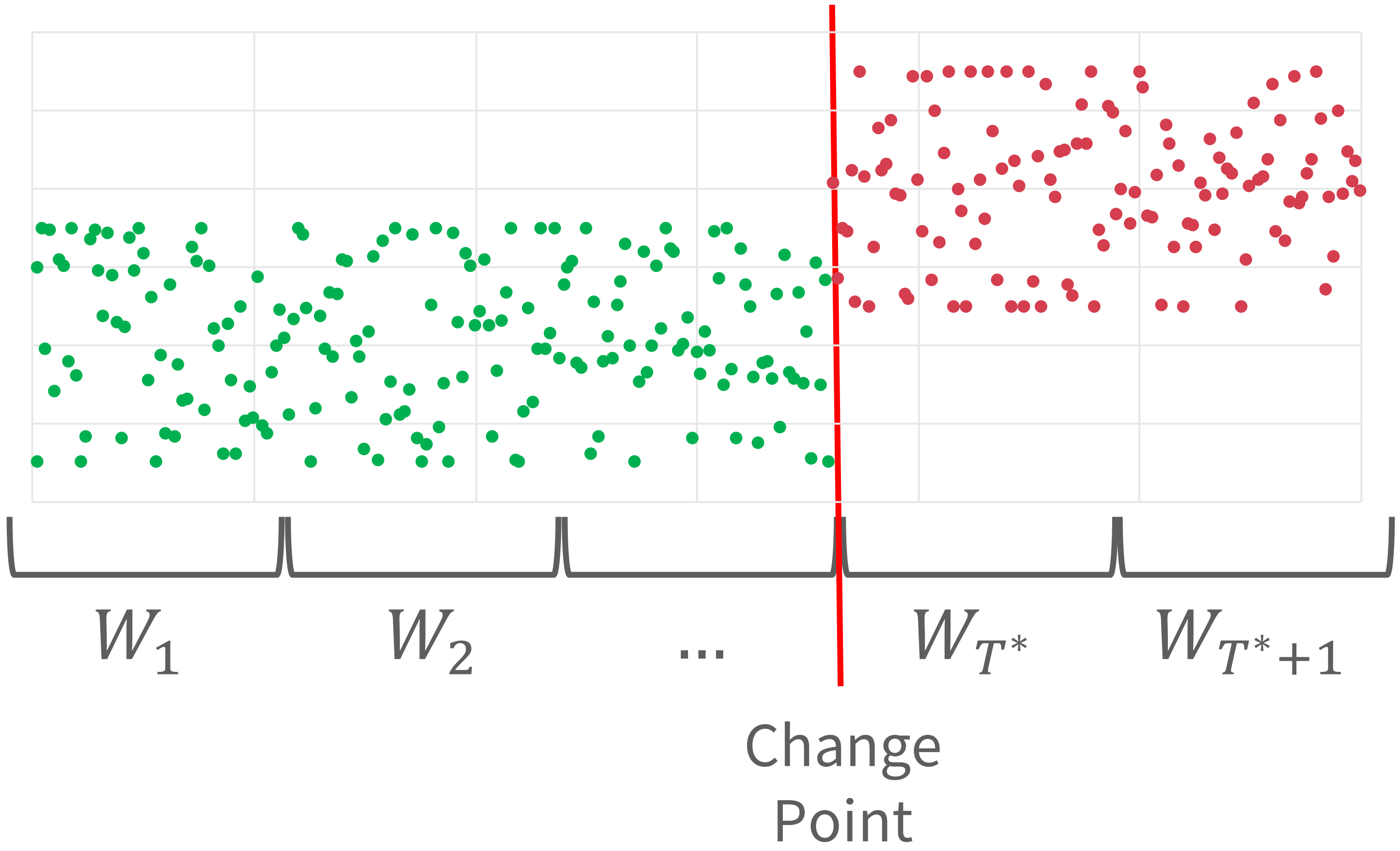
Kernel QuantTree

Diego Stucchi, Paolo Rizzo, Nicolò Folloni, Giacomo Boracchi

Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

International Conference on Machine Learning (ICML)
23-29 July 2023

Batch-wise Multivariate Change Detection

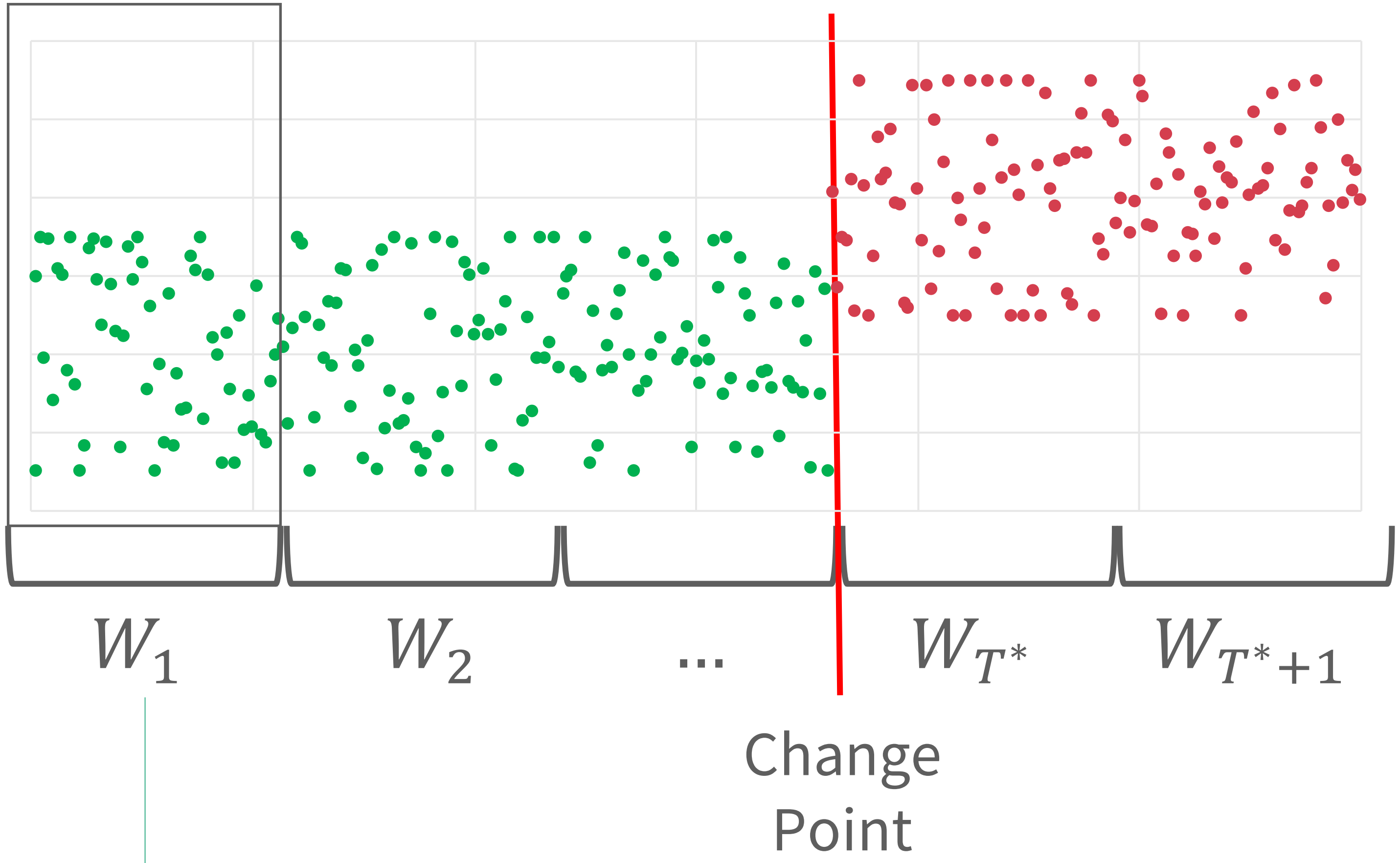


$$x \sim \begin{cases} \phi_0 \text{ if } x \in W_i, i \leq T^* \\ \phi_1 \text{ if } x \in W_i, i > T^* \end{cases}$$

Fixed-size batches
$|W| = \nu$

Test statistic
$\mathcal{T}: [\mathbb{R}^d]^\nu \to \mathbb{R}$

Multivariate Change Detection



$$x \sim \begin{cases} \phi_0 \text{ if } x \in W_i, i \leq T^* \\ \phi_1 \text{ if } x \in W_i, i > T^* \end{cases}$$

