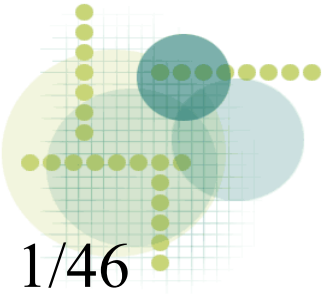
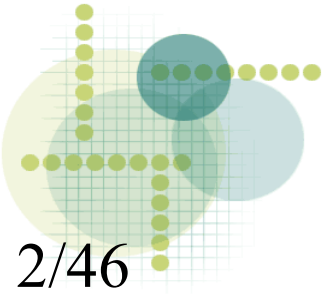
Kombinacja jądrowych estymatorów gęstości w klasyfikacji – wstępne wyniki

Mateusz Kobos, 10.12.2008
Seminarium Metody Inteligencji Obliczeniowej

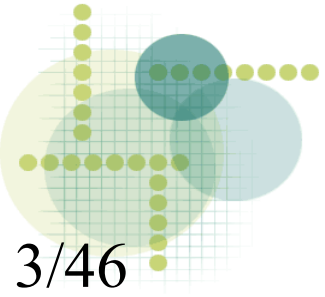


Spis treści



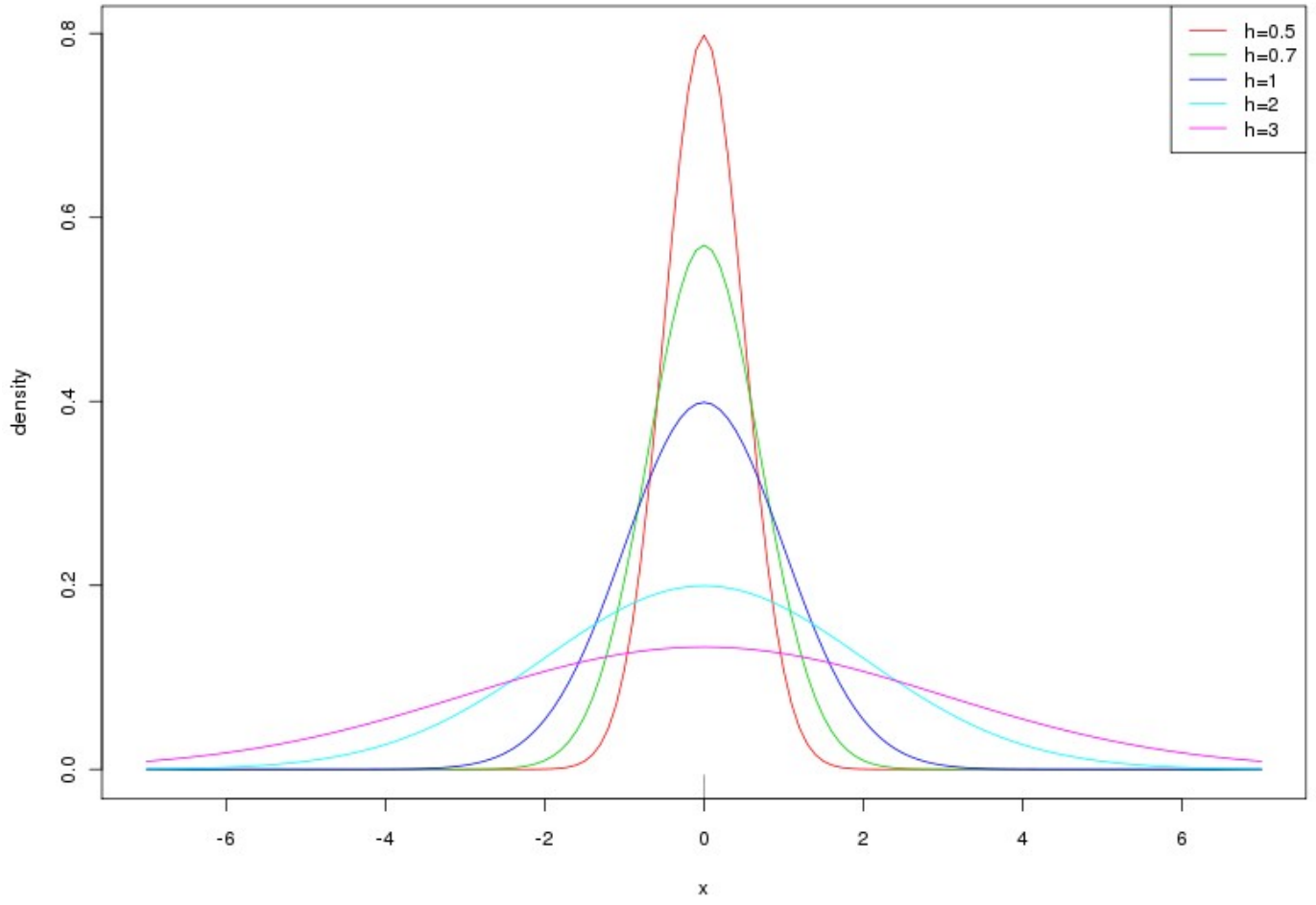
Działanie algorytmu

1. **Uczenie** na zbiorze uczącym: stwórz estymator gęstości dla każdej z klas
2. **Odtwarzanie**/klasyfikacja punktu testowego: oblicz prawd. przynależności do każdej z klas za pomocą wzoru Bayes'a



Jądra

- Dokonujemy estymacji jądrowej dla różnych szerokości jądra i uśredniamy

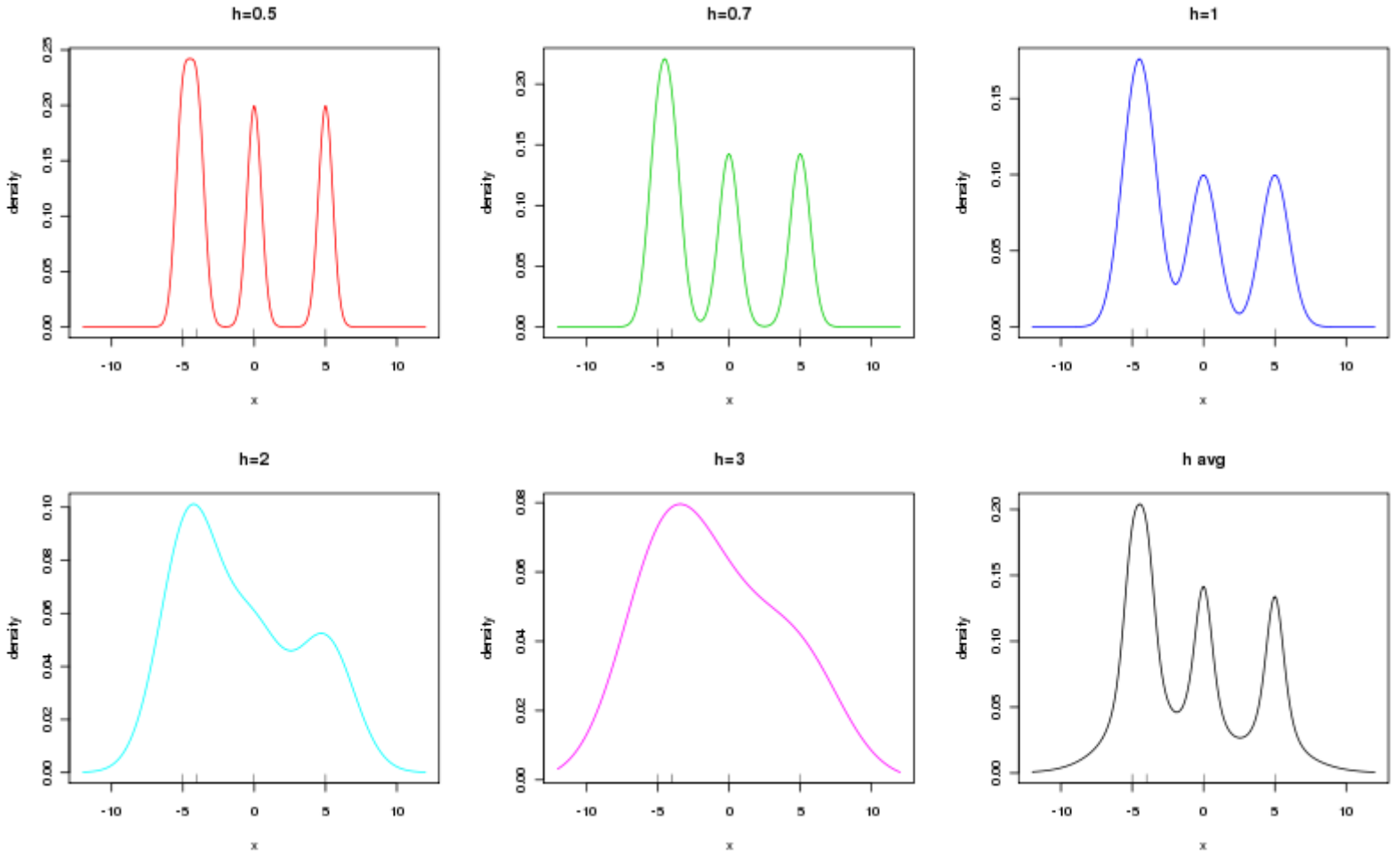


Jądra a preprocessing

- Jeśli punkty poddaliśmy preprocessingowi, to jest to równoważne zmianie kształtu jąder (które oryginalnie są kulami)
  - Standardyzacja
    - Elipsoida o osiach równoległych do osi układu współrzędnych
  - PCA
    - Elipsoida
- Dzięki temu (teoretycznie) możemy lepiej dopasować model do danych

Estymacja gęstości

- Estymacje gęstości dla różnych szerokości jąder



Wzór Bayesa

- Gdy mamy estymacje gęstości dla każdej z klas, korzystamy ze wzoru Bayesa

Klasyfikator Bayesowski:

$$P(c = i|x) = \frac{p(x|c = i)P(c = i)}{p(x)}$$

gdzie:

- $P(c = i|x)$ - prawdopodobieństwo, że punkt x należy do klasy i
- $p(x|c = i)$ - gęstość dla rozkładu punktów z klasy i w punkcie x
- $P(c = i)$ - prawdopodobieństwo pojawienia się punktu z klasy i
- $p(x) = \sum_{j=1}^{K} p(x|c = j)P(c = j)$ - gęstość w punkcie x
  - K - liczba klas

Uczenie

- Uczenie/dopasowywanie modelu do zbioru uczącego X

Algorithm 1 KDEC-Training($\mathbf{X}$)

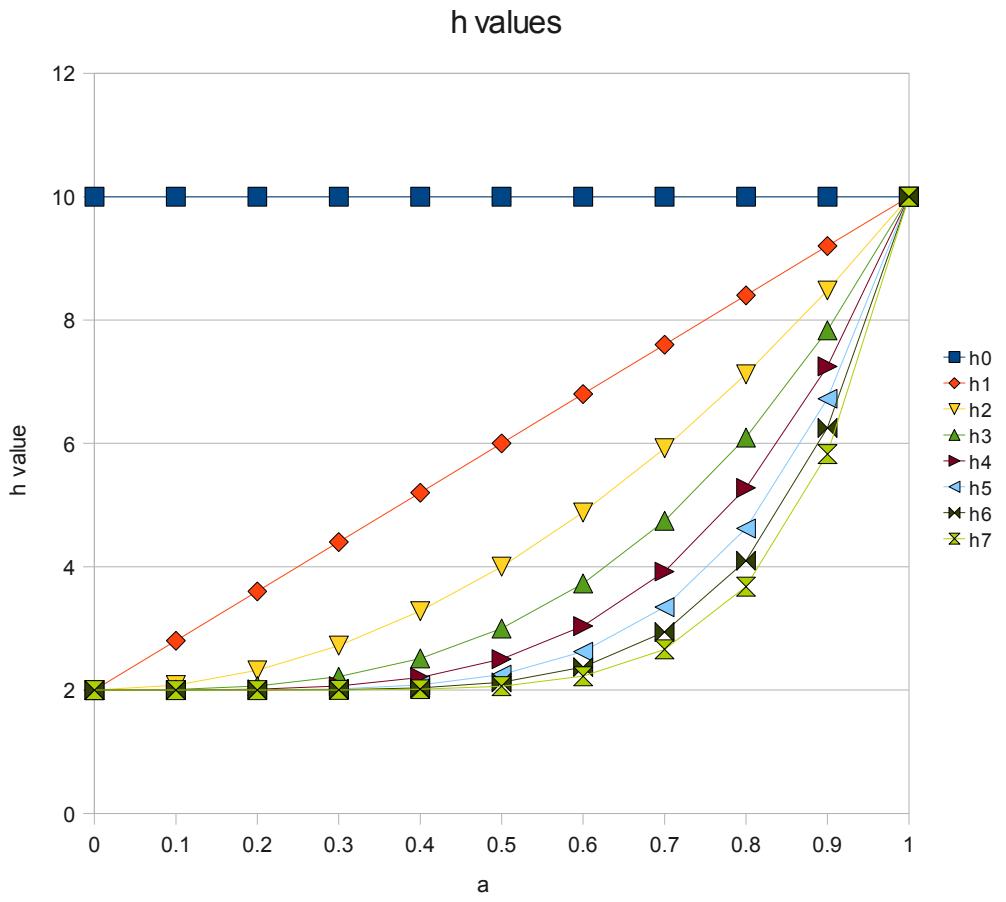
1: $\mathbf{X} \leftarrow$ ShuffleInstances($\mathbf{X}$)
2: $T \leftarrow$ CreateDataTransformation($\mathbf{X}$) ▷ Standardization or PCA
3: $\mathbf{X}' \leftarrow$ Transform($\mathbf{X}, T$)
4: $[h_{min}, h_{max}] \leftarrow$ CalculateHRange($\mathbf{X}'$) ▷ Based on kernel probability mass
5: $f_{cv} \leftarrow$ CreateCVErrorEstimator($k = 10, \mathbf{X}, [h_{min}, h_{max}], T$)
6: $a \leftarrow$ Minimize($f_{cv}, [0, 1]$)
7: **return** a

