MINIMUM ENCLOSING BALL FOR ANOMALY DETECTION USING FRANK-WOLFE VARIANTS

Project by :

- Dejan Dichoski
- Marija Cveevska
- Suleyman Erim

Mentor :

- Prof. F. Rinaldi

University of Padova

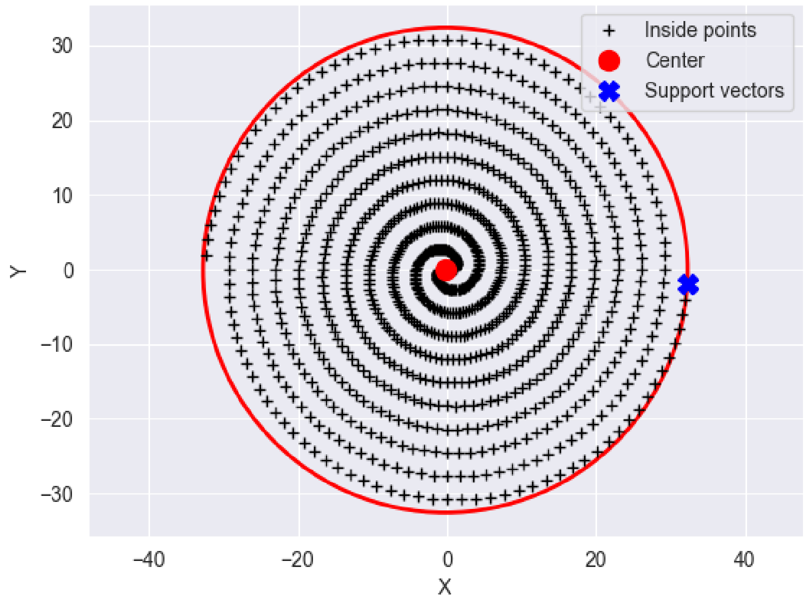
INTRODUCTION

- **Applications of MEB**: clustering, nearest neighbor search, data classification, SVM, facility location, collision detection, computer graphics, **anomaly detection**.
- **Approach**: formalize the problem in terms of constrained quadratic optimization, and solve using FW variants.

We will implement 3 algorithms with the goal of solving the MEB problem and we will test them on artificial and real-world datasets for detecting anomalies. Finally, we will compare our results.

THE PRIMAL MEB PROBLEM

$$\boxed{c^*, r^* = \underset{c,r}{\operatorname{argmin}}\ r^2 \quad s.t.\ \|a_i - c\|^2 - r^2 \le 0, \qquad i = 1, \ldots, m}$$



THE DUAL MEB PROBLEM

$$\boxed{\begin{aligned} u^* = \underset{u}{\operatorname{argmin}} -\Phi(u) = \underset{u}{\operatorname{argmin}}\, u^T A^T A u - u^T z \\ s.t.\quad u^T 1 = 1, u \ge 0 \end{aligned}}$$

FROM LAGRANGE MULTIPLIERS TO MEB PARAMETERS

$$\boxed{c^* = Au^* = \sum_{i=1}^{m} a_i u_i^*}$$

$$\boxed{r^* = \sqrt{-\Phi(u^*)}}$$

ALGORITHMS

FRANK-WOLFE

Frank Wolfe algorithm provides a straightforward approach for solving a convex minimization problem over a compact convex set.

AWAY-STEPS FW

A simple improvement over the standard FW that deals with the zig-zagging problem by introducing the possibility of taking “away steps”.

BLENDED PAIRWISE CONDITIONAL GRADIENT

Combination of the Pairwise CG with the blending criterion from the Blended CG that eliminates the occurrence of swap steps.

$(1+\varepsilon)$-APPROXIMATION TO MEB

Adaptation of the standard FW to the MEB problem. It generates a sequence of increasing balls until a ball with desired properties is computed.