Fixed-size batches

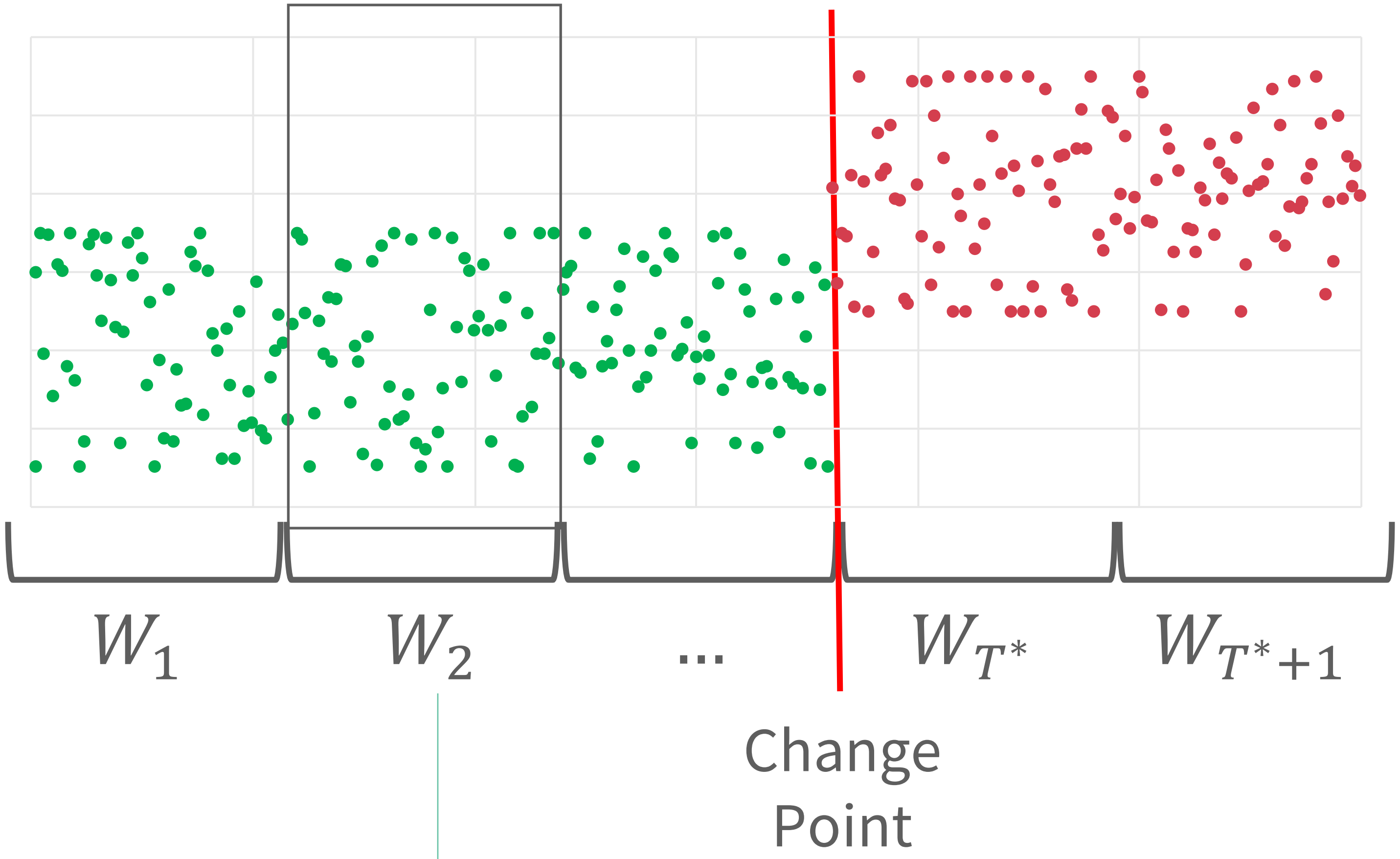
$|W| = \nu$

Test statistic

$\mathcal{T}: [\mathbb{R}^d]^\nu \to \mathbb{R}$

$\mathcal{T}(W_1) < \tau \ \rightarrow \ W_1 \sim \phi_0$

Multivariate Change Detection



$$x \sim \begin{cases} \phi_0 \text{ if } x \in W_i, i \leq T^* \\ \phi_1 \text{ if } x \in W_i, i > T^* \end{cases}$$