Obliczanie przedziału h

- Metoda opisana w [Ghosh06]: Badamy rozkład odległości w zbiorze danych
  - h_max = odległość między dwoma najbardziej oddalonymi punktami
  - h_min = 1/3 odległości między najbliższymi punktami, a jeśli to jest =0, to 1/3 pierwszego percentyla odległości między punktami
- Metoda stosowana w KDEC:
  - h_max = jw.
  - h_min = jw. tylko, że zamiast 1/3 stosuję wielkość zależną od przedziału, w którym znajduje się większość masy prawd. jądra

Optymalizacja błędu

$$h_i(a) = h_{min} + a^i(h_{max} - h_{min})$$



- Chcemy dobrać odpowiednią „szybkość” (parametr a) zmniejszania się jąder w sensownym przedziale szerokości jąder (h_min, h_max)

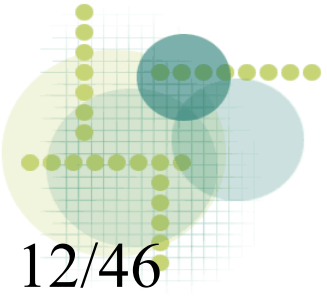
Optymalizacja błędu

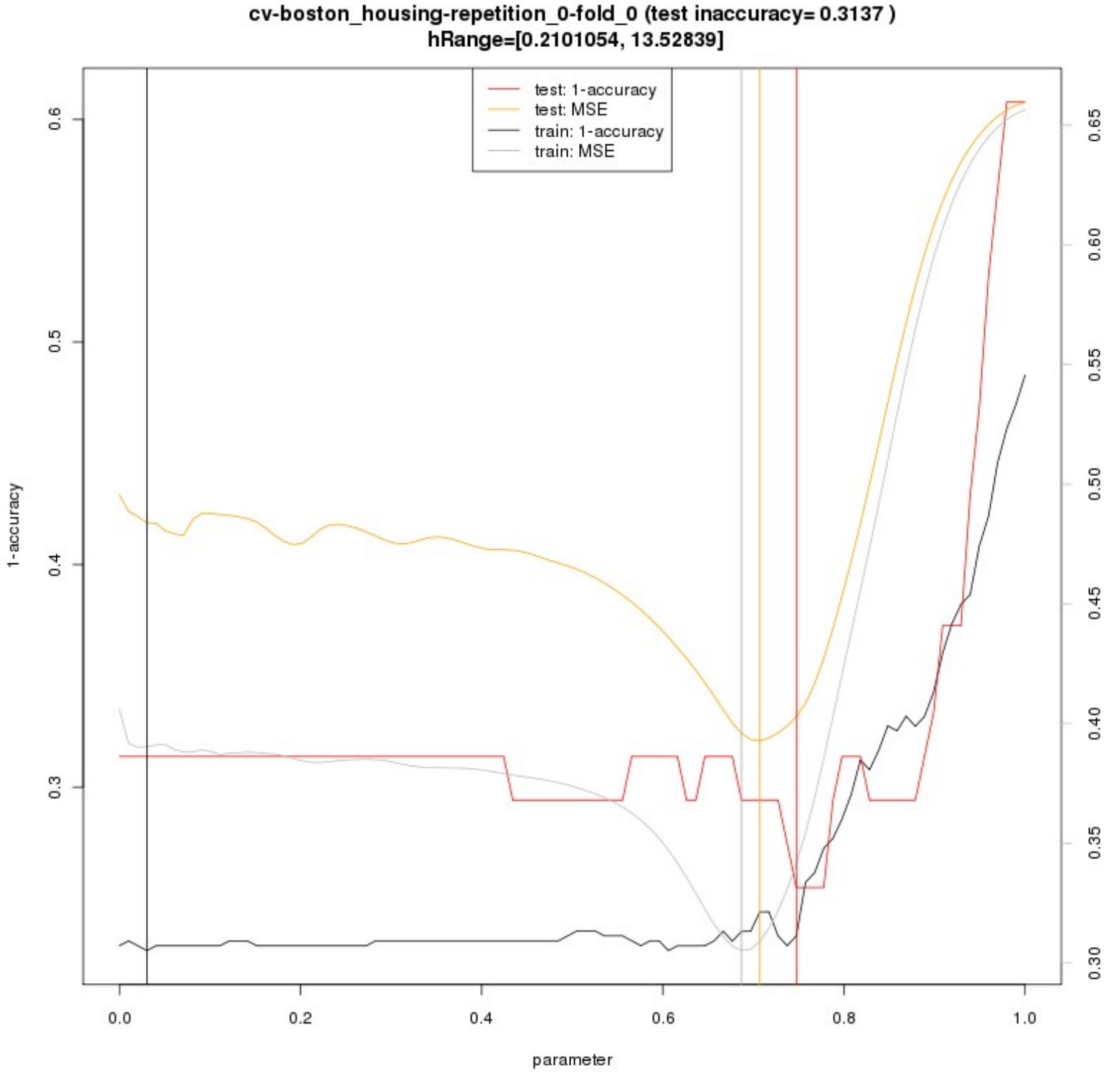
- Optymalizujemy błąd, który dla danego *a* obliczany za pomocą 10-fold cross-validation (używanie zbiorów walidujących jest konieczne ze względu na budowę algorytmu)
- Wzór dla jednego fold-a :

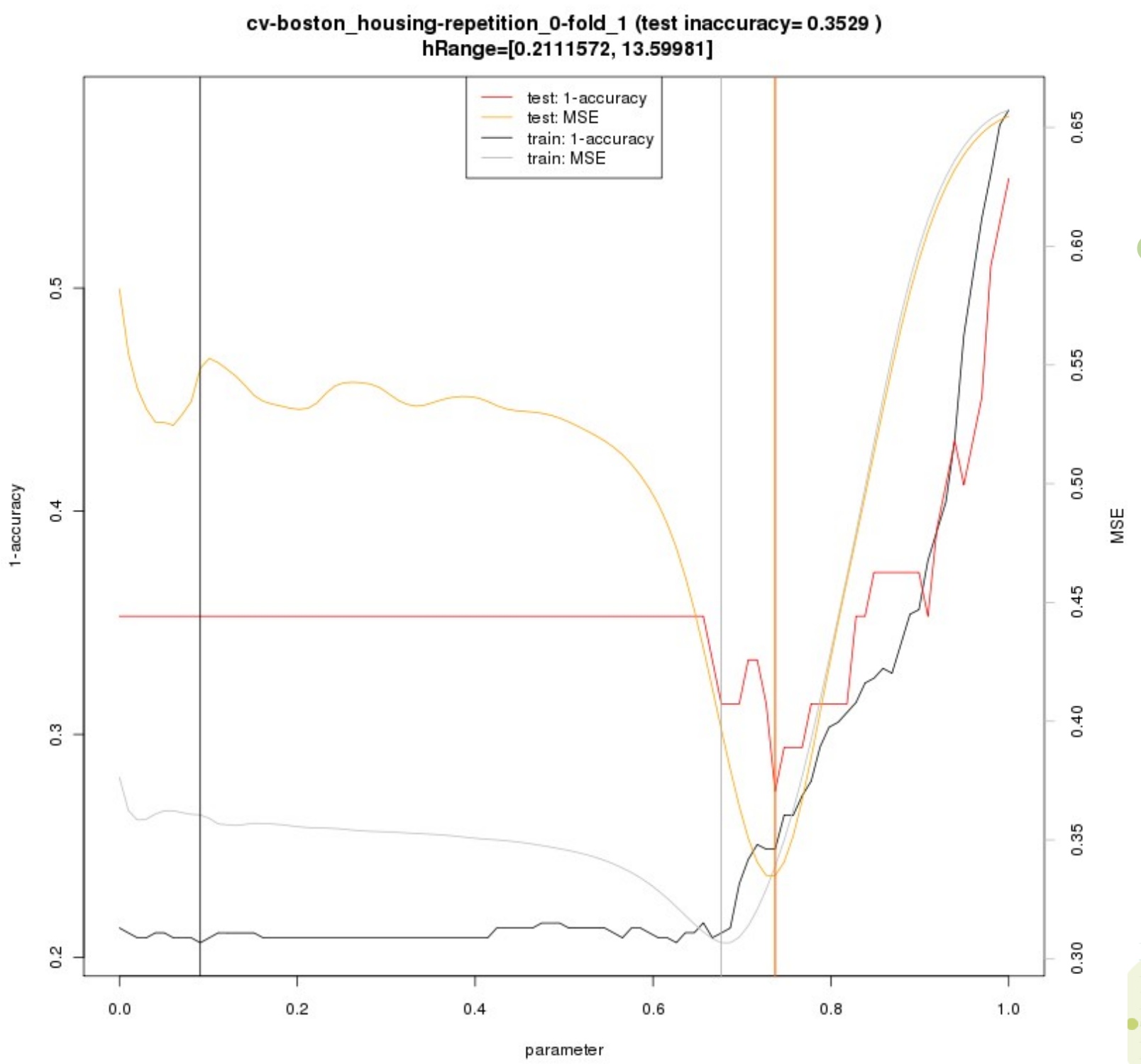
$$\text{Error}(a) = MSE(\hat{p}(\cdot; a), \mathcal{D}^v) = \frac{1}{|\mathcal{D}^v|} \sum_{\mathbf{x} \in \mathcal{D}^v} \sum_{i=1}^{c} (\hat{p}(\omega_i|\mathbf{x}; a) - \mathbf{t}(\mathbf{x})[i])^2$$

