KARL: Fast Kernel Aggregation Queries

Authors: (Edison) Tsz Nam Chan[1,2], Man Lung Yiu[2] and Leong Hou U[3]

1: The University of Hong Kong

2: Hong Kong Polytechnic University

3: University of Macau

(Presented by: Edison)

Kernel Aggregation Function and Machine learning Models

Kernel aggregation function: $\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w_i \exp(-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2)$

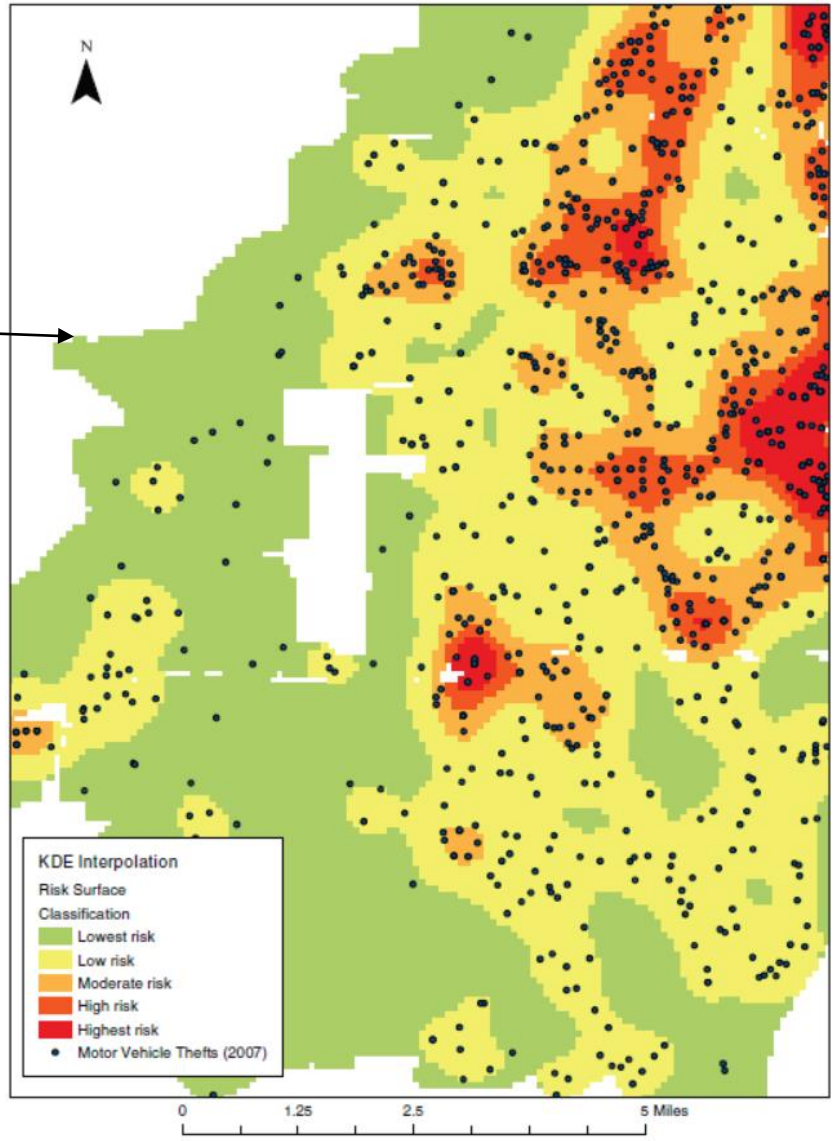
(query: $\mathbf{q}$; dataset: P; weighting: w_i; constant: γ; Euclidean distance: $dist(\mathbf{q}, \mathbf{p_i})^2$)

Type of weighting	**Used in model**	**Application(s)**
Type I: identical, positive w_i (most specific)	Kernel Density Classification	Crime rate prediction Ecological modeling Particle Searching
Type II: positive w_i (subsuming Type I)	1-class SVM	Network fault detection Outlier detection
Type III: no restriction on w_i (subsuming Type II)	2-class SVM	Network fault detection Tumor samples classification Image classification