Fixed-size batches

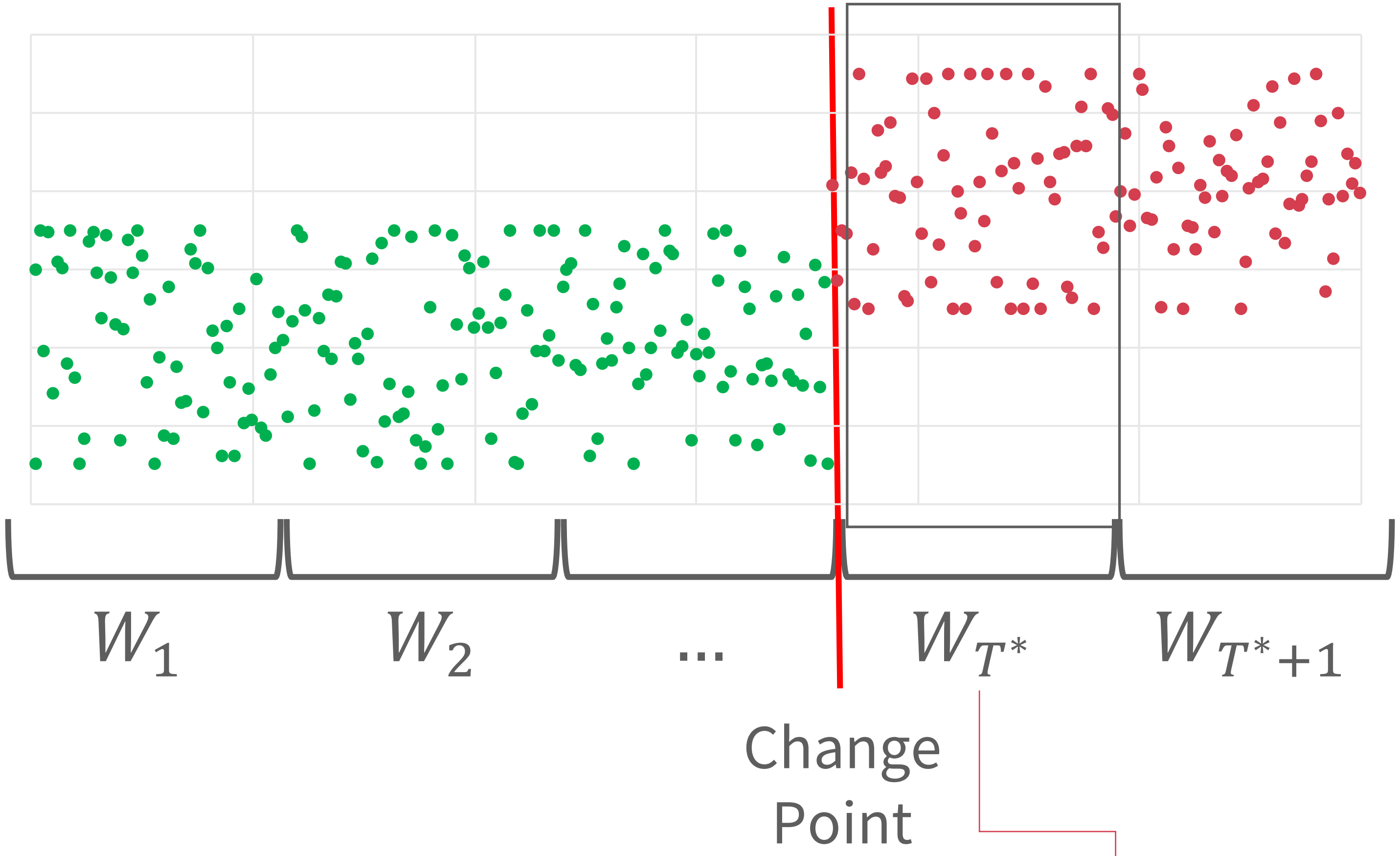
$|W| = \nu$

Test statistic

$\mathcal{T}: [\mathbb{R}^d]^\nu \to \mathbb{R}$

$\mathcal{T}(W_2) < \tau \ \rightarrow \ W_2 \sim \phi_0$

Multivariate Change Detection



$$x \sim \begin{cases} \phi_0 \text{ if } x \in W_i, i \le T^* \\ \phi_1 \text{ if } x \in W_i, i > T^* \end{cases}$$