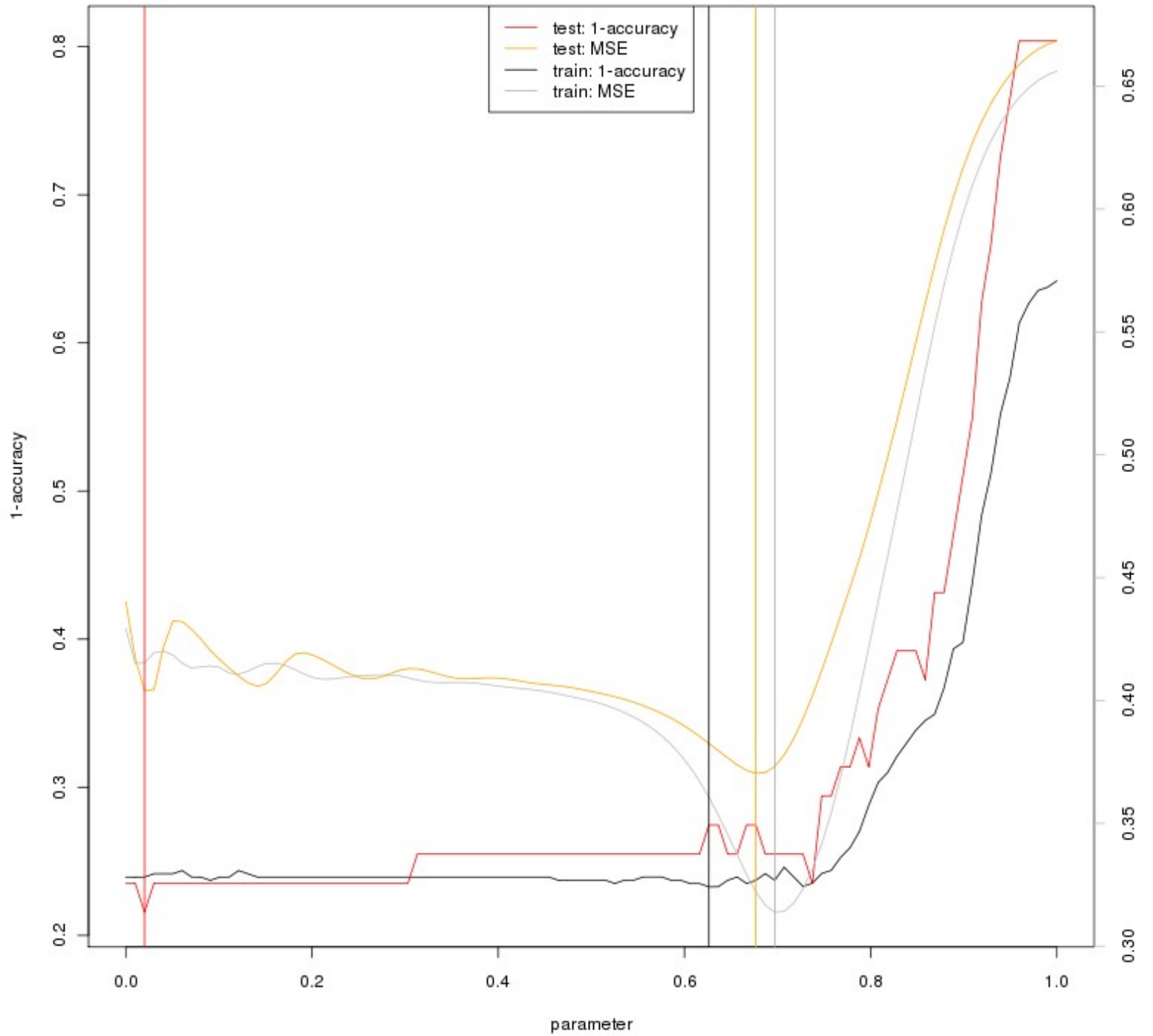
- a - estimator's parameter
- $\mathcal{D}^v$ - validation set
- c - number of classes
- $\hat{p}(\omega_i|\mathbf{x}; a)$ - estimation of class ω_i probability in point $\mathbf{x}$, where estimator's parameter is equal to a
- $\mathbf{t}(\mathbf{x})[i]$ - actual value of point $\mathbf{x}$ probability of class ω_i

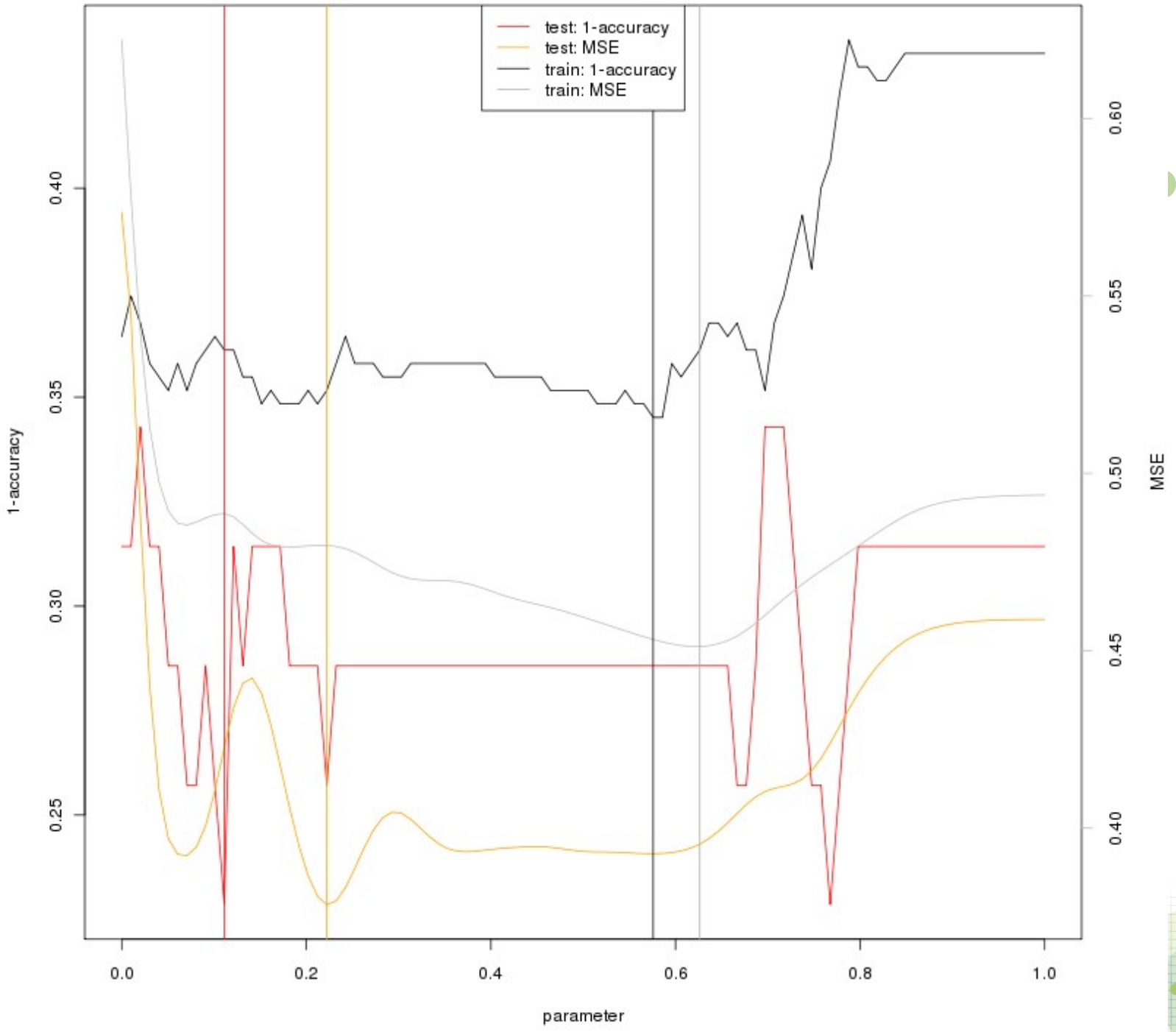
Błąd dla estymacji z 10 estymatorami dla różnych zbiorów danych

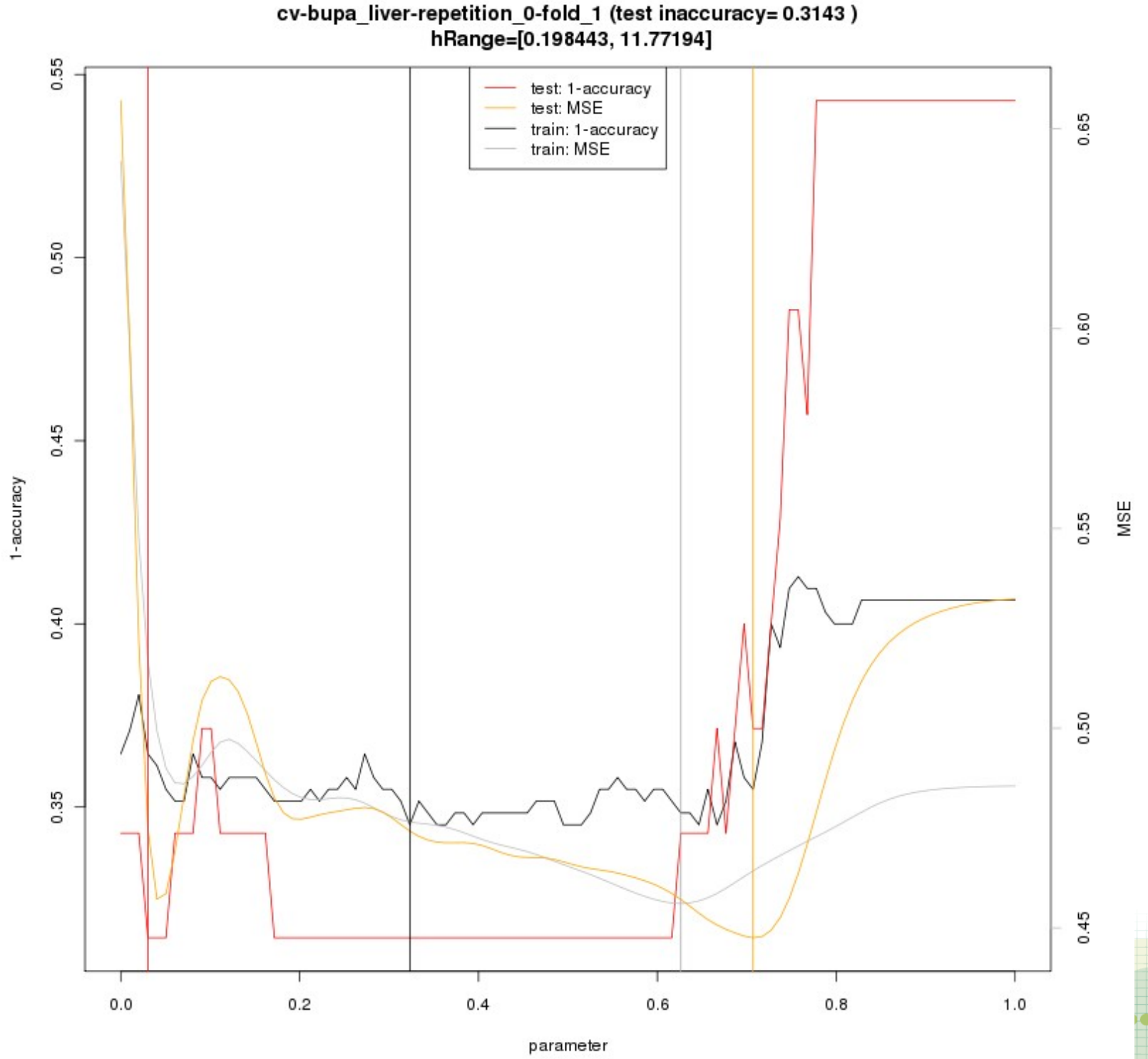


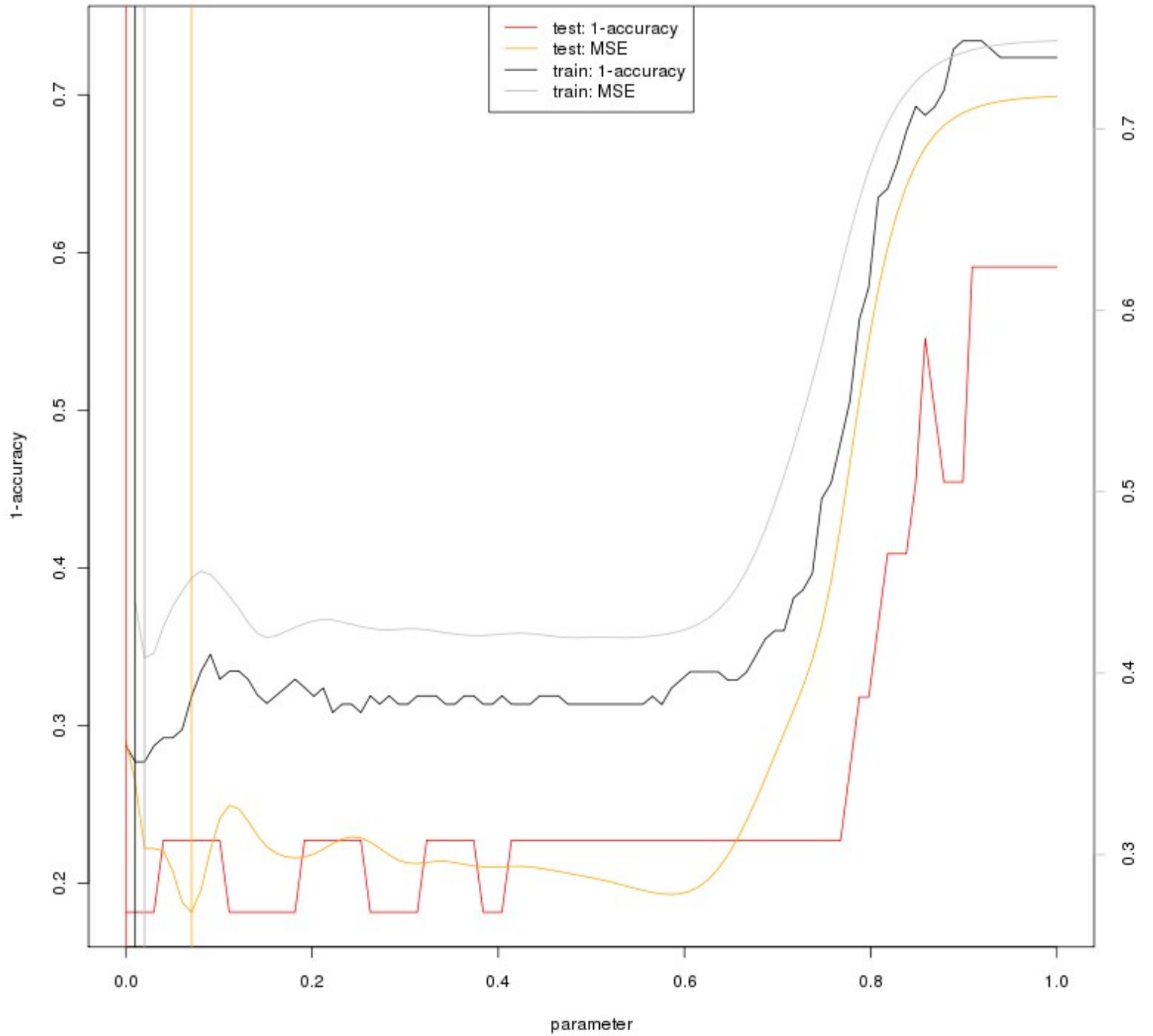

cv-boston_housing-repetition_0-fold_0 (test inaccuracy= 0.3137)
hRange=[0.2101054, 13.52839]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65