AWAY-STEPS FRANK-WOLFE

Algorithm 1: Away-steps Frank-Wolfe algorithm

$\boldsymbol{u^0} \leftarrow [\mathbf{1\ 0\ 0} \dots \mathbf{0}] \in \mathbb{R}^m$, $\boldsymbol{S^0} \leftarrow \{\boldsymbol{u^0}\}$ *// so that $\omega_{\boldsymbol{v}}^0 = 1$ for $\boldsymbol{v} = \boldsymbol{u^0}$ and 0 otherwise*
For $t = 0, \dots, maxIter - 1$ **do**
 $\boldsymbol{s^t} \leftarrow \mathrm{LMO}_A\big(\nabla\Phi(\boldsymbol{u^t})\big)$ and $d\_FW^t \leftarrow \boldsymbol{s^t} - \boldsymbol{u^t}$ *// the FW direction*
 $\boldsymbol{v^t} \leftarrow \underset{\boldsymbol{v} \in \boldsymbol{S^t}}{\mathrm{argmax}} \langle \nabla\Phi(\boldsymbol{u^t}),\ \boldsymbol{v}\rangle$ and $d\_A^t \leftarrow \boldsymbol{u^t} - \boldsymbol{v^t}$ *// the away direction*
 if $(g\_FW^t \leftarrow \langle -\nabla\Phi(\boldsymbol{u_t}),\ \boldsymbol{d\_FW^t}\rangle) \le \varepsilon$ **then return** $\boldsymbol{u^t}$
 if $\langle -\nabla\Phi(\boldsymbol{u_t}),\ \boldsymbol{d\_FW^t}\rangle \ge \langle -\nabla\Phi(\boldsymbol{u^t}), \boldsymbol{d\_A^t}\rangle$ **then**
  $\boldsymbol{d^t} \leftarrow \boldsymbol{d\_FW^t}$, and $\alpha_{max} = 1$ *// choose the FW direction*
 else
  $\boldsymbol{d^t} \leftarrow \boldsymbol{d\_A^t}$, and $\alpha_{max} = \frac{\omega_{v^t}}{1 - \omega_{v^t}}$ *// choose away direction; maximum feasible step-size*
 end if
 Line-search: $\alpha^t \in \underset{\alpha \in [0, \alpha_{max}]}{\mathrm{argmin}} \Phi(\boldsymbol{u^t} + \alpha \boldsymbol{d^t})$
 $\boldsymbol{u^t} \leftarrow \boldsymbol{u^{t-1}} + \alpha^t \boldsymbol{d^t}$ *// and accordingly for the weights $\boldsymbol{\omega^t}$*
 $\boldsymbol{S^t} \leftarrow \{\boldsymbol{v}\ \mathrm{s.t.}\ \omega_{\boldsymbol{v}}^t > 0\}$
end for

BLENDED PAIRWISE CONDITIONAL GRADIENT

Algorithm 2: Blended Pairwise Conditional Gradients (BPCG)

$\boldsymbol{u}^0 \leftarrow [\mathbf{1\ 0\ 0} \ldots \mathbf{0}] \in \mathbb{R}^m$, $\boldsymbol{S}^0 \leftarrow \{\boldsymbol{u}^0\}$
for $t = 0, \ldots, maxIter - 1$ **do**:
  $a^t \leftarrow \underset{v \in S^t}{\operatorname{argmax}} \langle \nabla\Phi(\boldsymbol{u}^t),\ \boldsymbol{v} \rangle$ — *// away vertex*
  $s^t \leftarrow \underset{v \in S^t}{\operatorname{argmax}} \langle \nabla\Phi(\boldsymbol{u}^t),\ \boldsymbol{v} \rangle$ — *// local FW*
  $w^t \leftarrow \operatorname{argmax} \nabla\Phi(\boldsymbol{u}^t)$ — *// global FW*

Indices to find.

  if $\langle \nabla\Phi(\boldsymbol{u}^t), \boldsymbol{a}^t - \boldsymbol{s}^t \rangle \geq \langle \nabla\Phi(\boldsymbol{u}^t), \boldsymbol{u}^t - \boldsymbol{w}^t \rangle$ **then**

local pairwise gap >= FW gap

    $\boldsymbol{d}^t = \boldsymbol{a}^t - \boldsymbol{s}^t$
    $\alpha_{max} \leftarrow \boldsymbol{u}^t[a_t]$
    $\alpha^t \leftarrow \underset{\alpha \in [0, \alpha_{max}]}{\operatorname{argmax}} \Phi(\boldsymbol{u}^t - \alpha \boldsymbol{d}^t)$
    if $\alpha^t < \alpha_{max}$ **then**
      $\boldsymbol{S}^t \leftarrow \boldsymbol{S}^{t-1}$ — *// descent step*
    else
      $\boldsymbol{S}^t \leftarrow \boldsymbol{S}^{t-1} \setminus \{\boldsymbol{a}^t\}$ — *// drop step*
    end if

The weights of the active atoms in St are optimized by the PCG locally.
Take a pairwise step = drop / descent.

  else
    $\boldsymbol{d}^t = \boldsymbol{u}^t - \boldsymbol{w}^t$
    $\alpha^t \leftarrow \underset{\alpha \in [0,1]}{\operatorname{argmin}} \Phi(\boldsymbol{u}^t - \alpha \boldsymbol{d}^t)$
    $\boldsymbol{S}^t \leftarrow \boldsymbol{S}^{t-1} \cup \{\boldsymbol{w}^t\}$ (or $\boldsymbol{S}^t \leftarrow \{\boldsymbol{w}^t\}$ **if** $\alpha^t = 1$) — *// FW step*
  end if

The local pairwise gap is smaller than the FW gap => Take a FW step.

  $\boldsymbol{u}^t \leftarrow \boldsymbol{u}^{t-1} - \alpha^t \boldsymbol{d}^t$
end for

(1+ ε)-APPROXIMATION TO MEB

Algorithm 3: $(\mathbf{1}+\boldsymbol{\varepsilon})$-approximation to MEB

$p \leftarrow \underset{i=1,\dots,n}{\operatorname{argmax}} \|u_i - u_1\|^2,\ q \leftarrow \underset{i=1,\dots,n}{\operatorname{argmax}} \|u_i - u_p\|^2$

$\boldsymbol{u}^{\mathbf{0}} \leftarrow \mathbf{0}$

$u_p^0 \leftarrow \frac{1}{2},\ u_q^0 \leftarrow \frac{1}{2}$ // Feasible solution

$\boldsymbol{S}^{\mathbf{0}} \leftarrow \{\boldsymbol{a}_p, \boldsymbol{a}_q\}$ // Assign core set