Fixed-size batches

$|W| = \nu$

Test statistic

$\mathcal{T}: [\mathbb{R}^d]^\nu \to \mathbb{R}$

$\mathcal{T}(W_{T^*}) \geq \tau \ \rightarrow \ W_{T^*} \sim \phi_1$

Background: QuantTree [1]

[1] Boracchi, Carrera, Cervellera, Macciò "QuantTree: histograms for change detection in multivariate data streams." ICML 2018

Training:

- Construct **histogram** $h = \{(S_k, \hat{\pi}_k)\}_{k=1}^{K}$ over **training set** TR of N samples
- Compute **detection threshold** $\tau = \tau(\alpha)$ by Monte Carlo simulations

Inference:

- Compute **bin counts** $y_k = |W \cap S_k|$ for all k
- Compute **test statistic** $\mathcal{T}(W) = \mathcal{T}(y_1, \ldots, y_K)$
- Detect change when $\mathcal{T}(W) > \tau$

Background: QuantTree [1]

[1] Boracchi, Carrera, Cervellera, Macciò "QuantTree: histograms for change detection in multivariate data streams." ICML 2018

Training:

- Construct **histogram** $h = \{(S_k, \hat{\pi}_k)\}_{k=1}^{K}$ over **training set** TR of N samples
- Compute **detection threshold** $\tau = \tau(\alpha)$ by Monte Carlo simulations

Inference:

- Compute **bin counts** $y_k = |W \cap S_k|$ for all k
- Compute **test statistic** $\mathcal{T}(W) = \mathcal{T}(y_1, \dots, y_K)$
- Detect change when $\mathcal{T}(W) > \tau$

Background: QuantTree [1]

[1] Boracchi, Carrera, Cervellera, Macciò "QuantTree: histograms for change detection in multivariate data streams." ICML 2018

Training:

- Construct **histogram** $h = \{(S_k, \hat{\pi}_k)\}_{k=1}^{K}$ over **training set** TR of N samples
- Compute **detection threshold** $\tau = \tau(\alpha)$ by Monte Carlo simulations

Inference:

- Compute **bin counts** $y_k = |W \cap S_k|$ for all k
- Compute **test statistic** $\mathcal{T}(W) = \mathcal{T}(y_1, \dots, y_K)$
- Detect change when $\mathcal{T}(W) > \tau$