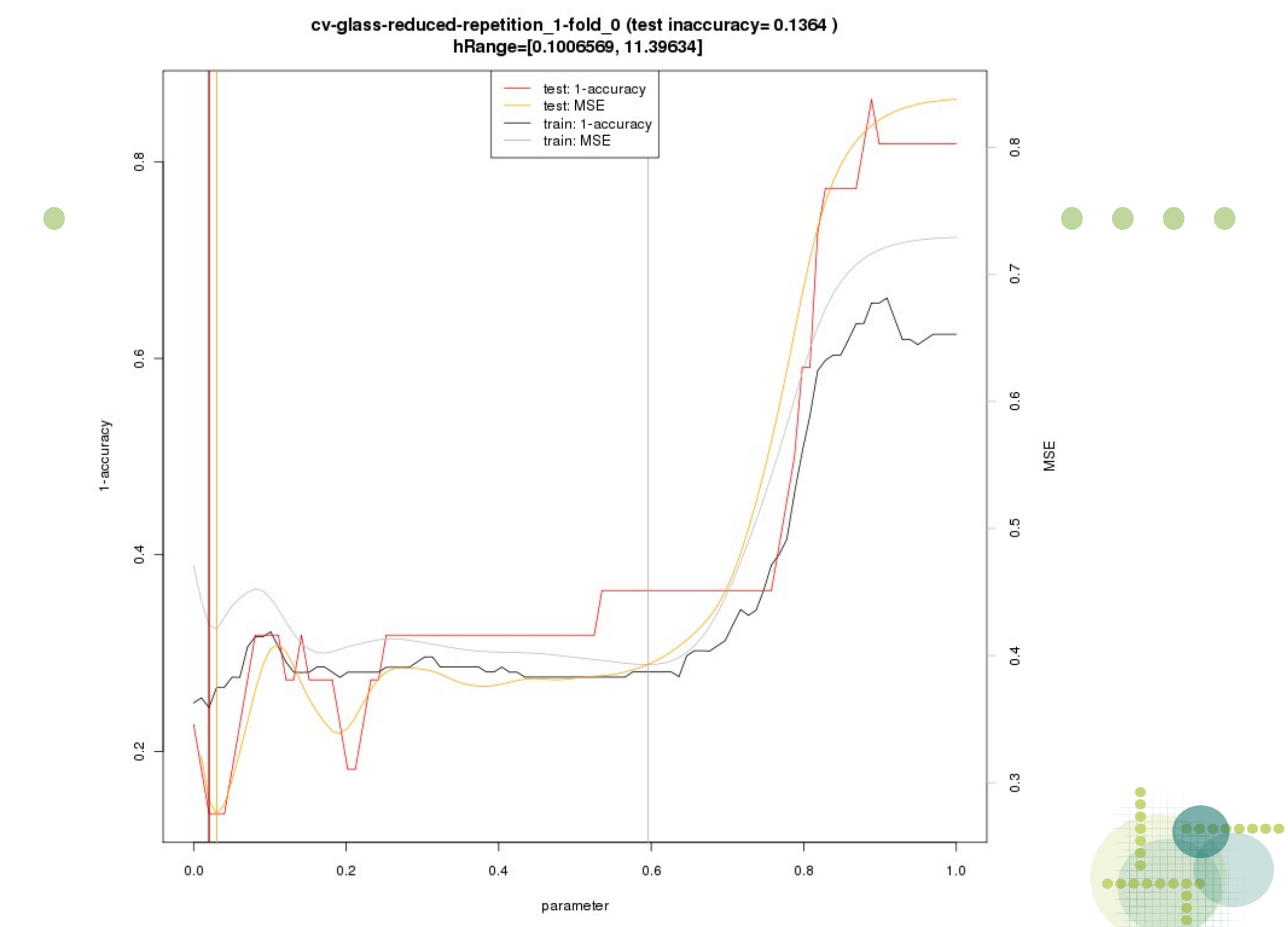

cv-boston_housing-repetition_0-fold_1 (test inaccuracy= 0.3529)
hRange=[0.2111572, 13.59981]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65

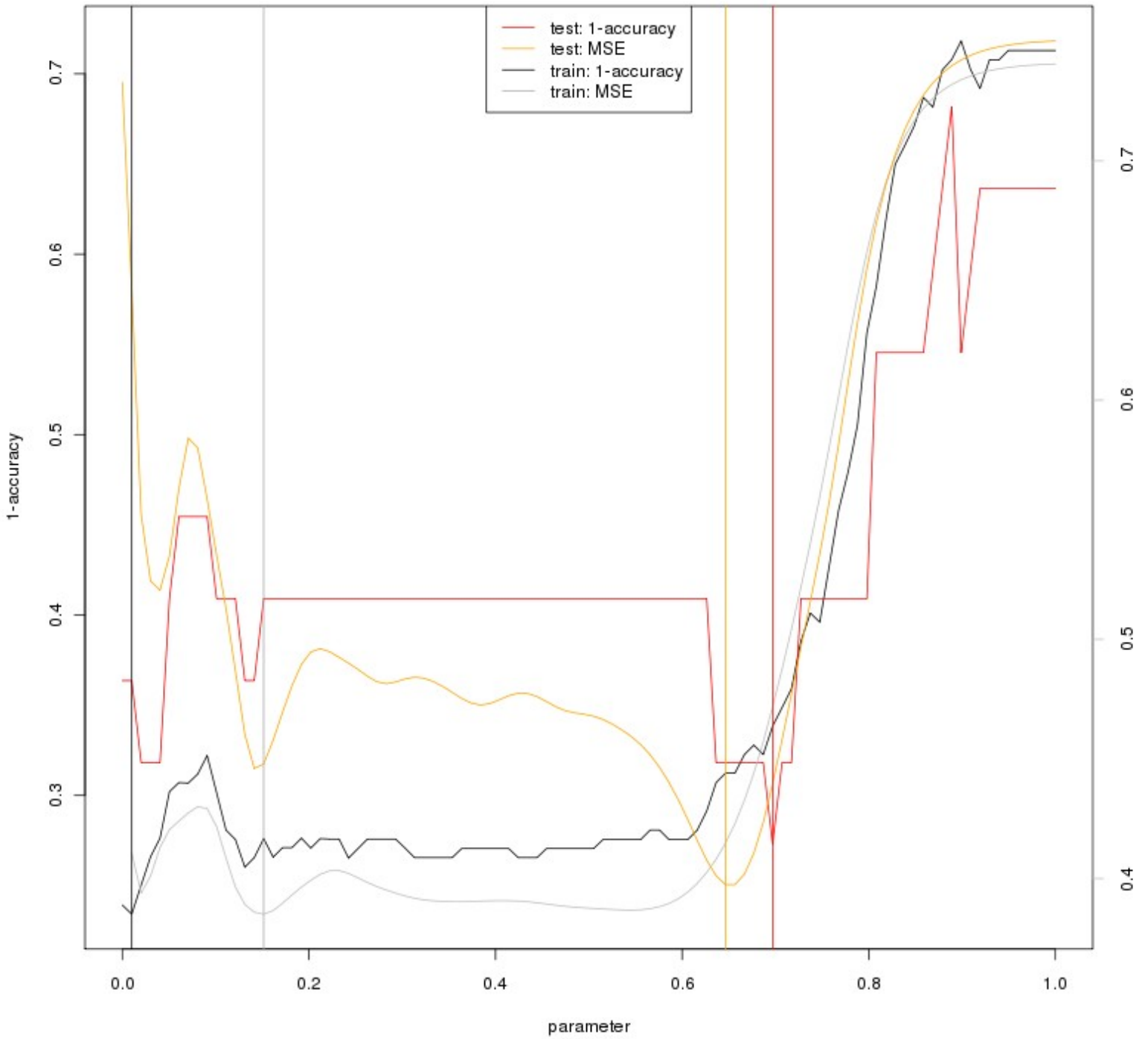

cv-boston_housing-repetition_0-fold_2 (test inaccuracy= 0.2745)
hRange=[0.2082502, 13.47565]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.0
0.2
0.4
0.6
0.8
1.0

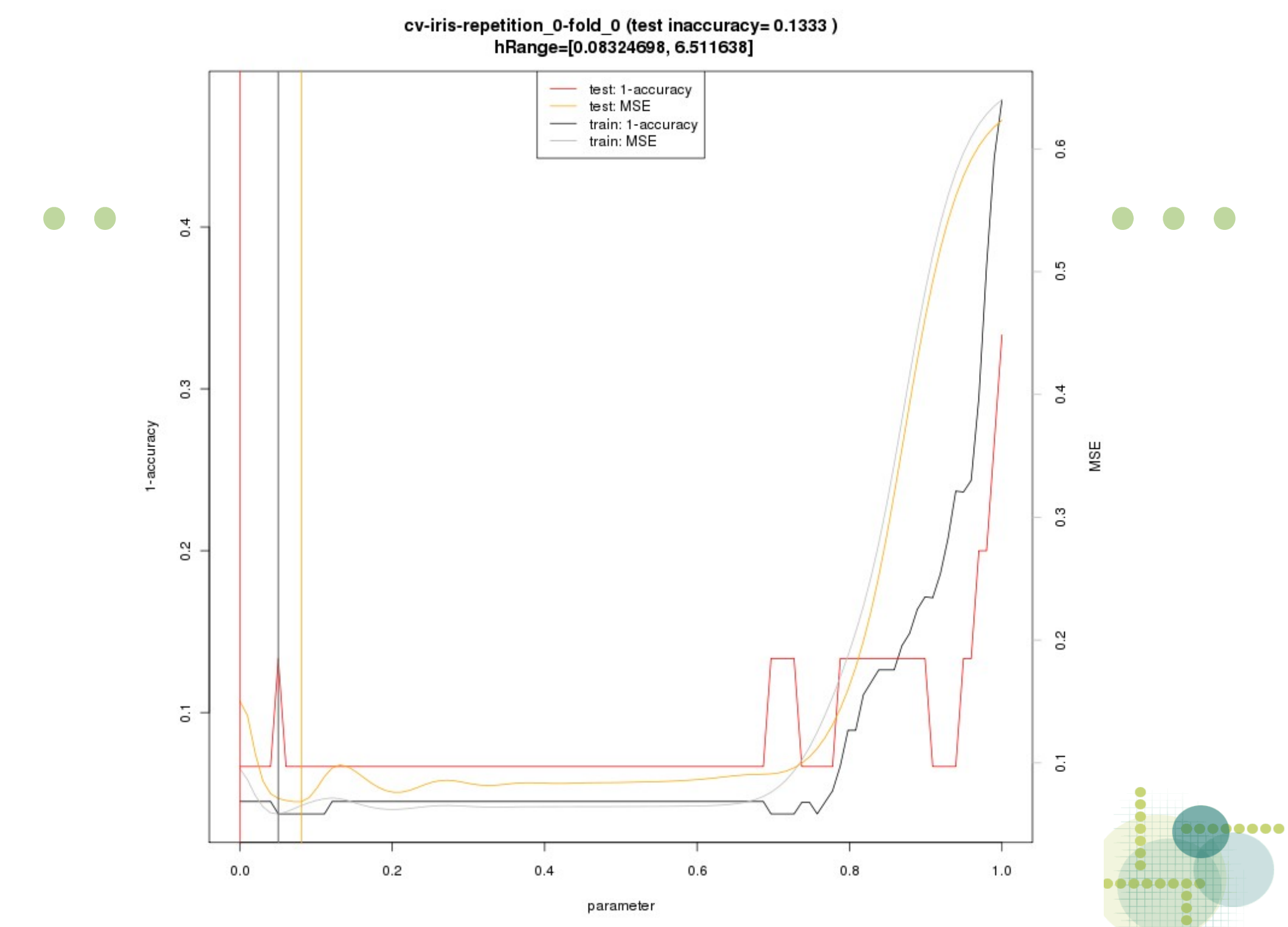

cv-bupa_liver-repetition_0-fold_0 (test inaccuracy= 0.2857)
hRange=[0.1959946, 11.01635]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.25
0.30
0.35
0.40
0.40
0.45
0.50
0.55
0.60
0.0
0.2
0.4
0.6
0.8
1.0