$\boldsymbol{c}^{\mathbf{0}} \leftarrow \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} u_i^0 \boldsymbol{a}_{\boldsymbol{i}} = \langle \boldsymbol{u}^{\mathbf{0}}, \boldsymbol{a} \rangle$ // Assign center

$\gamma^0 \leftarrow \boldsymbol{\Phi}(\boldsymbol{u}^{\mathbf{0}})$ // Assign r^2

$\kappa \leftarrow \underset{i=1,\dots,n}{\operatorname{argmax}} \|\boldsymbol{a}_{\boldsymbol{i}} - \boldsymbol{c}^{\mathbf{0}}\|^2$ // Find furthest point index

$\delta^0 \leftarrow (\|\boldsymbol{a}_{\boldsymbol{\kappa}} - \boldsymbol{c}^{\mathbf{0}}\|^2 / \gamma^0) - 1$ // Find error bound

$t \leftarrow 0$

While $\delta^t > (1+\varepsilon)^2 - 1$ and $t < maxIter$ **do**

loop

  $\alpha^t \leftarrow \delta^t / [2(1+\delta^t)]$ // Update learning rate

  $t \leftarrow t+1$

  $\boldsymbol{u}^{\boldsymbol{t}} \leftarrow (1-\alpha^{t-1})\boldsymbol{u}^{\boldsymbol{t-1}} + \alpha^{t-1}\boldsymbol{e}_\kappa$ // Update Lagrangian Multipliers

  $\boldsymbol{c}^{\boldsymbol{t}} \leftarrow (1-\alpha^{t-1})\boldsymbol{c}^{\boldsymbol{t-1}} + \alpha^{t-1}\boldsymbol{a}_\kappa$ // Update center

  $\boldsymbol{S}^{\boldsymbol{t}} \leftarrow \boldsymbol{S}^{\boldsymbol{t}} \cup \{\boldsymbol{a}_{\boldsymbol{\kappa}}\}$ // Update core set

  $\gamma^t \leftarrow \boldsymbol{\Phi}(\boldsymbol{u}^{\boldsymbol{t}})$ // Update r^2

  $\kappa \leftarrow \underset{i=1,\dots,n}{\operatorname{argmax}} \|\boldsymbol{a}_{\boldsymbol{i}} - \boldsymbol{c}^{\boldsymbol{t}}\|^2$ // Find furthest point index

  $\delta^t \leftarrow (\|\boldsymbol{a}_{\boldsymbol{\kappa}} - \boldsymbol{c}^{\boldsymbol{t}}\|^2 / \gamma^t) - 1$ // Find error bound

end loop

LINE SEARCH STRATEGIES

INVERSE TIME DECAY

$$\frac{t}{t+2}$$

Initial strategy, satisfactory radius and center parameters, but didn't converge.

GOLDEN SECTION SEARCH

Good results with Algorithm 1 but didn't converge with Algorithm 2.

ARMIJO'S RULE

Even though the obtained results for the center and radius were close to the optimal ones, it was evident that this line search strategy is very slow, and often failed to converge.

EXACT LINE SEARCH

Best results, fast convergence for both algorithms.

$$\boxed{\alpha = -\frac{\nabla\Phi(\boldsymbol{u}^t)^T \boldsymbol{d}^t}{2\boldsymbol{d}^{t^T} \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{d}^t}}$$

EXPERIMENTS

- **Goal**: Assess the quality of the three algorithms.

Hyperparameter: epsilon (ε) – stopping criteria

- Methodology: Training and testing the algorithms on:

DATASETS

SYNTHETIC DATA:

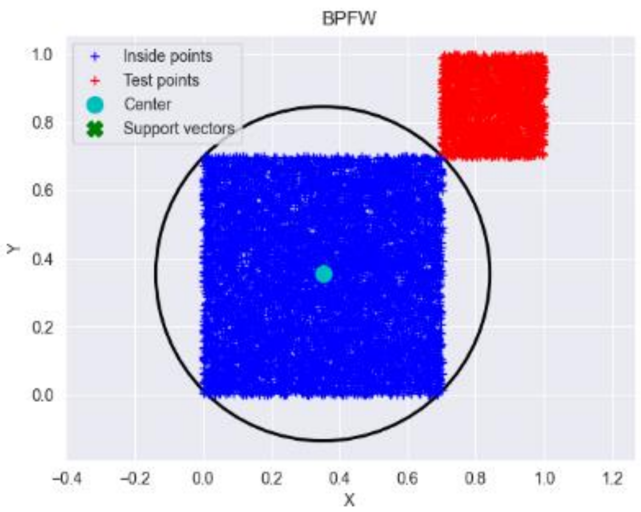
- Uniform data
- Gaussian data

REAL-WORLD DATA:

- Breast Cancer Wisconsin dataset
- Customer Churn dataset

EXPERIMENT: UNIFORM DATASET

	TRAIN				TEST	
ALGORITHM	**ITERATIONS**	**CPU TIME (S)**	**RADIUS**	**ACTIVE SET SIZE**	**RECALL (%)**	**F1 SCORE (%)**
ASFW	379	23.632	1.019375	15	100	100
BPCD	272	16.606	1.019365	14	100	100
APPFW	747	22.108	1.019697	18	100	100