Kernel Density Classification

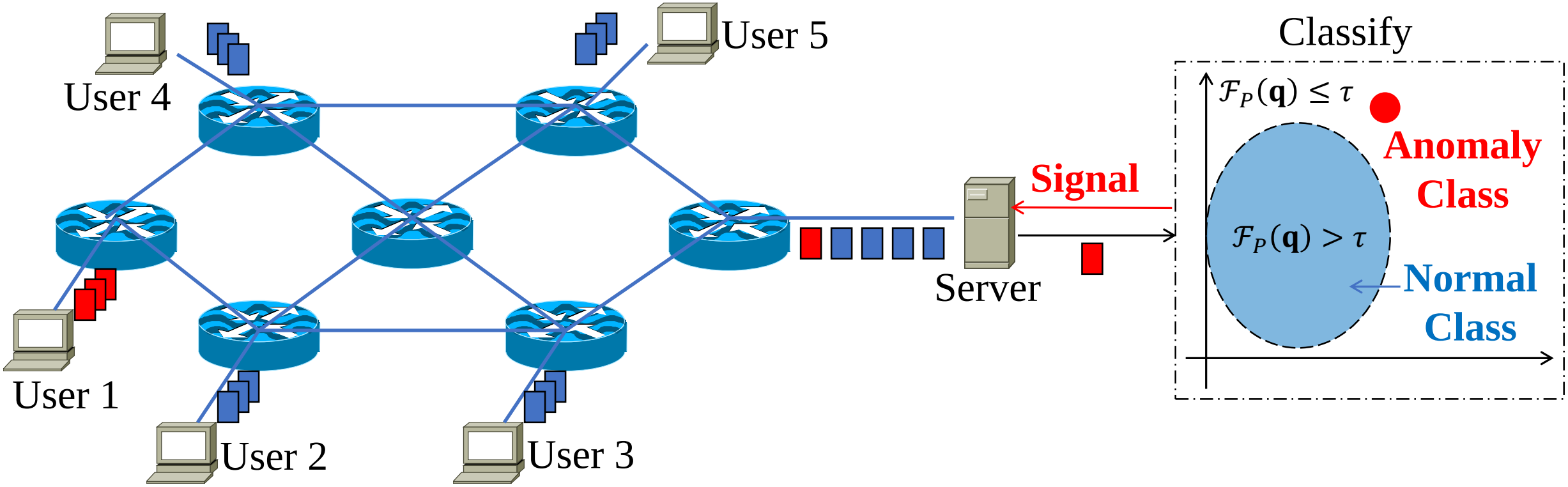
- Black dots (Crimes)
  - Aggravated assault
  - Robbery
  - Commercial burglary
  - motor vehicle theft
- Regions:
  - Crime rates prediction

How to generate color?
$\mathcal{F}_P(\mathbf{q})$



T. Hart and P. Zandbergen "Kernel density estimation and hotspot mapping Examining the influence of interpolation method, grid cell size, and bandwidth on crime forecasting", International Journal of Police Strategies and Management 2014

Kernel Support Vector Machine Classification



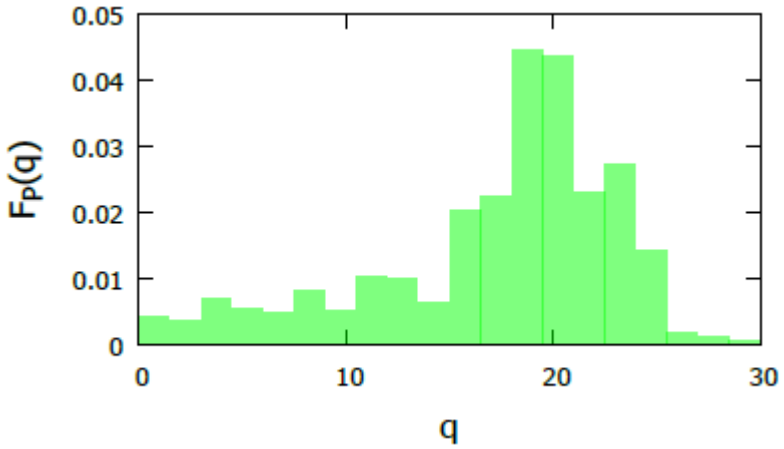
A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys and Tutorials, 2016.