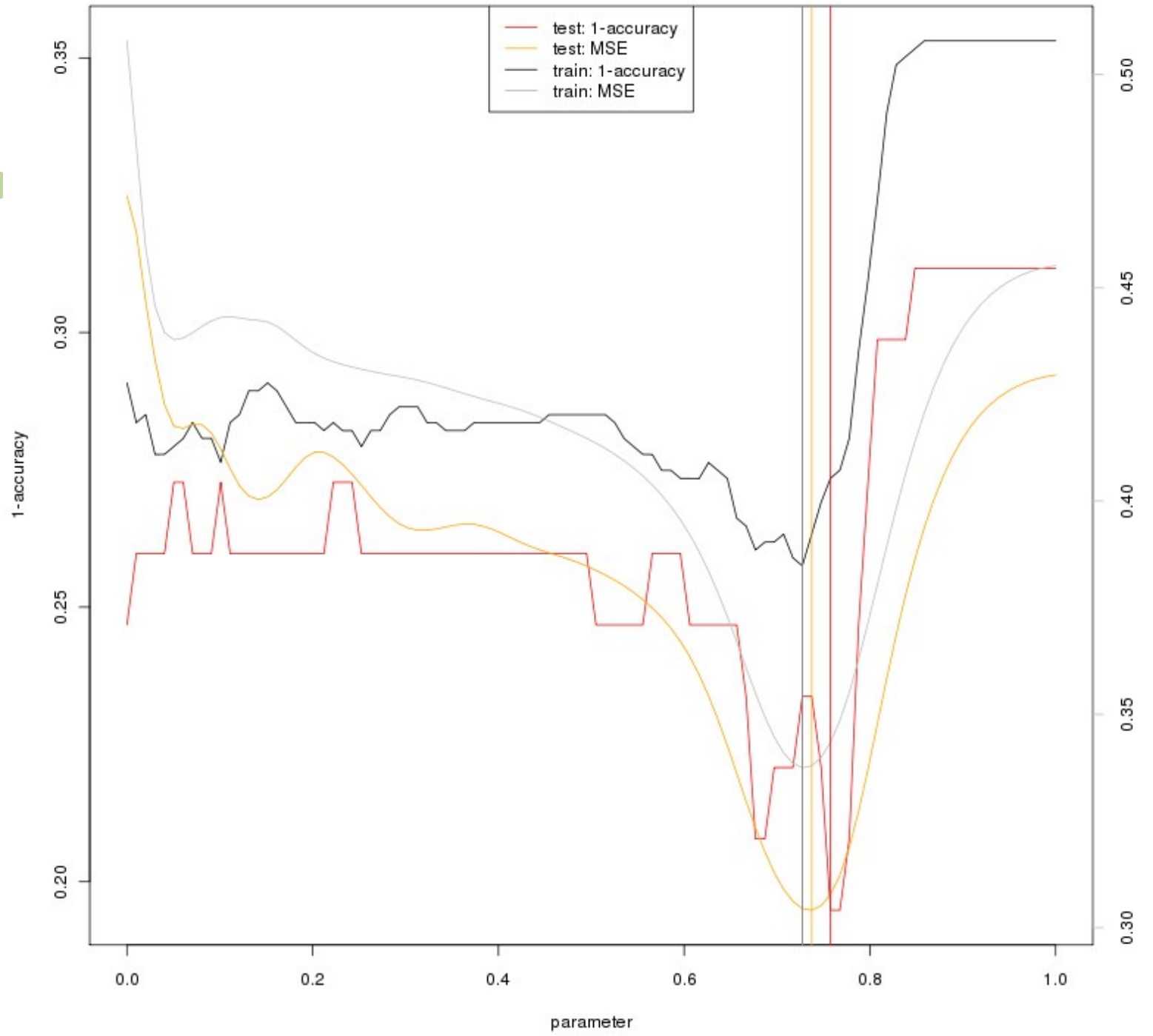

cv-bupa_liver-repetition_0-fold_1 (test inaccuracy= 0.3143)
hRange=[0.198443, 11.77194]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.0
0.2
0.4
0.6
0.8
1.0


cv-glass-reduced-repetition_0-fold_0 (test inaccuracy= 0.1818)
hRange=[0.1014668, 14.67544]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7

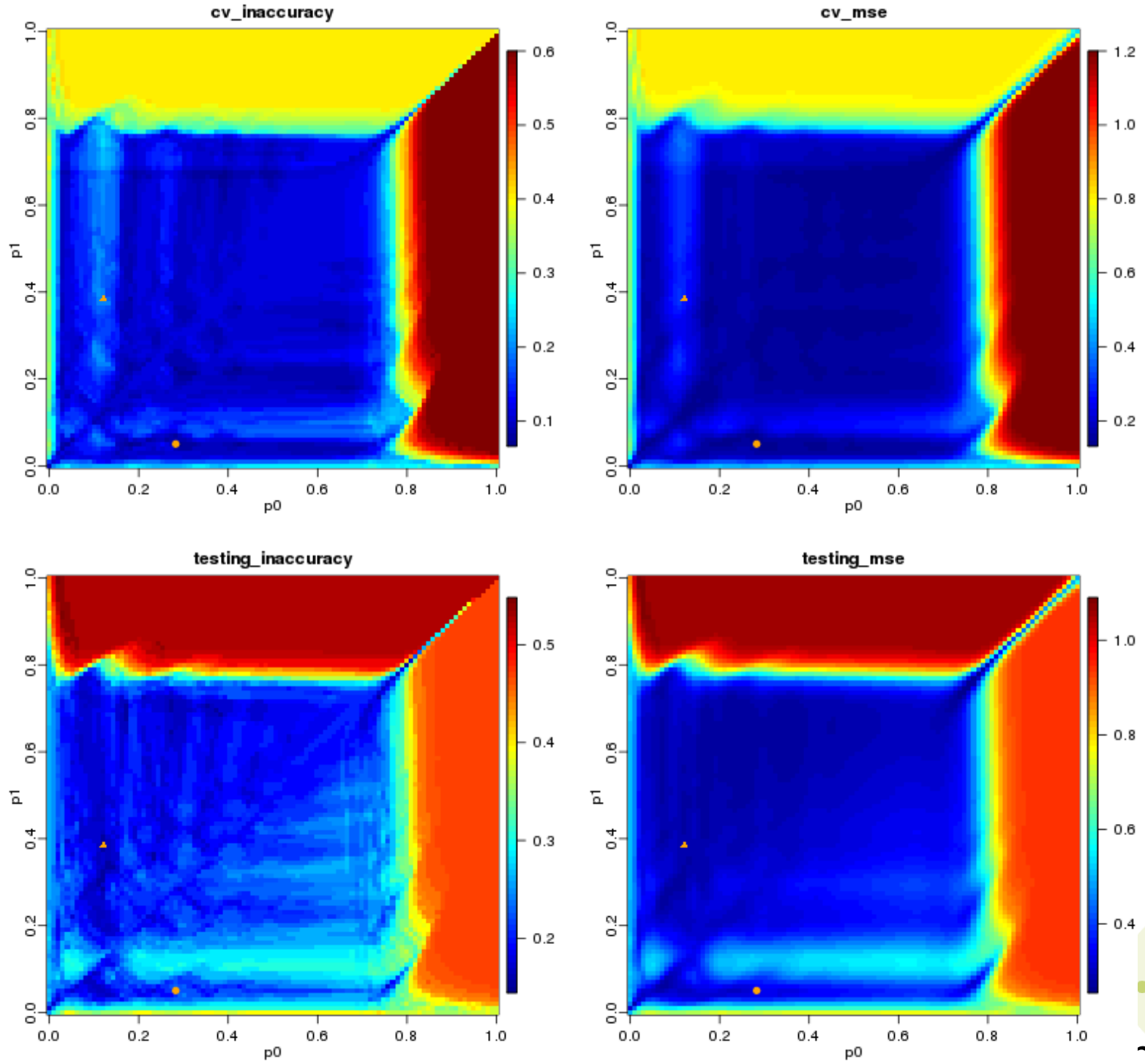

cv-glass-reduced-repetition_1-fold_0 (test inaccuracy= 0.1364)
hRange=[0.1006569, 11.39634]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8

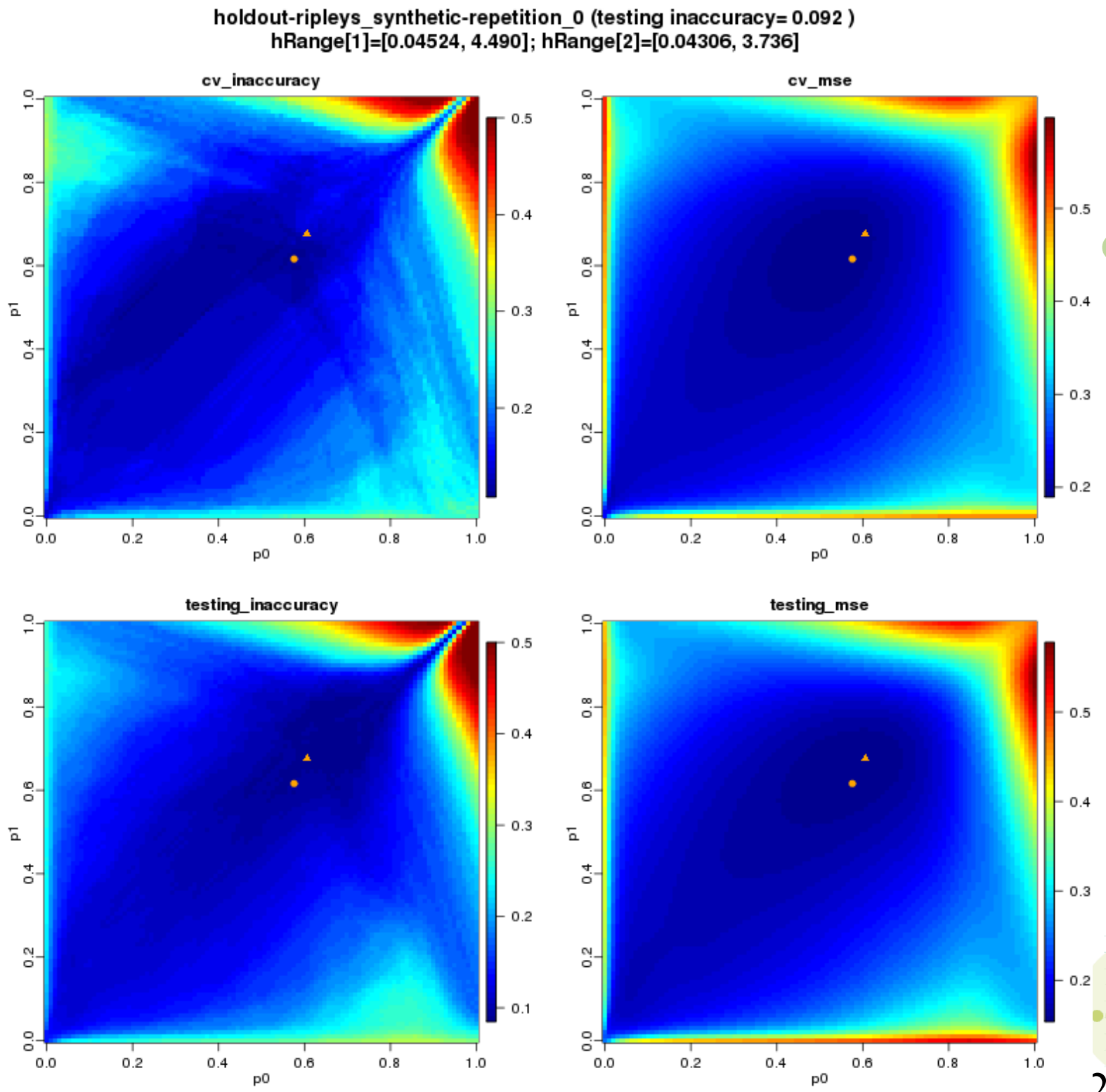

cv-glass-reduced-repetition_1-fold_1 (test inaccuracy= 0.3636)
hRange=[0.09324603, 14.19717]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.7