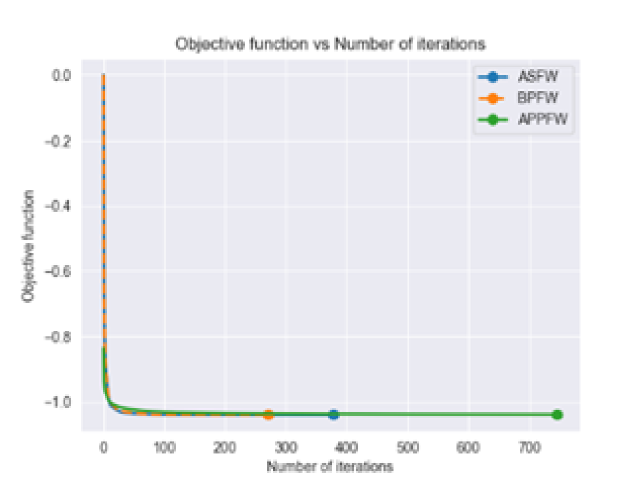


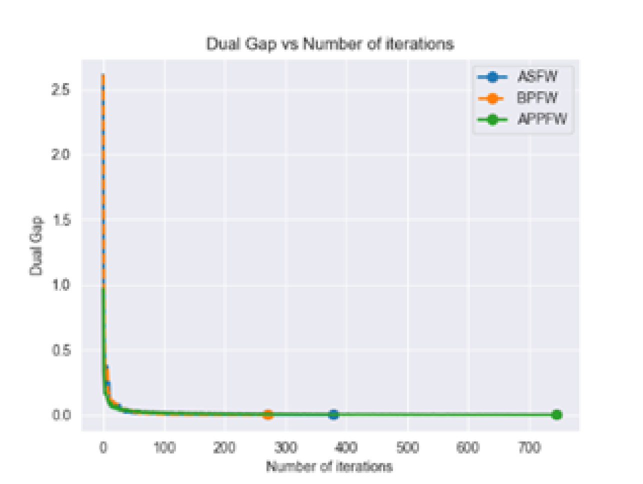
We created two closely spaced yet separable clusters:

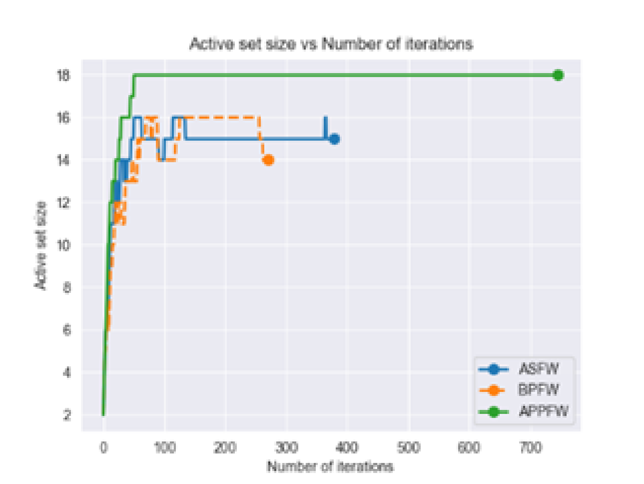
- Train (nominal data): 8000 data points ~ U[0.0, 0.7);
- Test (anomaly data): 2000 points ~ U[0.7, 1.0).

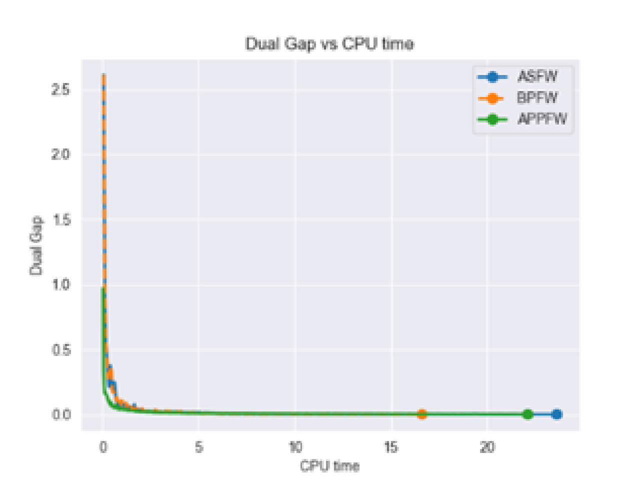
Blue cluster = training data.
Red cluster = test points.
The circle shows the MEB constructed by Algorithm 2.