Pros and Cons of QuantTree

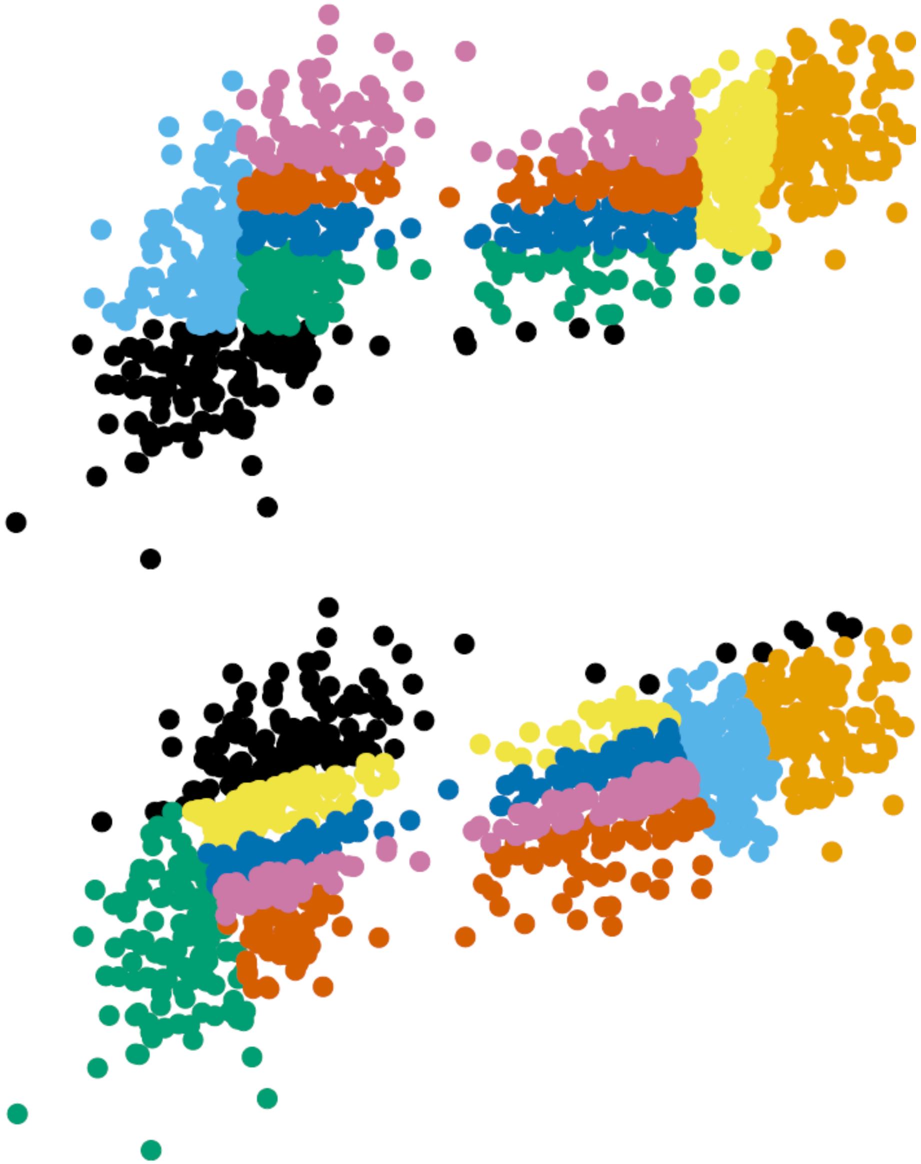
✓ Practical monitoring

✓ Control of the False Positive Rate

Pros and Cons of QuantTree

- ✓ Practical monitoring
- ✓ Control of the False Positive Rate
- ✗ Limited to axis-aligned splits
- ✗ Mostly bins of non-finite volume