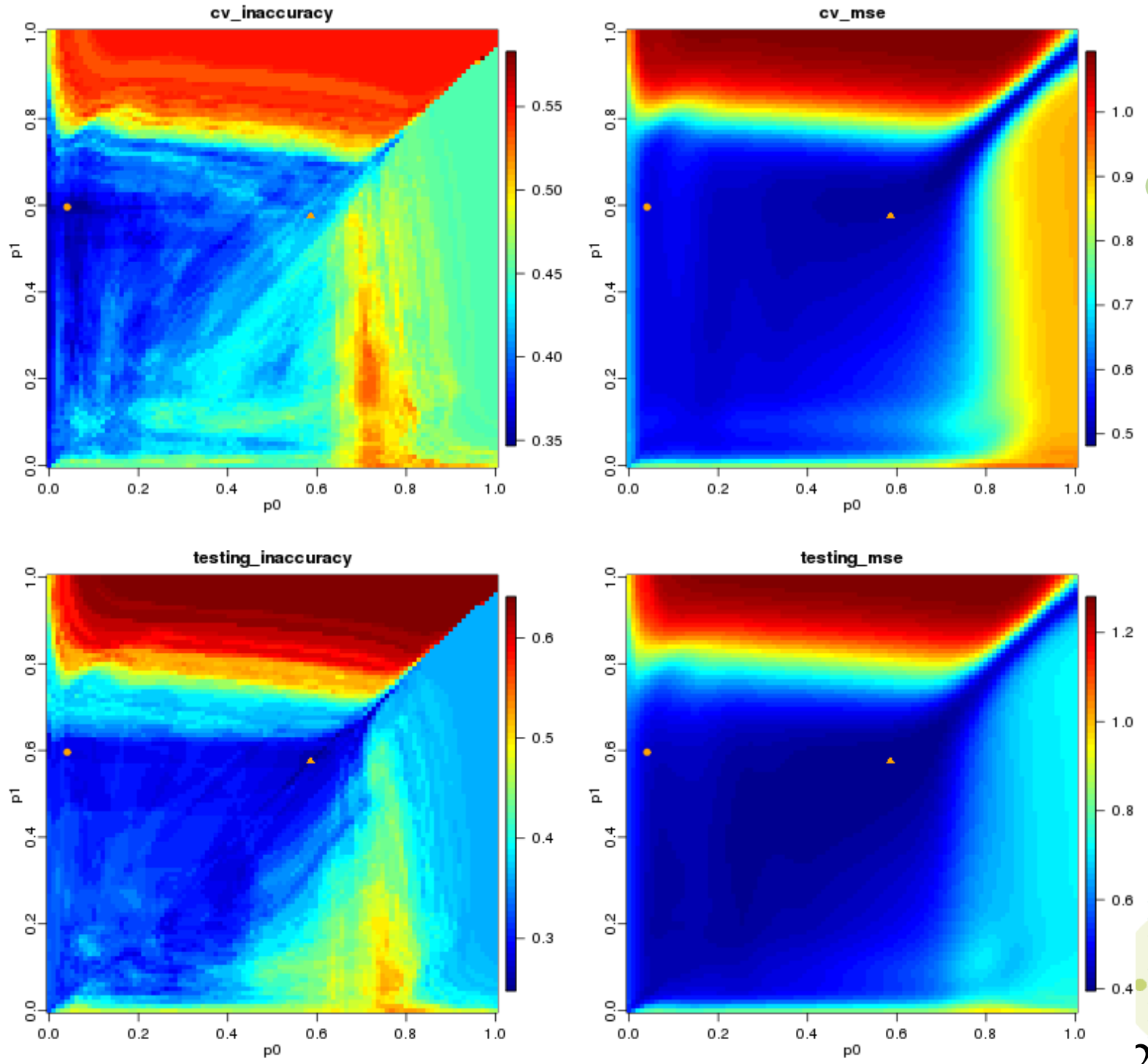

cv-iris-repetition_0-fold_0 (test inaccuracy= 0.1333)
hRange=[0.08324698, 6.511638]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.2
0.3
0.4
0.5
0.6

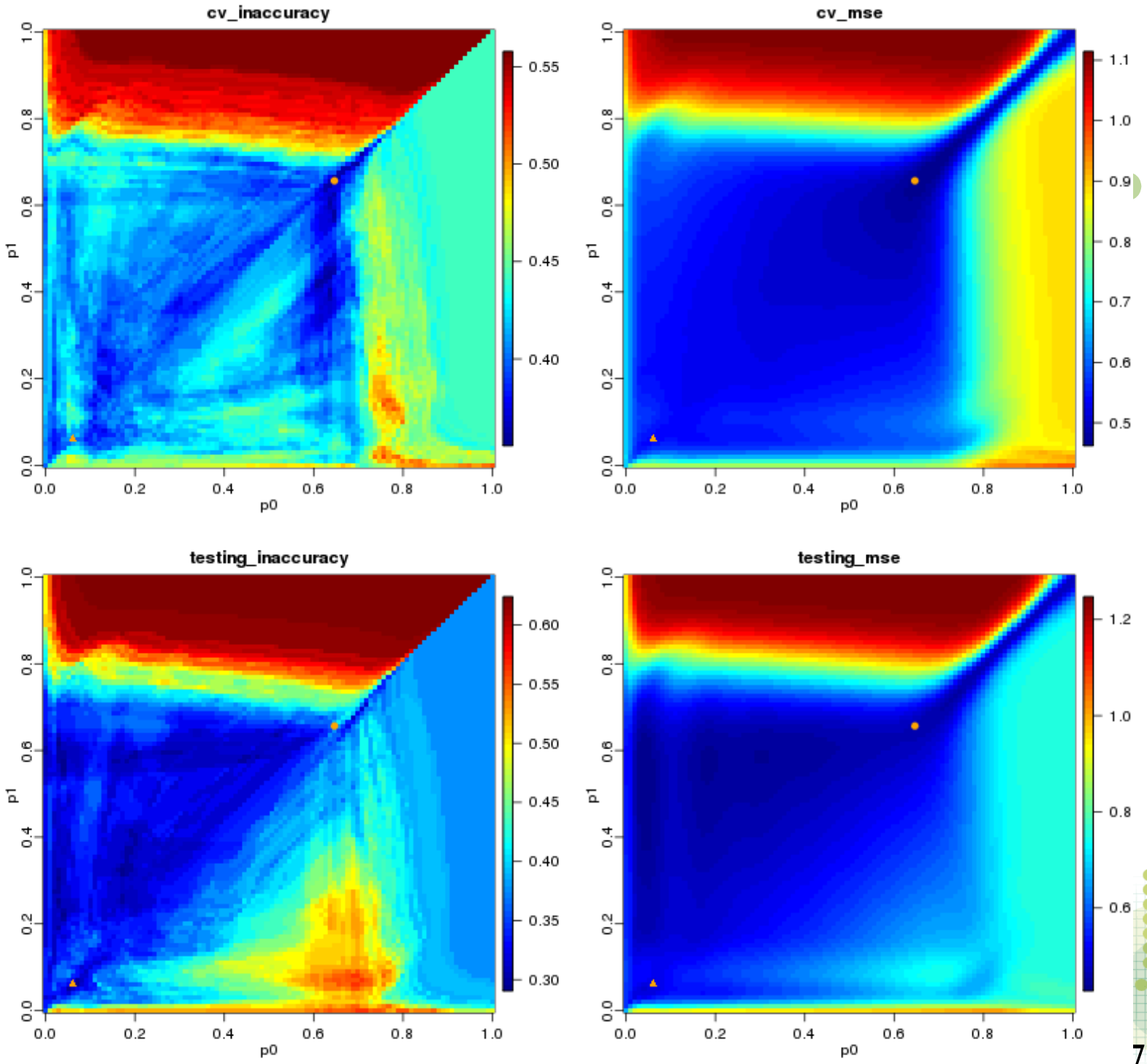

cv-pima_indians_diabetes-repetition_0-fold_0 (test inaccuracy= 0.2338)
hRange=[0.2803036, 11.93288]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.20
0.25
0.30
0.35
0.30
0.35
0.40
0.45
0.50
0.0
0.2
0.4
0.6
0.8
1.0

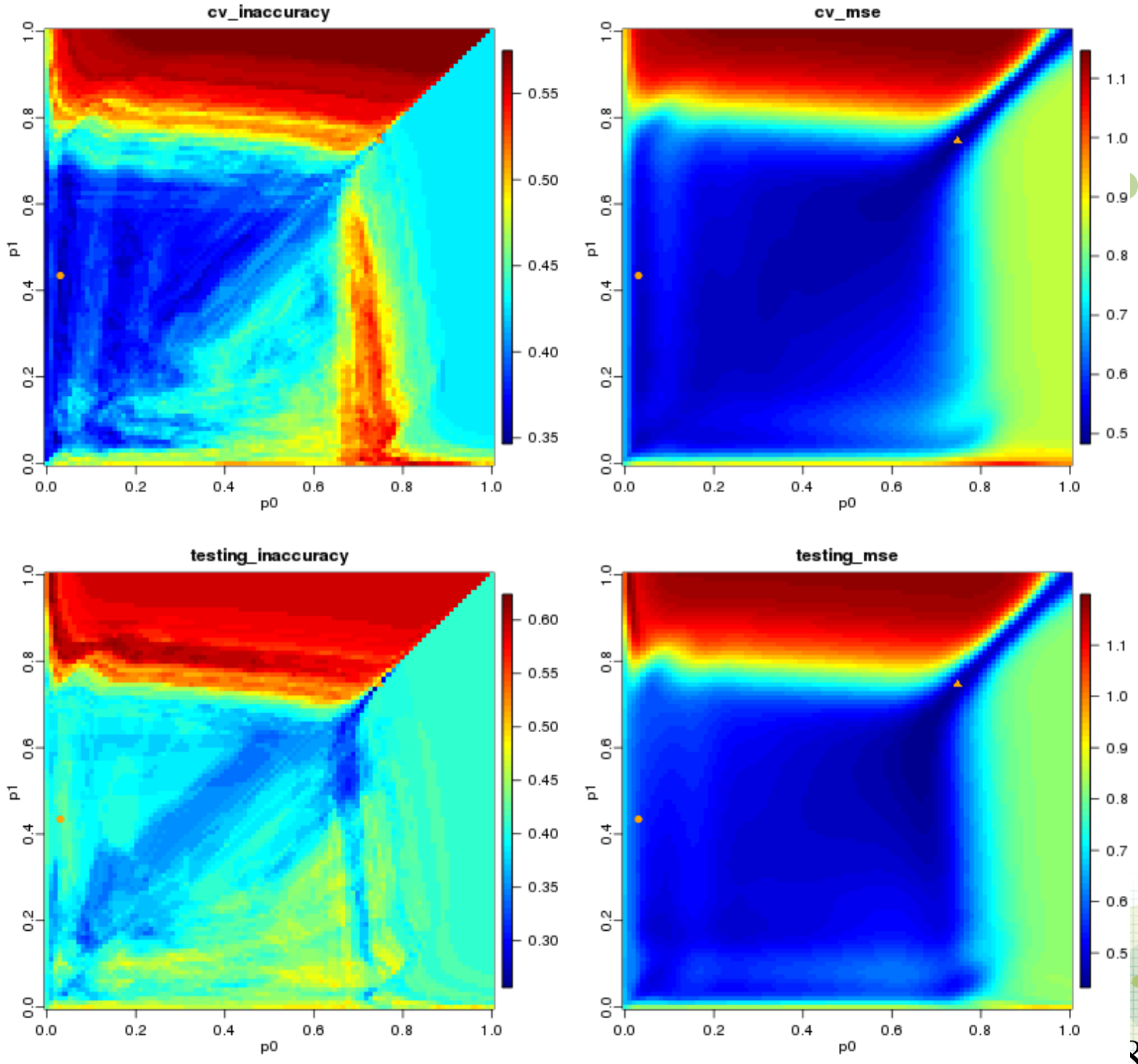
- Błąd estymacji z 10 estymatorami oddzielne a dla każdej z klas (z przedziałami h dopasowywanymi dla każdej z klas oddzielnie oraz wspólnie)


holdout-sonar-repetition_0 (testing inaccuracy= 0.2212)
hRange[1]=[0.3255, 11.98]; hRange[2]=[0.2947, 11.85]
cv_inaccuracy
cv_mse
testing_inaccuracy
testing_mse
p0
p1


holdout-ripleys_synthetic-repetition_0 (testing inaccuracy= 0.092)
hRange[1]=[0.04524, 4.490]; hRange[2]=[0.04306, 3.736]
cv_inaccuracy
cv_mse
testing_inaccuracy
testing_mse
p0
p1
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.2
0.3
0.4
0.5