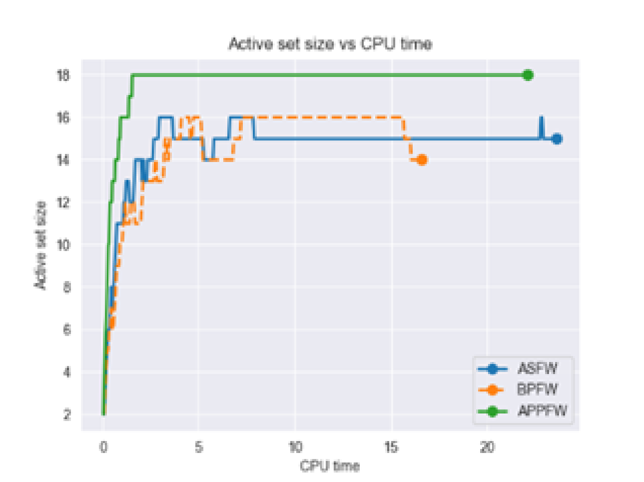
TRAINING RESULTS - UNIFORM DATASET





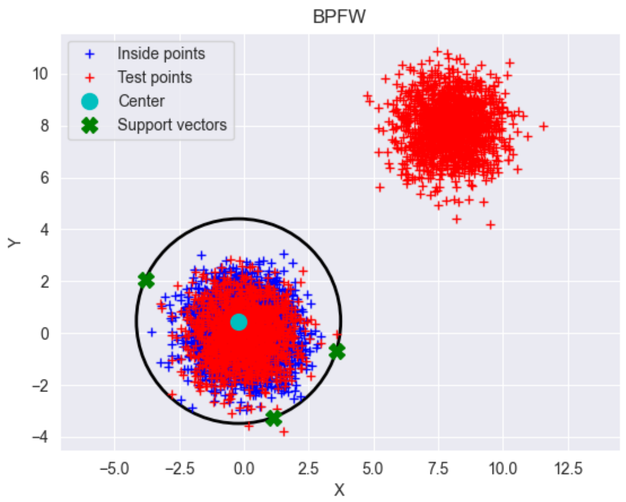






EXPERIMENT: GAUSSIAN DATASET

	TRAIN				TEST	
ALGORITHM	**ITERATIONS**	**CPU TIME (S)**	**RADIUS**	**ACTIVE SET SIZE**	**RECALL (%)**	**F1 SCORE (%)**
ASFW	177	9.614	5.45032	9	100	99.85
BPCG	130	7.219	5.45032	9	100	99.85
APPFW	733	21.904	5.45191	12	100	99.85

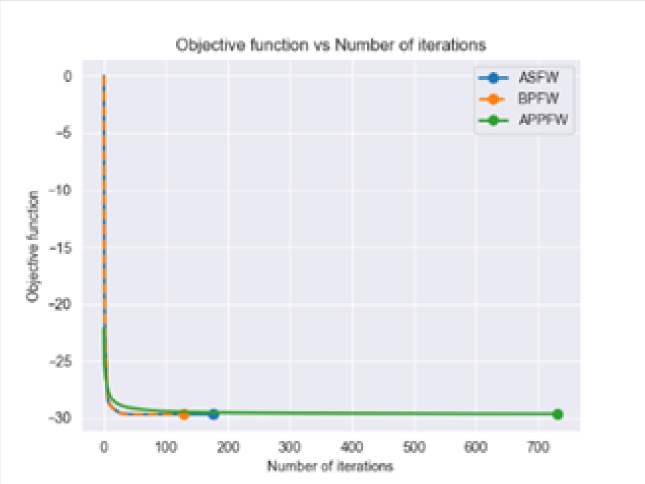


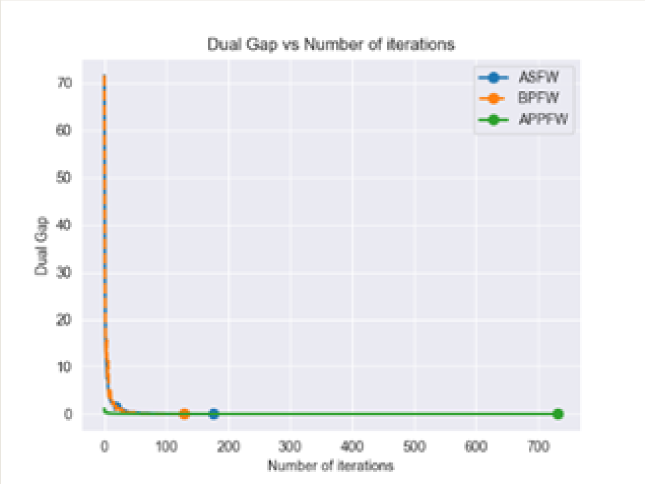
We created two very separable clusters:

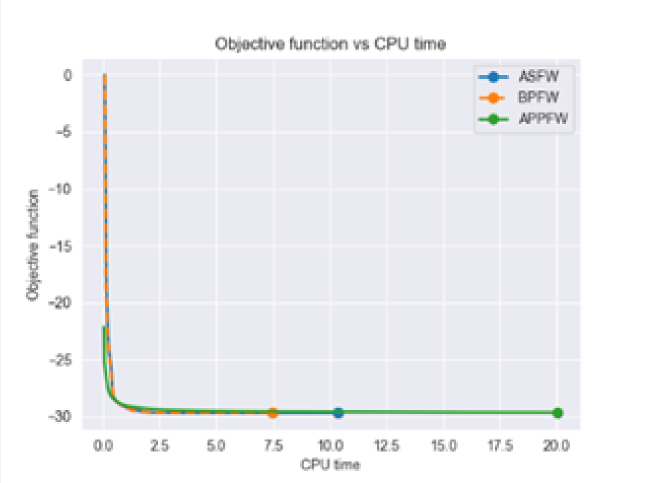
- **Training (nominal data)**: 8000 data points ~ N(0, 1);
- **Testing (nominal + anomaly data)**: 1000 points ~ N(0, 1), and 1000 ~ N(7, 1).