Pros and Cons of QuantTree



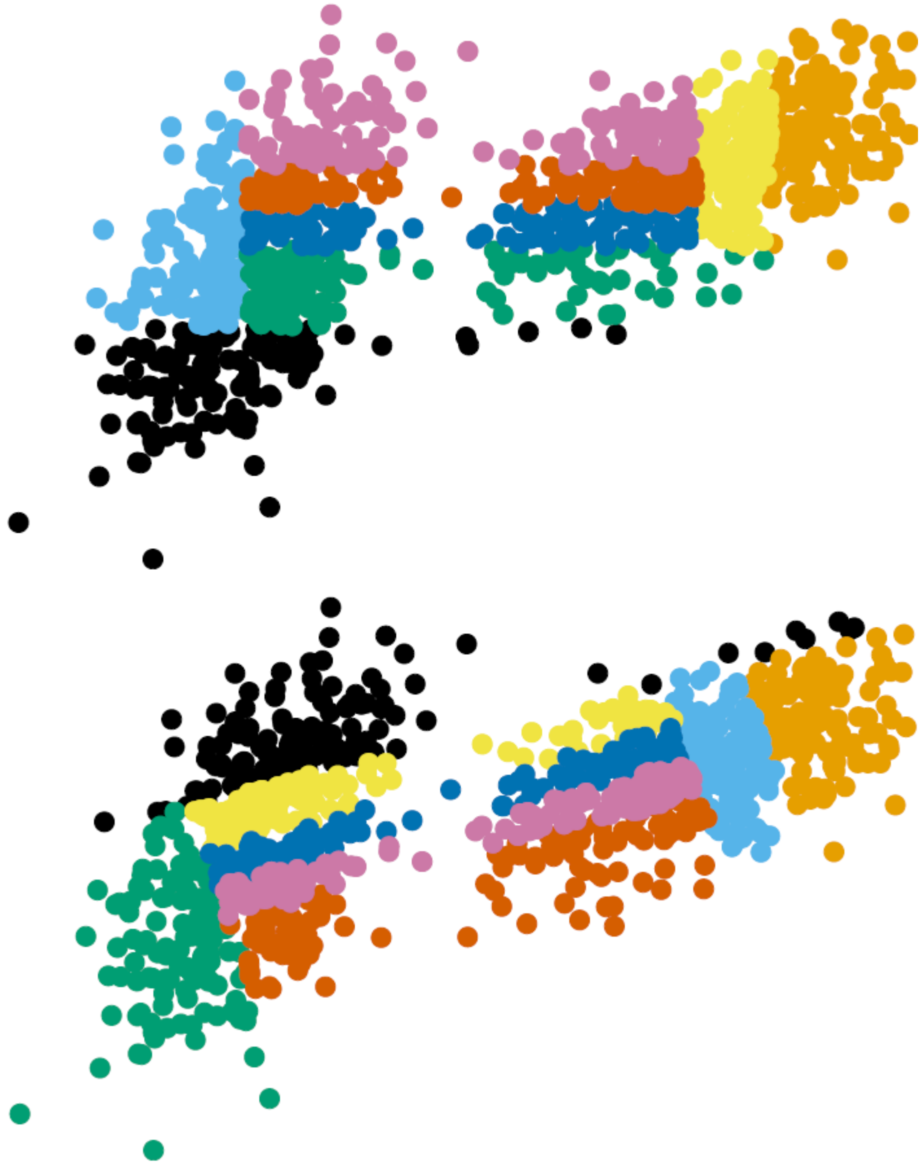
✓ Practical monitoring

✓ Control of the False Positive Rate

✗ Limited to axis-aligned splits

✗ Mostly bins of non-finite volume

Pros and Cons of QuantTree



- ✓ Practical monitoring
- ✓ Control of the False Positive Rate
- ✗ Limited to axis-aligned splits
- ✗ Mostly bins of non-finite volume
- ✗ PCA not always beneficial to performance

Kernel QuantTree

- **Compact bins** defined as sublevel sets of **measurable kernel functions**
- **Generalization** of the theoretical properties of QuantTree, enabling **control of the False Positive Rate**
- Independent of **roto-translations**, hence **does not require PCA** preprocessing

Kernel QuantTree: Histogram Construction

$\mathbb{R}^d$

$S_1 = \{x \in \mathbb{R}^d \mid f_1(x) \leq q_1\}$

$S_2 = \{x \in \mathbb{R}^d \setminus S_1 \mid f_2(x) \leq q_2\}$

…

$S_{K-1} = \{x \in \mathbb{R}^d \setminus \bigcup_{j<K-1} S_j \mid f_2(x) \leq q_2\}$

$S_K = \mathbb{R}^d \setminus (S_1 \cup \cdots S_{K-1})$

$f_k: \mathbb{R}^d \to \mathbb{R}$
Measurable Kernel Functions

$q_k \in \mathbb{R}$
Split Values

Kernel QuantTree: Kernel Functions

$$f_k: \mathbb{R}^d \to \mathbb{R}$$

$$f_k(x) = (x - c_k)^T A (x - c_k)$$

Kernel matrix - Determines the resulting distance

Centroid – Determines the location of the bin

Euclidean Distance

$$f_k(x) = (x - c_k)^T (x - c_k)$$

$$A = \mathbb{I}_d$$

Identity matrix

Mahalanobis Distance

$$f_k(x) = (x - c_k)^T \Sigma^{-1} (x - c_k)$$

$$A = \Sigma^{-1}$$

Sample covariance matrix of TR

Weighted Mahalanobis Distance [2]

[2] Tipping "Deriving cluster analytic distance functions from gaussian mixture models." ICANN 1999

$$f_k(x) = (x - c_k)^T A(x)(x - c_k)$$

$$A(x) = \frac{\sum_{m=1}^{M} \rho_m \cdot i_m(x, c_k) \cdot C_m^{-1}}{\sum_{m=1}^{M} \rho_m \cdot i_m(x, c_k)}$$

Weighted average of covariance matrices C_m from a Gaussian Mixture Model fitted to TR

Performance-Complexity Tradeoff

Euclidean
Distance

Mahalanobis
Distance

Weighted Mahalanobis
Distance

Detection Performance

Computational Complexity

Theoretical Results – Independence Theorem

Theorem 1. *Let* $h = \{(S_k, \hat{\pi}_k)\}_{k=1}^{K}$ *be a KQT histogram constructed using measurable functions* $f_k: \mathbb{R}^d \to \mathbb{R}$. *Let* $\mathcal{T}_h$ *be a statistic defined over batches* W *depending only on the number* $\{y_k\}$ *of samples of* W *falling in the bins of* h. *Then, the distribution of* $\mathcal{T}_h$ *over stationary batches* $W \sim \phi_0$ *depends only on the* ν, $N = |TR|$ *and target probabilities* $\{\pi_k\}_k$.