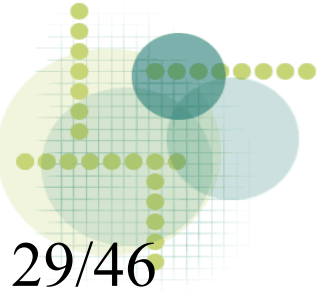

holdout-bupa_liver-repetition_0 (testing inaccuracy= 0.2906)
hRange[1]=[0.1811, 8.485]; hRange[2]=[0.2193, 11.04]
cv_inaccuracy
cv_mse
testing_inaccuracy
testing_mse
p0
p1


holdout-bupa_liver-repetition_0 (testing inaccuracy= 0.3761)
hRange[1]=[0.1815, 11.43]; hRange[2]=[0.1815, 11.43]
cv_inaccuracy
cv_mse
testing_inaccuracy
testing_mse
p0
p1


holdout-bupa_liver-repetition_1 (testing inaccuracy= 0.4103)
hRange[1]=[0.1740, 11.51]; hRange[2]=[0.1740, 11.51]
cv_inaccuracy
cv_mse
testing_inaccuracy
testing_mse
p0
p1

Zalety i wady algorytmu

- Zalety
  - Spojrzenie na dane z różnych odległości
  - Klasyfikacja z wieloma klasami
  - Zwraca bezpośrednio prawdopodobieństwa przynależności punktu testowego do każdej z klas
- Wady
  - Powolna klasyfikacja - wymaga iteracji po wszystkich elementach ze zbioru uczącego
  - Nie jest tworzony żaden model danych
  - Działa tylko na zbiorach danych z atrybutami numerycznymi



Testy

- Zbiory danych:
  - Źródło: z literatury w wersji używanej w jednym z 2 artykułów ([Lim00], [Ghosh06])
- Sposób testowania:
  - Holdout: zbiór ze zdefiniowanym zbiorem testowym: 10 powtórzeń i uśrednienie
  - Cv: Bez zdefiniowanego zbioru testowego: 10 x 10-fold cross-validation
- Wersja algorytmu:
  - Transformacja: standardyzacja
  - Co optymalizujemy: parametr a (taki sam dla każdej klasy)

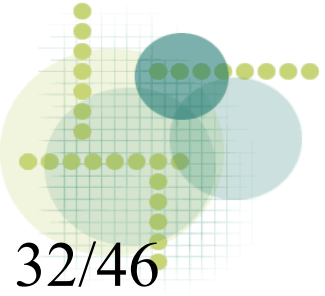
Testy - wyniki

Dataset name	test type	source	classes	attributes	instances
Boston housing	cv	[Lim00]	3	13	506
BUPA liver disorders	cv	[Lim00]	2	6	345
Glass	cv		6	9	214
Glass – reduced	cv	[Ghosh06]	6	5	214
iris	cv		3	4	150
PIMA indians diabetes	cv	[Lim00]	2	7	532
StatLog vehicle silhouettes	cv	[Lim00]	4	18	846
ripleys synthetic	holdout	[Ghosh06]	2	2	1250
sonar averaged	holdout	[Ghosh06]	2	20	208
StatLog satellite image	holdout	[Lim00]	6	36	6435

	10 estimators				1 estimator		
Dataset name	MSE opt error	quantile	inaccuracy opt error	quantile	MSE opt error	inaccuracy opt error	
Boston housing	0.2352	0.122	0.2441	0.3211	0.2328	0.2414	
BUPA liver disorders	0.3573	0.761	0.3637	0.7713	0.4105	0.3803	
Glass	0.3146		0.3124		0.3197	0.3217	
Glass – reduced	0.2859	0.036	0.2515		0.2608	0.2604	lit. Min:0.285
iris	0.0513		0.0520		0.0540	0.0573	
PIMA indians diabetes	0.2582	0.907	0.2588	0.9094	0.2529	0.2562	
StatLog vehicle silhouettes	0.2881	0.622	0.2938	0.6415	0.2882	0.2933	
ripleys synthetic	0.0967	0.643	0.1026	0.8875	0.1003	0.1165	B. min: 0.08
sonar averaged	0.2106	0.931	0.1952	0.6649	0.1981	0.1962	
StatLog satellite image					0.0965	0.0965	lit. Min:0.098

Testy - spostrzeżenia

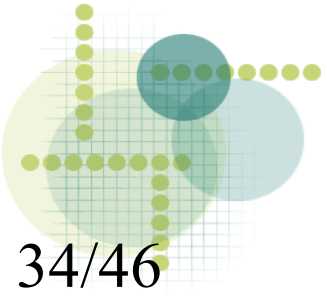
- Generalnie wyniki są poniżej średniej w porównaniu z innymi klasyfikatorami (kolor czerwony)
- Dla 2 zbiorów udało się uzyskać wyniki (trochę) lepsze niż te literaturowe
- MSE jest dobrym przybliżeniem błędu klasyfikacji
- Wyniki dla liczby jąder =10 są podobne do dla jednego jądra ale trochę lepsze (chociaż nieznacznie)
  - a nauka dla 10 jąder trwa 10-krotnie dłużej!