Kernel Aggregation Queries (KAQ)

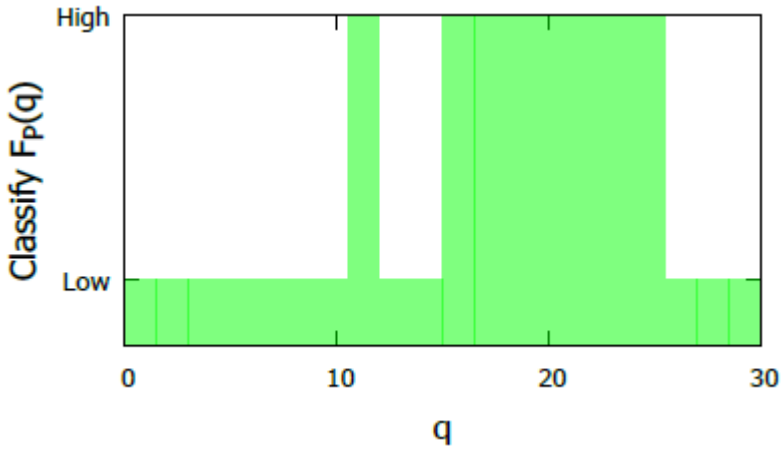
Kernel aggregation function: $\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w_i \exp(-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2)$

Threshold Kernel Aggregation Query (τ-KAQ):

- Input: query vector $\mathbf{q}$, dataset P, threshold τ
- Output: Boolean value
  - high (if $\mathcal{F}_P(\mathbf{q}) \geq \tau$) or low (if $\mathcal{F}_P(\mathbf{q}) < \tau$)



KAQ



τ-KAQ, τ=0.01

Kernel Aggregation Queries are slow!

Le et al. *"Despite their successes, what makes kernel methods difficult to use in many large scale problems is the fact that* ***computing the decision function is typically expensive, especially at prediction time.****"* [1]

$$\mathcal{F}_P(\mathbf{q}) = \underbrace{\sum_{\mathbf{p_i} \in P} w_i \exp(-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2)}_{\text{O(|P| × d) time}}$$

How large are |P| and d?

[1] Quoc V. Le, Tamas Sarlos, and Alexander J. Smola. "Fastfood- computing hilbert space expansions in loglinear time." In ICML2013

Our contribution: KARL

- Kernel Aggregation Rapid Library (KARL)
  - **2.5-738x** speed up over state-of-the-art in different datasets
  - The **Fastest** library in **online prediction phase**

Library	**Support indexing**	**Response time**
LibSVM	no	high
Scikit-learn	yes	high
KARL (this paper)	yes	low

KARL: https://github.com/edisonchan2013928/KARL-Fast-Kernel-Aggregation-Queries
LibSVM: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
Scikit-learn: https://scikit-learn.org/

How to speed up?

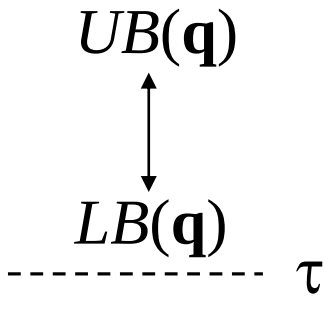
$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w \exp(-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2)$$

$$LB(\mathbf{q}) \leq \mathcal{F}_P(\mathbf{q}) \leq UB(\mathbf{q})$$

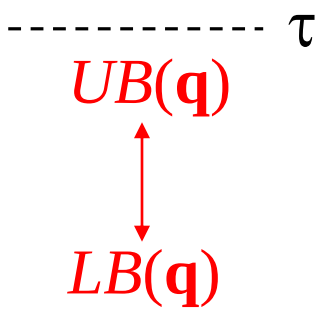
1. **Much faster than O(|P| × d) time**
2. **Should be tight**

Conditions for avoiding $\mathcal{F}_P(\mathbf{q})$ evaluation:

Case 1 ($\mathcal{F}_P(\mathbf{q}) \geq \tau$):