- Enables computing detection thresholds by **Monte Carlo simulations**
- Thresholds are **independent** of ϕ_0 or its dimension d, no bootstrap or training date are required.
- Thresholds can be set to **control the False Positive Rate**

Theoretical Results – Roto-translational Invariance

Theorem 2. *Let* $\Phi: \mathbb{R}^d \to \mathbb{R}^d$ *be a roto-translation. Let* $h = \{(S_k, \hat{\pi}_k)\}$ *and* $h' = \{(S'_k, \hat{\pi}'_k)\}$ *be the KQT histograms constructed from the training sets* $TR \subset \mathbb{R}^d$ *and* $TR' = \Phi(TR)$, *where the kernel function is either the Euclidean, Mahalanobis or Weighted Mahalanobis distance. Then, we have that* $S'_k = \Phi(S_k)$ *and* $\hat{\pi}'_k = \hat{\pi}_k$ *for* $k = 1, \dots, K$. *In particular, for any batch* W, *if we compute* $W' = \Phi(W)$, *we have that* $\mathcal{T}_h(W) = \mathcal{T}_{h'}(W')$.

- **No PCA** preprocessing is required

Experimental Validation

[3] Liu, Lu, Zhang "Concept drift detection via equal intensity k-means space partitioning." IEEE ToCybernetics 2020.
[4] Kuncheva "Change detection in streaming multivariate data using likelihood detectors." IEEE TKDE 2011

	d	QT (w/o PCA)	QT (w/ PCA)	KQT Euclidean	KQT Mahalanobis	KQT Weighted Mahalanobis	EIkM [3]	SPLL [4] (w/ PCA)
Unimodal	4	4,83%/0,95	4,81%/0,97	4,86%/0,94	4,82%/0,99	4,83%/0,99	4,82%/0,87	5,92%/0,98
Bimodal	4	4,80%/0,90	4,81%/0,93	4,80%/0,90	4,81%/0,95	4,80%/**0,96**	4,82%/0,82	6,02%/0,89
Nino	5	5,04%/0,84	4,99%/0,90	5,00%/0,60	5,02%/0,90	5,01%/**0,92**	4,83%/0,52	7,69%/0,84
Protein	9	4,97%/0,89	4,98%/0,98	4,98%/0,61	4,98%/0,99	5,03%/**0,99**	4,88%/0,51	8,42%/0,95
Credit	28	4,84%/0,69	4,96%/0,86	4,89%/0,60	4,85%/0,78	5,06%/**1,00**	4,96%/0,50	16,06%/0,66
Insects	33	4,93%/0,99	4,91%/0,99	4,92%/1,00	4,96%/**1,00**	5,25%/**1,00**	4,96%/0,96	6,16%/1,00
Sensorless	48	4,84%/0,86	5,01%/1,00	4,82%/0,54	5,01%/**1,00**	7,42%/**1,00**	4,93%/0,50	4,83%/**1,00**
Particle	50	4,85%/0,88	4,87%/0,93	4,81%/0,55	4,94%/0,97	5,80%/**0,99**	4,84%/0,50	6,07%/0,90
Avg. Rank		5.24	4.93	7.08	3.82	**2.98**	9.37	5.34

High-Dimensional Data

		KQT (Mahalanobis)		KQT (Weighted Mahalanobis)	
	d	N=4096	N=16384	N=4096	N=16384
Unimodal	16	4.81%	4.89%	4.88% (99.5)	4.79% (67.7)
Unimodal	32	4.95%	4.88%	4.99% (150.6)	4.81% (123.6)
Unimodal	64	5.80%	4.95%	5.81% (315.2)	4.87% (223.8)
Unimodal	128	16.52%	5.31%	77.74% (344.0)	5.45% (307.0)
Bimodal	16	4.88%	4.82%	4.88% (68.3)	4.87% (90.3)
Bimodal	32	4.95%	4.86%	5.36% (177.2)	4.83% (120.4)
Bimodal	64	5.66%	4.86%	5.70% (253.9)	5.03% (220.3)
Bimodal	128	15.44%	5.32%	76.60% (276.9)	5.46% (244.7)

Conclusion

Kernel QuantTree
(Weighted Mahalanobis)

- **State-of-the-art** detection performance
- Very **practical monitoring**
- **Compact bins** defined as sublevel sets of **measurable kernel functions**
- **Generalization** of the theoretical properties of QuantTree, enabling **control of the False Positive Rate**
- Independent of **roto-translations**, hence **does not require PCA** preprocessing