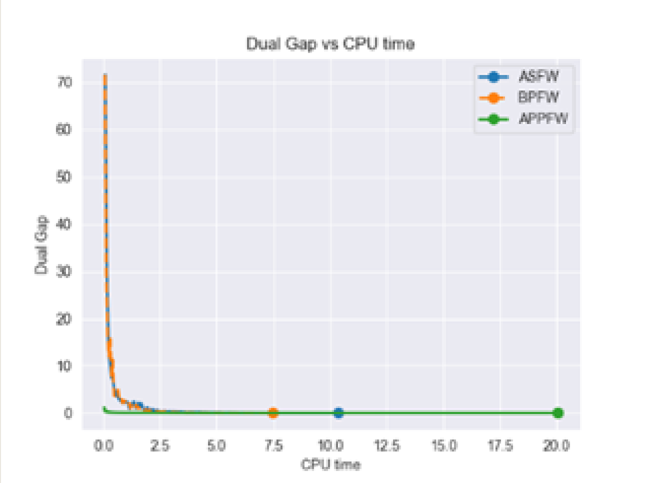
Blue cluster = training points.
Red cluster = test points:
- nominal: around the center of the MEB
- anomaly: the red cluster on the top right.
The circle shows the MEB constructed by Algorithm 2.

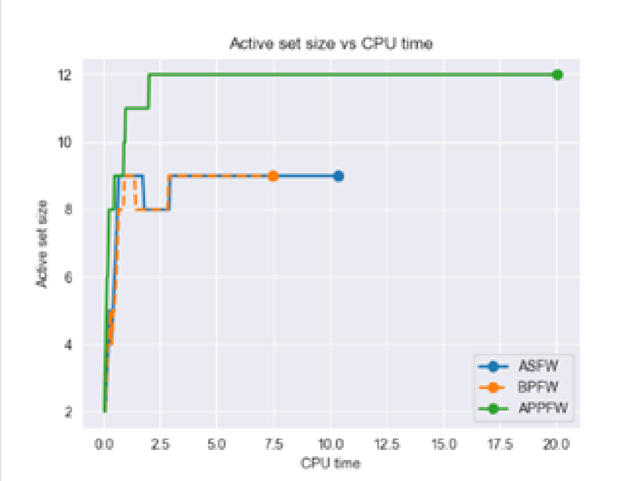
TRAINING RESULTS: GAUSSIAN DATASET











EXPERIMENT: BREAST CANCER WISCONSIN DATASET

The Breast Cancer Wisconsin dataset consists of: **569 samples** and **32 features.**

To create training and testing datasets, we first separated the two classes:

- **nominal data = 357 benign cases;**
- **anomaly data = 212 malignant cases.**

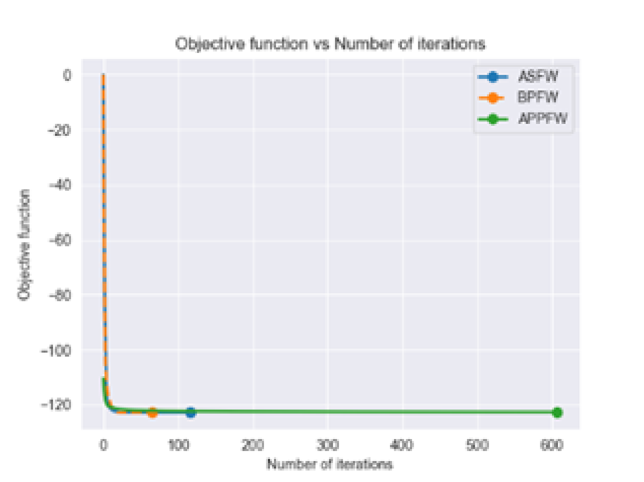
❑ **Training data**: Half of the nominal cases (178 samples).
❑ **Testing data**: Nominal (179) + anomaly (212) samples = 391 samples.

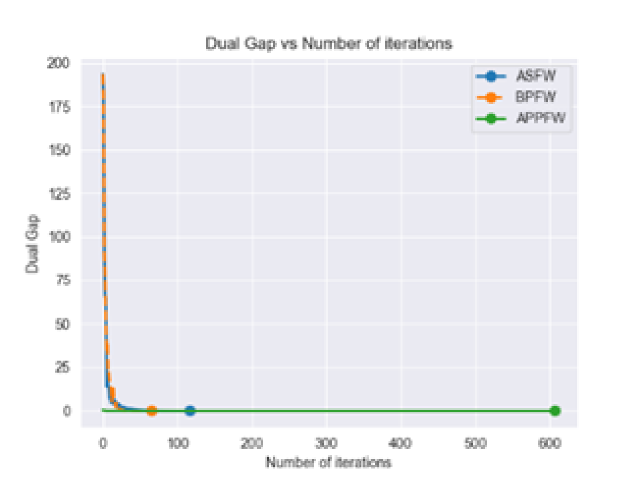
Various thresholds (ε) were employed as stopping criteria (next slide).