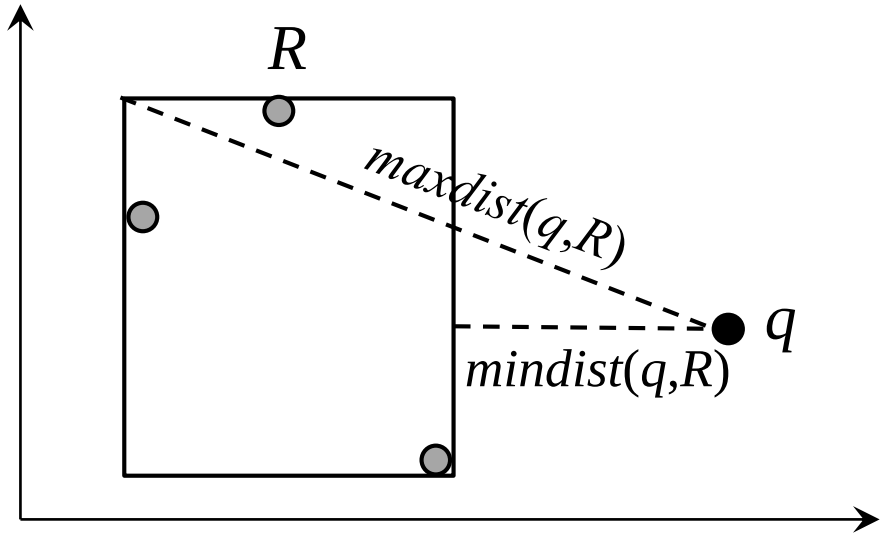


Case 2 ($\mathcal{F}_P(\mathbf{q}) < \tau$):



Existing Work: Bounding Functions

Example: |P|=3



$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w \exp(-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2)$$

$$LB_R(\mathbf{q}) = w \cdot R.\text{count} \cdot \exp(-\gamma \cdot maxdist(\mathbf{q}, R)^2)$$

$$UB_R(\mathbf{q}) = \underbrace{w \cdot R.\text{count} \cdot \exp(-\gamma \cdot mindist(\mathbf{q}, R)^2)}$$

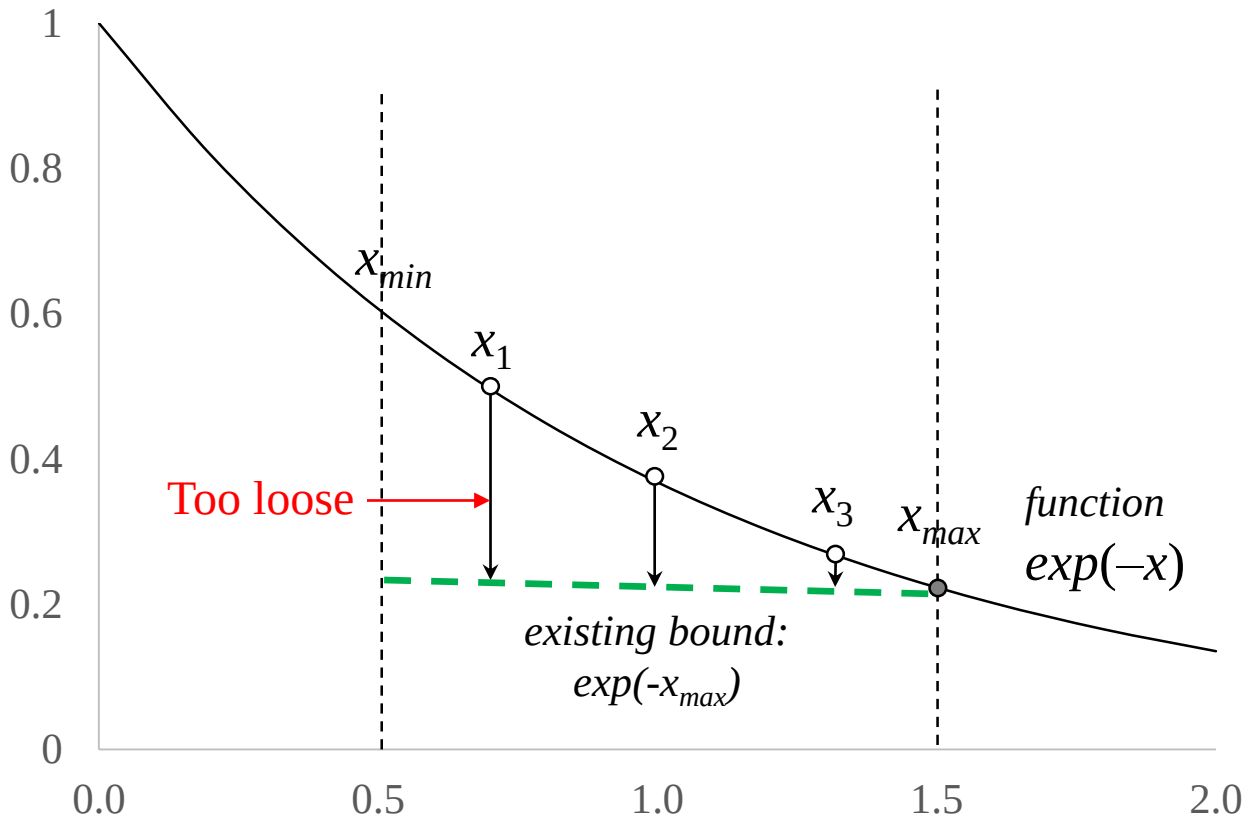
O(d) time

[1] E. Gan and P. Bailis. "Scalable kernel density classification via threshold based pruning" SIGMOD2017
[2] A. G. Gray, A. W. Moore “Nonparametric Density Estimation: Toward Computational Tractability” SDM03

Weakness of Existing Bound Functions

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w \exp(\underbrace{-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2}_{\mathrm{x_i}})$$

$$LB_R(\mathbf{q}) = w \cdot R.\mathrm{count} \cdot \exp(\underbrace{-\gamma \cdot maxdist(\mathbf{q}, R)^2}_{\mathrm{x_{max}}})$$



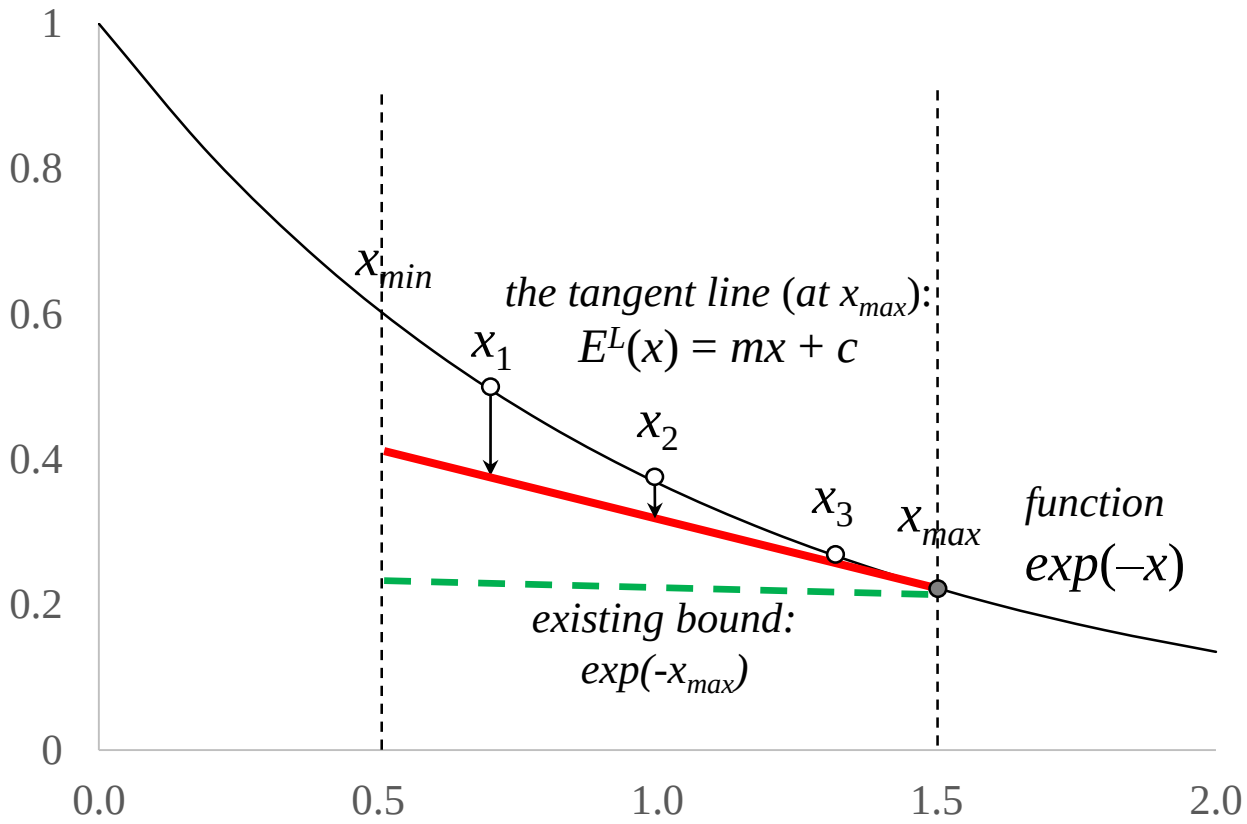
<u>Research Question 1:</u>

Can we develop the tighter lower bound function?