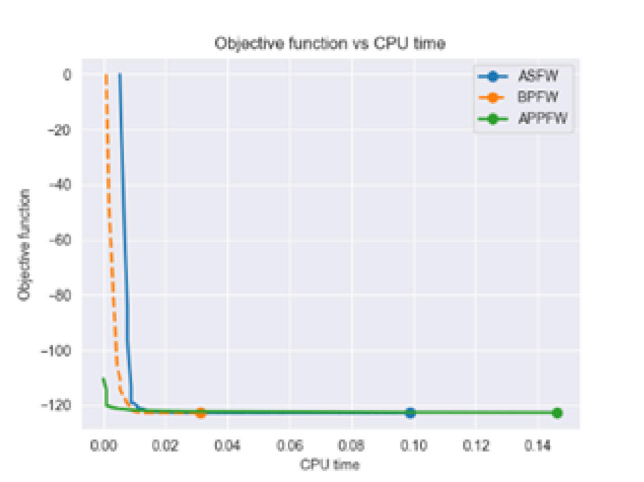
EXPERIMENT: BREAST CANCER WISCONSIN

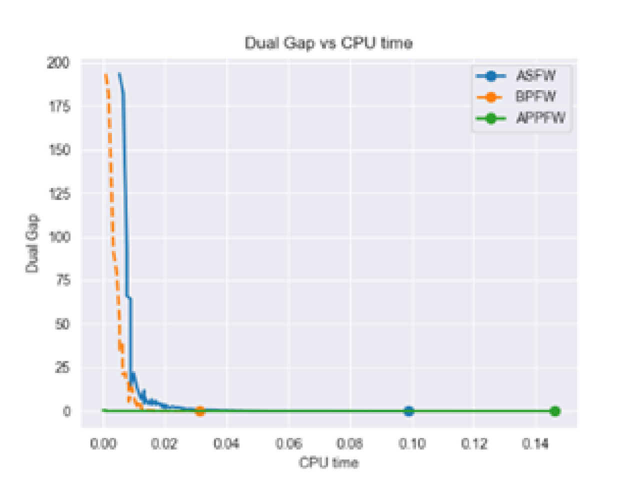
	TRAIN					TEST	
ALGORITHM	ε	ITERATIONS	CPU TIME (ms)	RADIUS	ACTIVE SET SIZE	RECALL (%)	F1 SCORE (%)
ASFW	0.1	64	26.38	11.077366	6	76.415	85.488
	0.01	92	36.97	11.077386	6	76.415	85.488
	0.001	118	98.634	11.077386	6	76.415	85.488
BPFW	0.1	32	18.165	11.077378	6	76.415	85.488
	0.01	50	24.112	11.077386	6	76.415	85.488
	0.001	66	31.402	11.077386	6	76.415	85.488
APPFW	0.1	4	2.004	11.758557	5	72.17	83.152
	0.01	35	11.0	11.130932	7	76.415	85.488
	0.001	608	146.278	11.080393	7	76.415	85.488

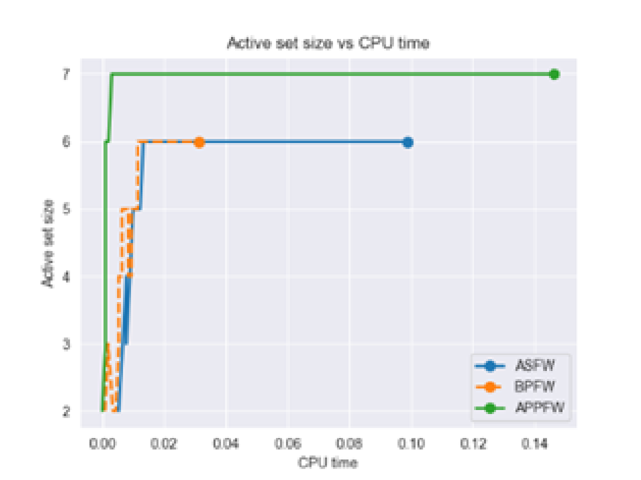
TRAINING RESULTS: BREAST CANCER DATASET











EXPERIMENT: CUSTOMER CHURN DATASET

The Iranian Churn Dataset consists of **3,150 samples** and **14 features:**

To create training and testing datasets, we first separated the two classes:

- **nominal data = 2655 retention cases;**
- **anomaly data = 495 churn cases.**

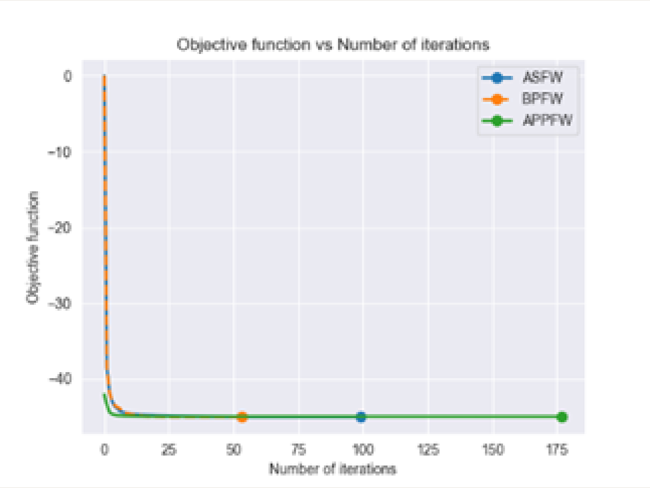
- ❑ **Training data**: Half of the nominal cases (1327 samples).
- ❑ **Testing data**: Nominal (1327) + anomaly (495) samples = 1823 samples.