Our Work: Tangent Bound

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w \exp(\underbrace{-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2}_{\mathrm{x_i}})$$

$$LB_R(\mathbf{q}) = w \cdot R.\mathrm{count} \cdot \exp(\underbrace{-\gamma \cdot maxdist(\mathbf{q}, R)^2}_{\mathrm{x_{max}}})$$



Tangent Bound:

$$\mathcal{FL}_P(\mathbf{q}, Lin_{m,c}) = \sum_{\mathbf{p_i} \in P} w\,(m(\underbrace{\gamma dist(\mathbf{q}, \mathbf{p_i})^2}_{\mathrm{x_i}}) + c)$$

Research Question 2:

How to compute this tighter bound quickly?

O(d)-time Tighter Linear Bound

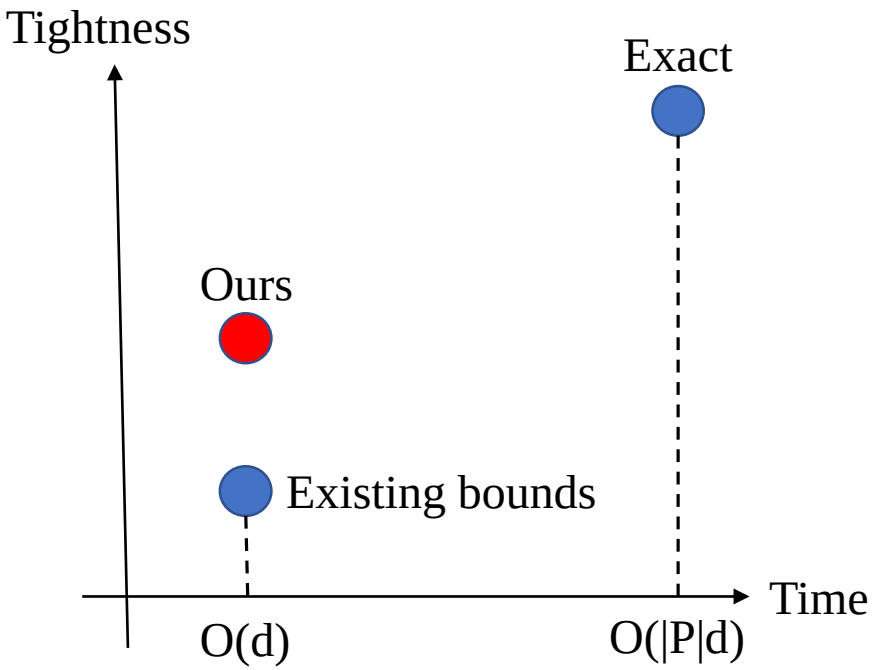
$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p_i} \in P} w \exp(\underbrace{-\gamma \cdot dist(\mathbf{q}, \mathbf{p_i})^2}_{\mathrm{x_i}})$$

Linear bound of $\mathcal{F}_P(\mathbf{q})$:

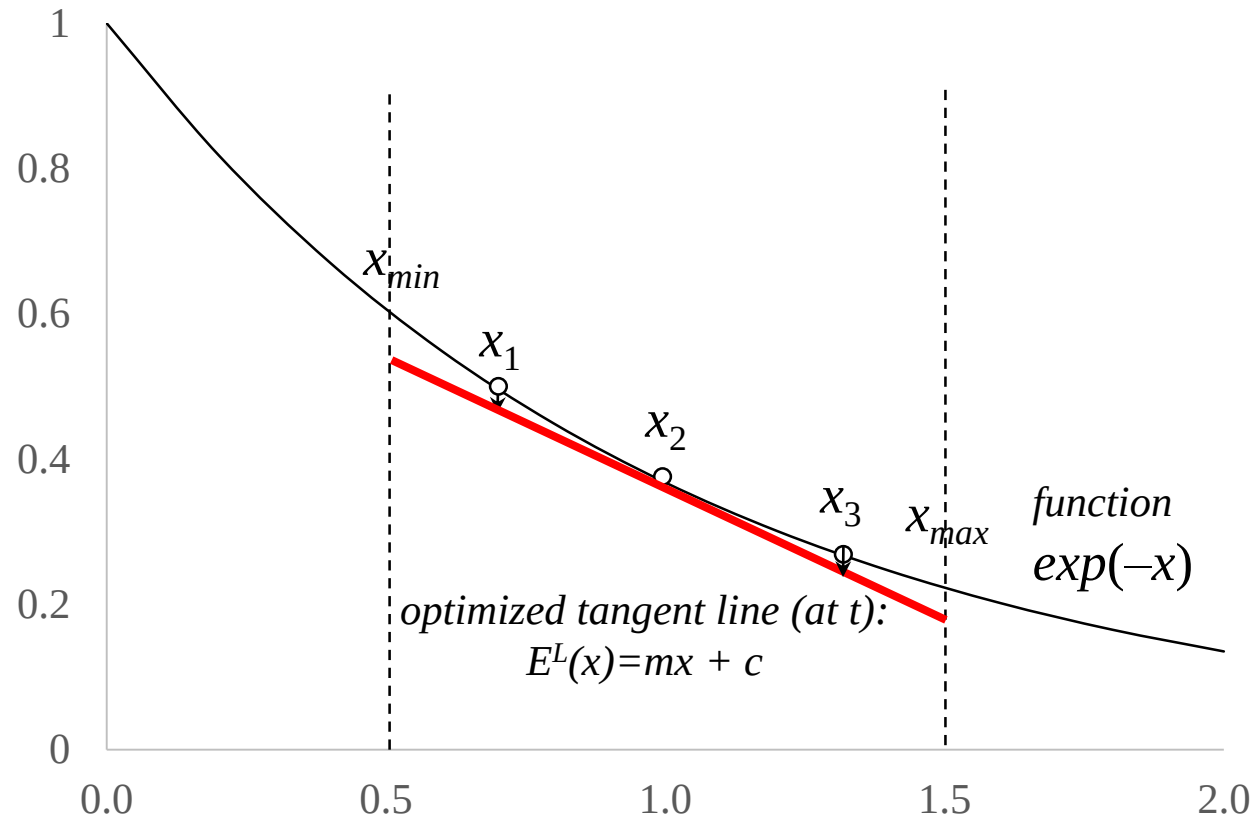
$$\mathcal{FL}_P(\mathbf{q}, Lin_{m,c}) = \sum_{\mathbf{p_i} \in P} w\,(\underbrace{m(\gamma dist(\mathbf{q}, \mathbf{p_i})^2)}_{\mathrm{x_i}} + c)$$

$$\mathcal{FL}_P(\mathbf{q}, Lin_{m,c}) = wm\gamma(|P|\underbrace{\|\mathbf{q}\|^2}_{\mathrm{O(d)}} - 2\underbrace{\mathbf{q} \cdot \mathbf{a_P}}_{\mathrm{O(d)}} + b_P) + wc|P|$$

where $\mathbf{a_P} = \sum_{\mathbf{p_i} \in P} \mathbf{p_i}$ and $b_P = \sum_{\mathbf{p_i} \in P} \|\mathbf{p_i}\|^2$



Advanced Version: Optimized Tangent Bounds



Finding the optimized tangent line is also in O(d) time.

Experimental Results

Methods: SCAN (baseline), LIBSVM, SOTA and KARL
Datasets: UCI Machine Learning Repository/ LibSVM website

Throughput (Queries/sec)