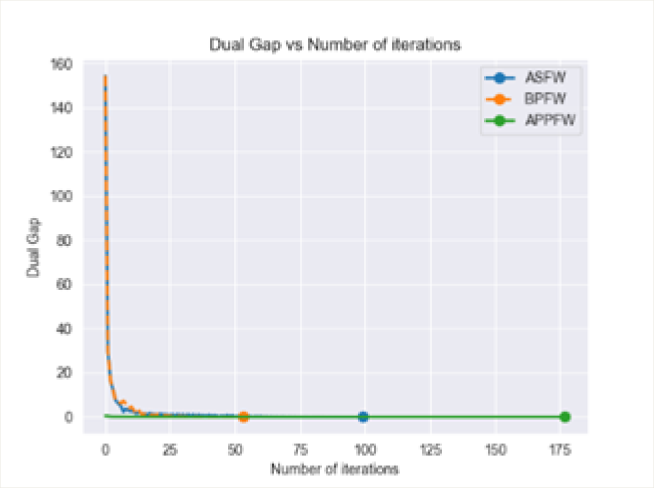
Various thresholds (ε) were employed as stopping criteria (next slide).

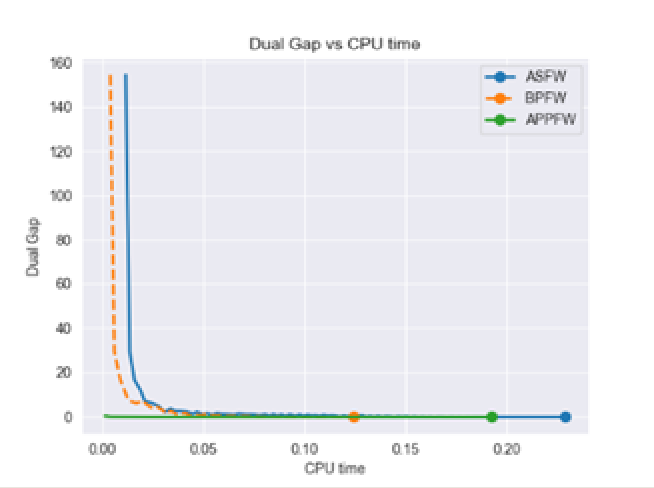
EXPERIMENT: CUSTOMER CHURN DATASET

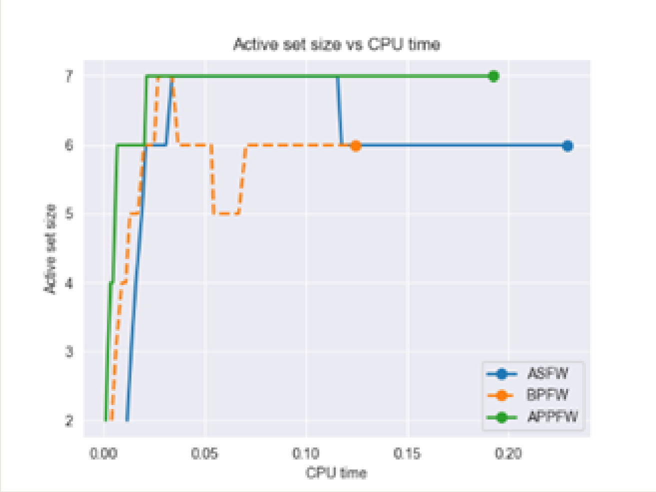
	TRAIN					TEST	
ALGORITHM	ε	ITERATIONS	CPU TIME (ms)	RADIUS	ACTIVE SET SIZE	RECALL (%)	F1 SCORE (%)
ASFW	0.1	55	131.633	6.708818	6	10.101	18.282
	0.01	80	154.788	6.709065	6	9.697	17.615
	0.001	100	217.184	6.709067	6	9.697	17.615
BPFW	0.1	32	74.47	6.709004	6	9.495	17.248
	0.01	40	97.306	6.709067	6	9.697	17.615
	0.001	54	116.559	6.709067	6	9.697	17.615
APPFW	0.1	3	5.293	7.184624	4	0.808	1.6
	0.01	13	14.184	6.753838	6	8.283	15.27
	0.001	178	178.887	6.711496	7	9.697	17.615

TRAINING RESULTS: CUSTOMER CHURN DATASET









CONCLUSION

We implemented 3 adaptations of the Frank-Wolfe algorithm:

1. **Away-steps FW** - satisfactory results, highlighting its improvement over the vanilla FW algorithm.
2. **BPCG** - consistently outperformed the others in terms of metrics such as iterations, CPU time, and active set size, displaying a superior convergence rate.
3. **(1+ε)-approx. to MEB** - initially underperformed with a strict stopping criterion but showed significant improvement with a larger ε value.

Ultimately, our evaluation on different datasets emphasized the importance of selecting optimization algorithms based on the dataset's characteristics, as the algorithm's performance is highly dependent on data structure.

THANK YOU
FOR YOUR ATTENTION

You can find the project at the following link.