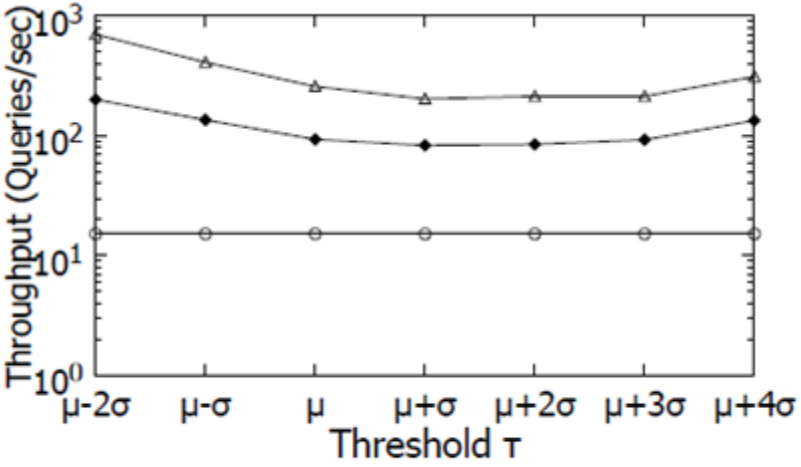
Type	Datasets	SCAN	LIBSVM	SOTA	KARL
I-τ	miniboone	36.1	34	102	**510**
	home	15.2	14.1	93.2	**258**
	susy	2.02	1.86	3.58	**83.4**
II-τ	nsl-kdd	283	481	748	**20668**
	kdd99	260	520	1269	**11324**
	covtype	158	462	448	**6022**
III-τ	ijcnn1	903	1170	1119	**826928**
	a9a	162	610	546	**6885**
	covtype-b	13	38.4	33.9	**274**

SOTA: E. Gan and P. Bailis. "Scalable kernel density classification via threshold based pruning" SIGMOD2017

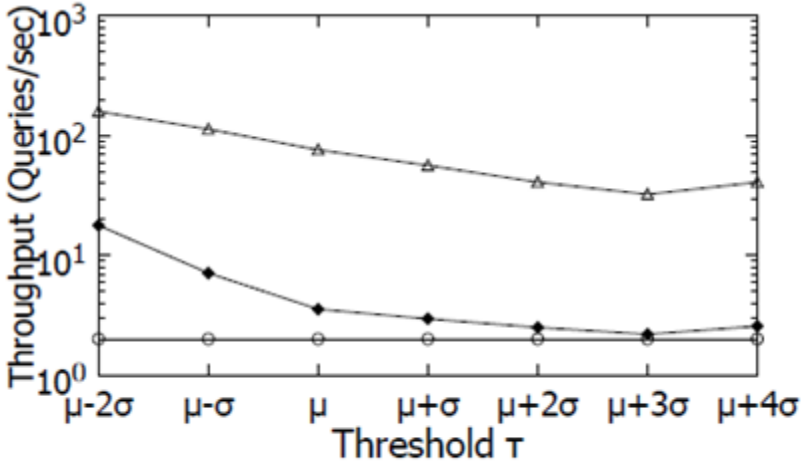
Scalability Experiments (in Type I-τ)

SCAN ○ SOTA ♦ KARL △

Vary seven thresholds

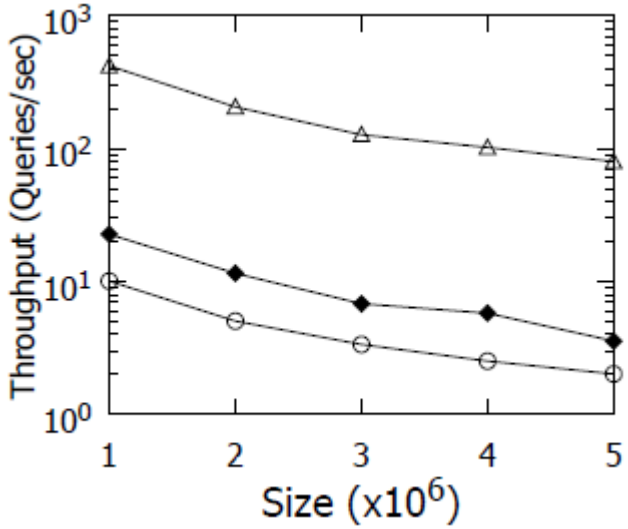


(b) home



(c) susy

Vary data size in dataset susy



Conclusion and Future Work

- What we have done:
  1. Develop Kernel Aggregation Rapid Library (KARL)
     1. Support a wide range of models (in prediction stage)
        1. Kernel Density Classification
        2. One Class SVM
        3. Two Class SVM
     2. Achieve higher throughput than the state-of-the-art by 2.5-738x times
- Future Work:
  1. Integrate the implementation into the library (Ongoing)
  2. Support different types of machine learning models (Ongoing)
  3. Support different types of kernel functions (Ongoing)
  4. Support interesting applications, e.g., Kernel Density Visualization (Ongoing)