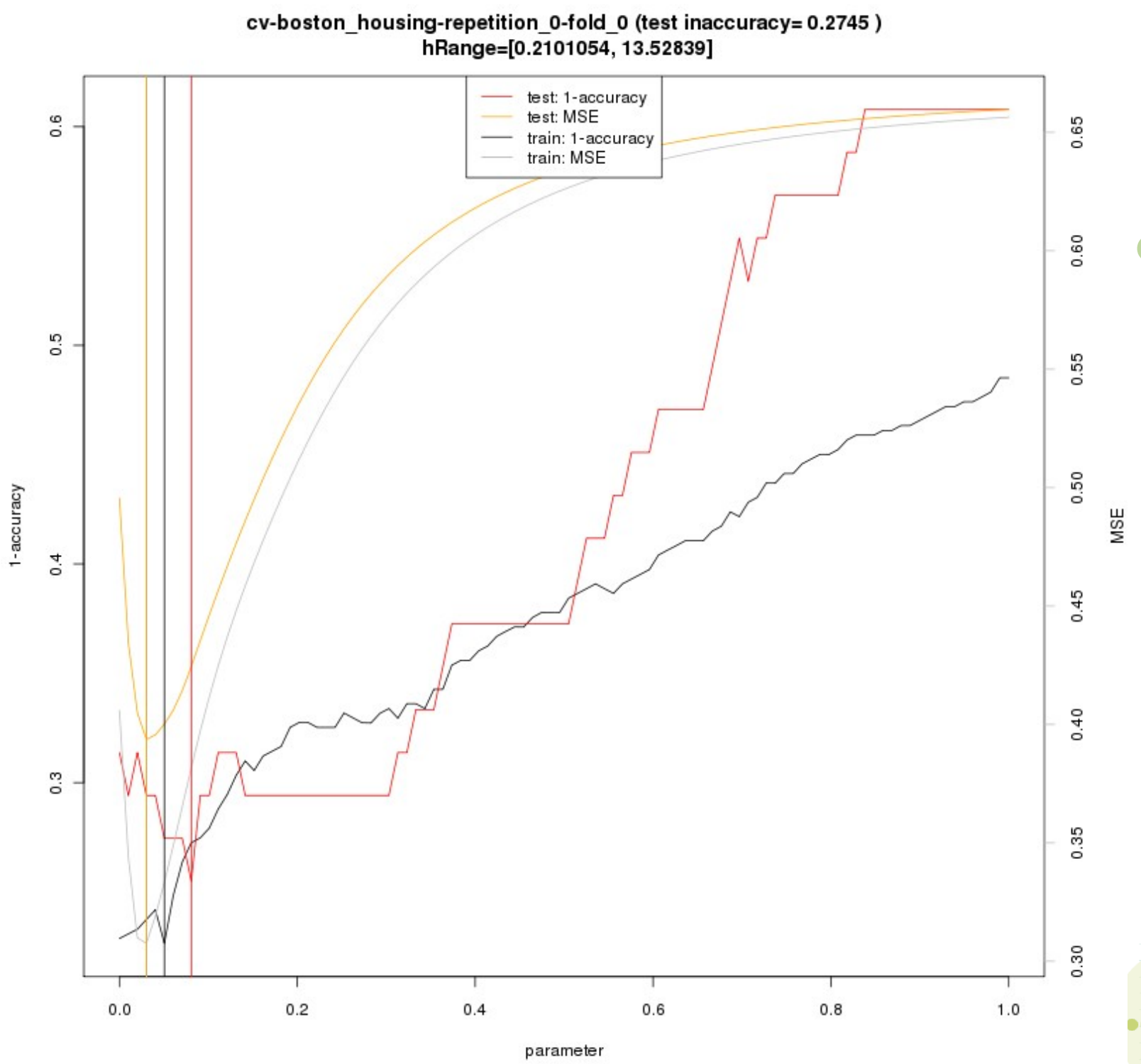


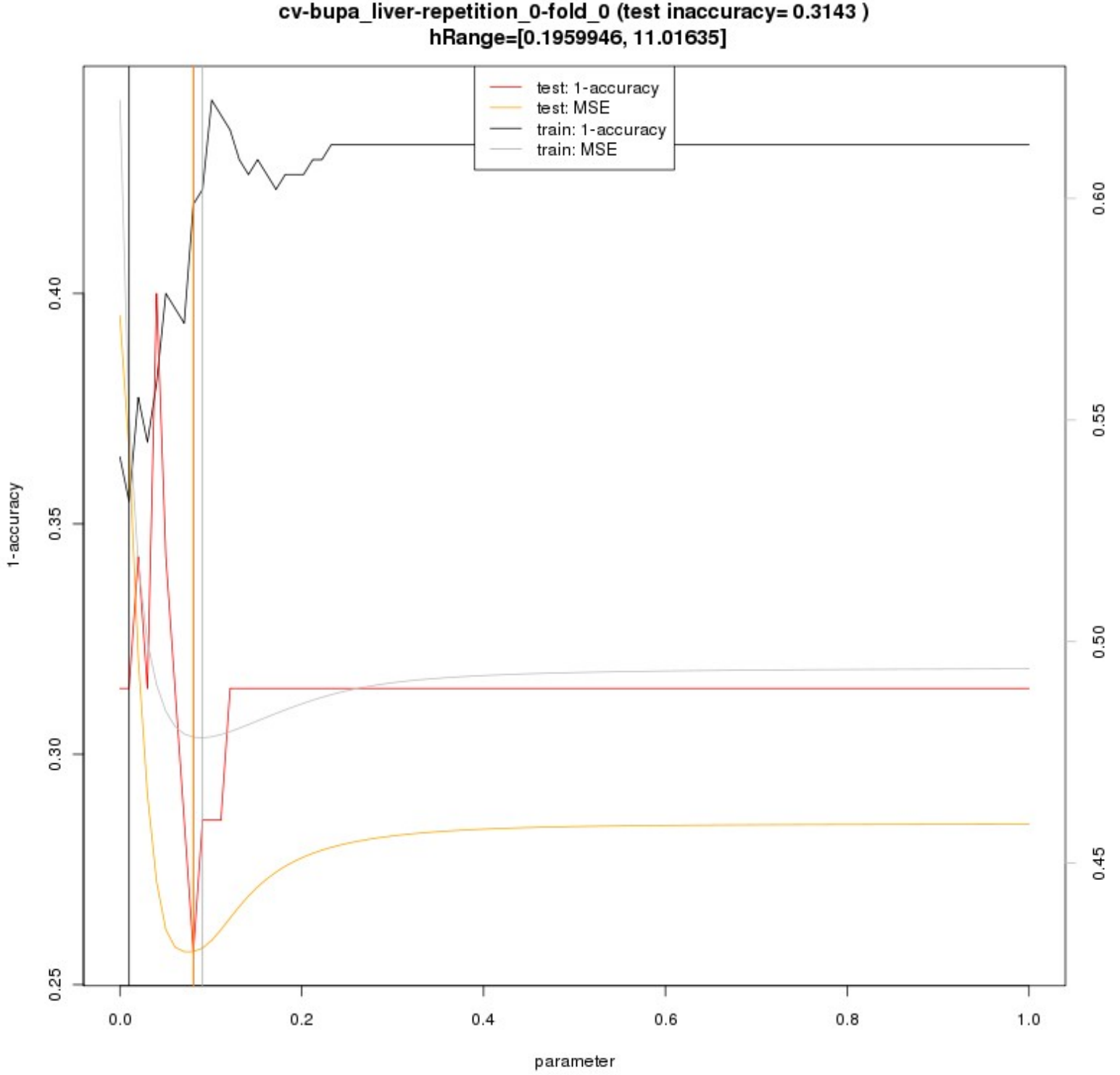
Testy - uwagi

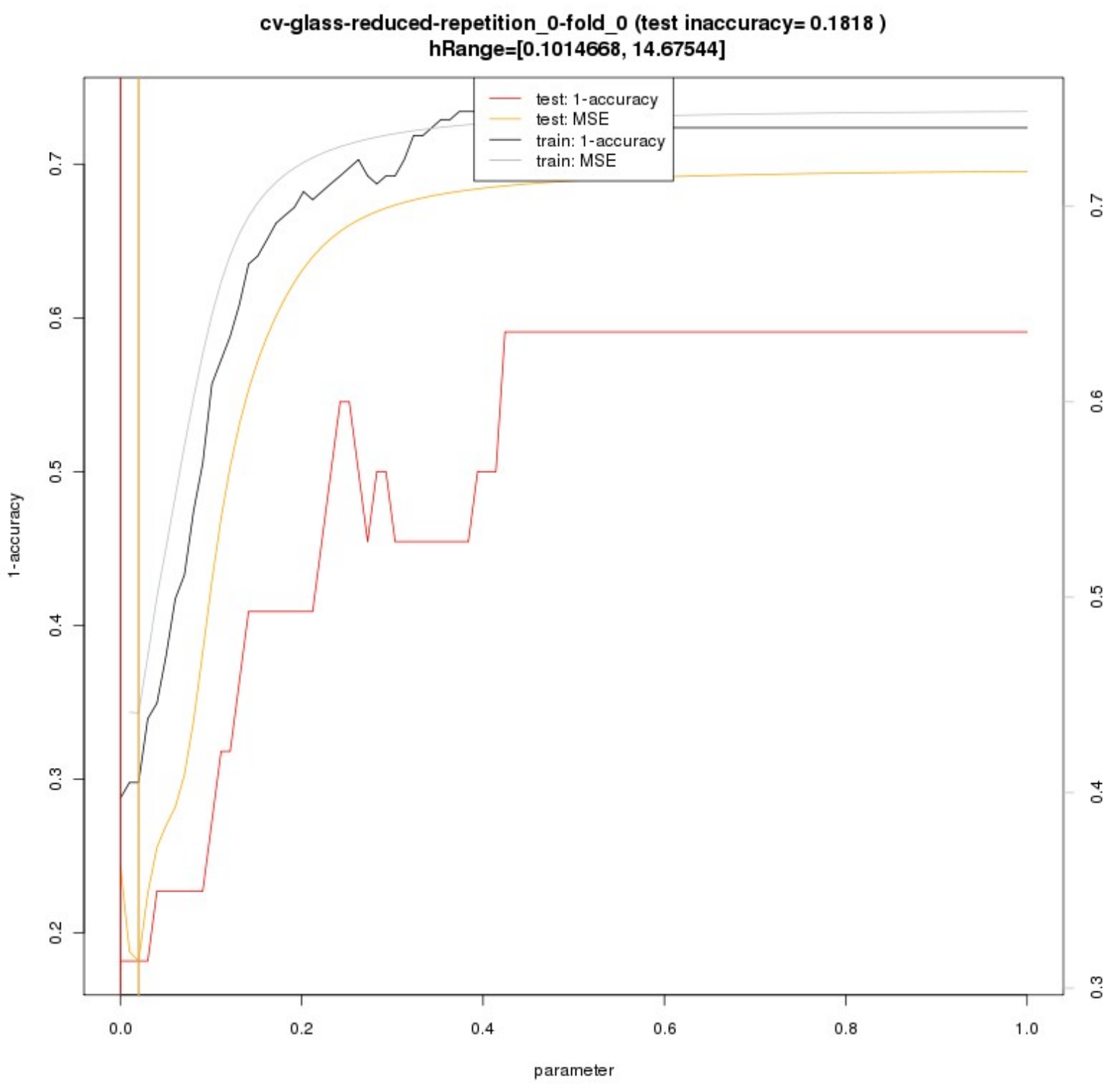
- Dziwne wyniki dla zbioru StatLog satellite image (dokładnie taki sam błąd inaccuracy za każdym razem (różnice w błędzie MSE też bliskie 0))
- Zła postać funkcji błędu dla estymacji z 1 estymatorem:
  - Minimum zazwyczaj na brzegu przedziału
  - Na dużej części przedziału funkcja jest płaska

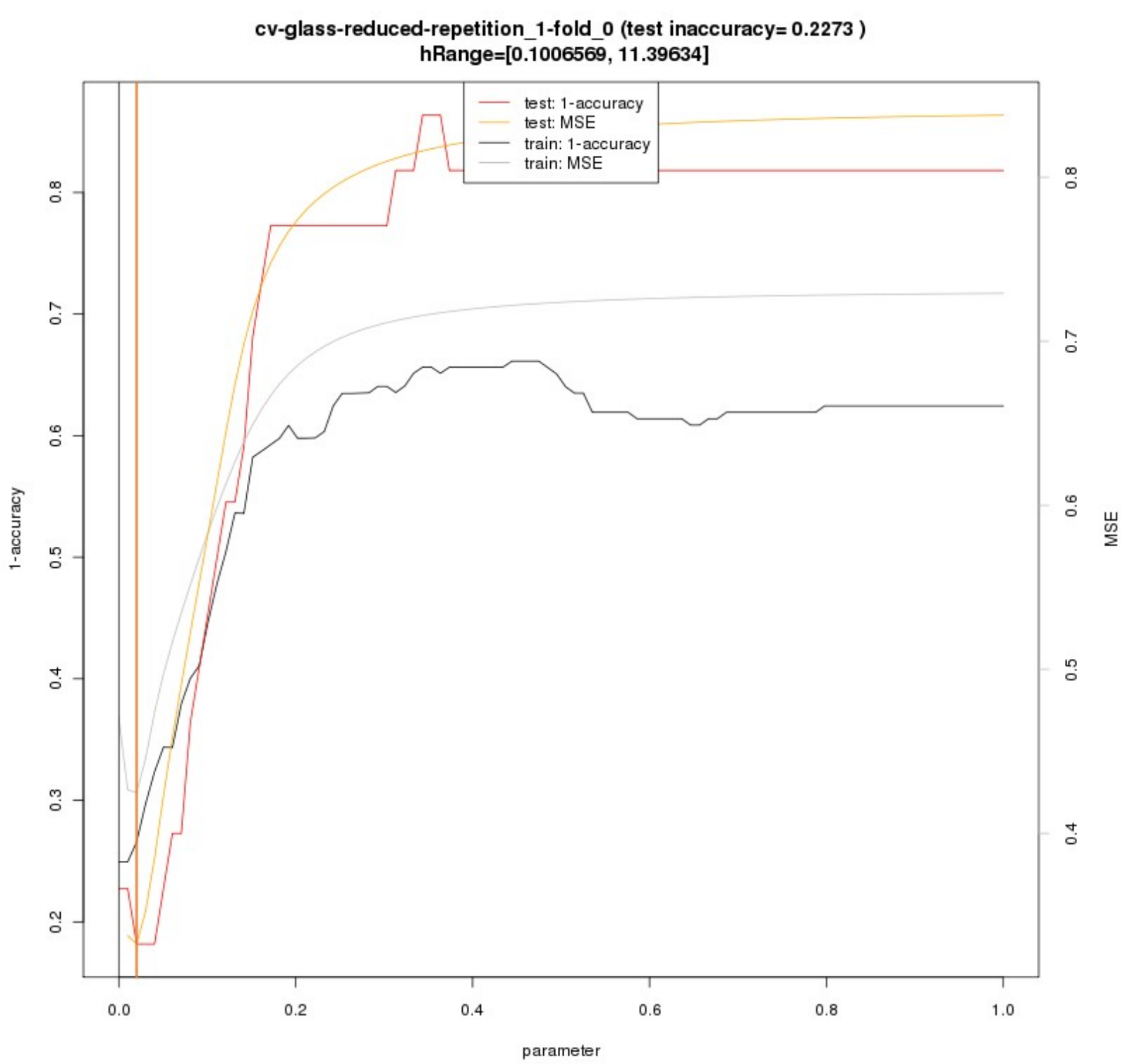
Błąd dla estymacji z 1 estymatorem dla różnych zbiorów danych

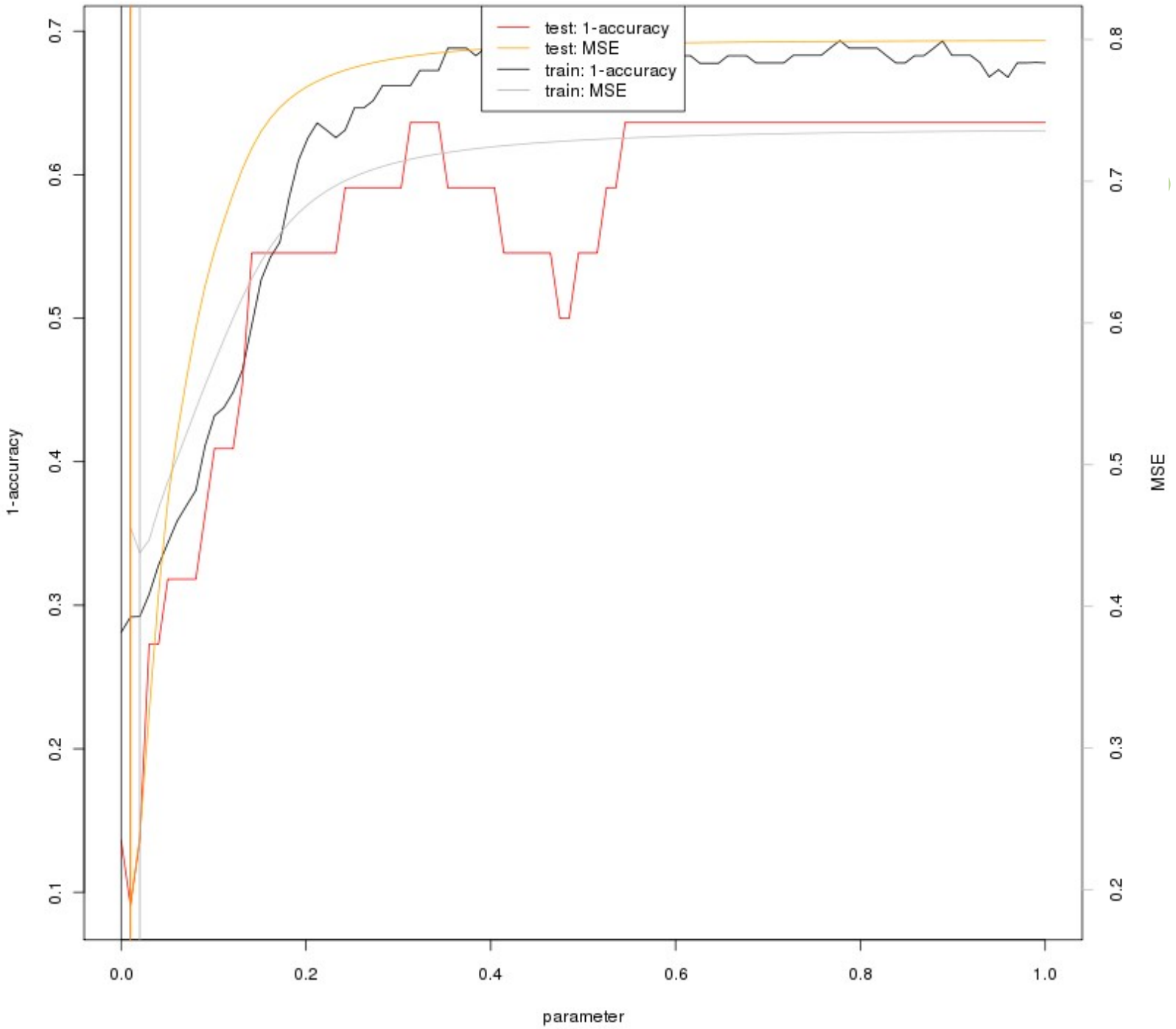


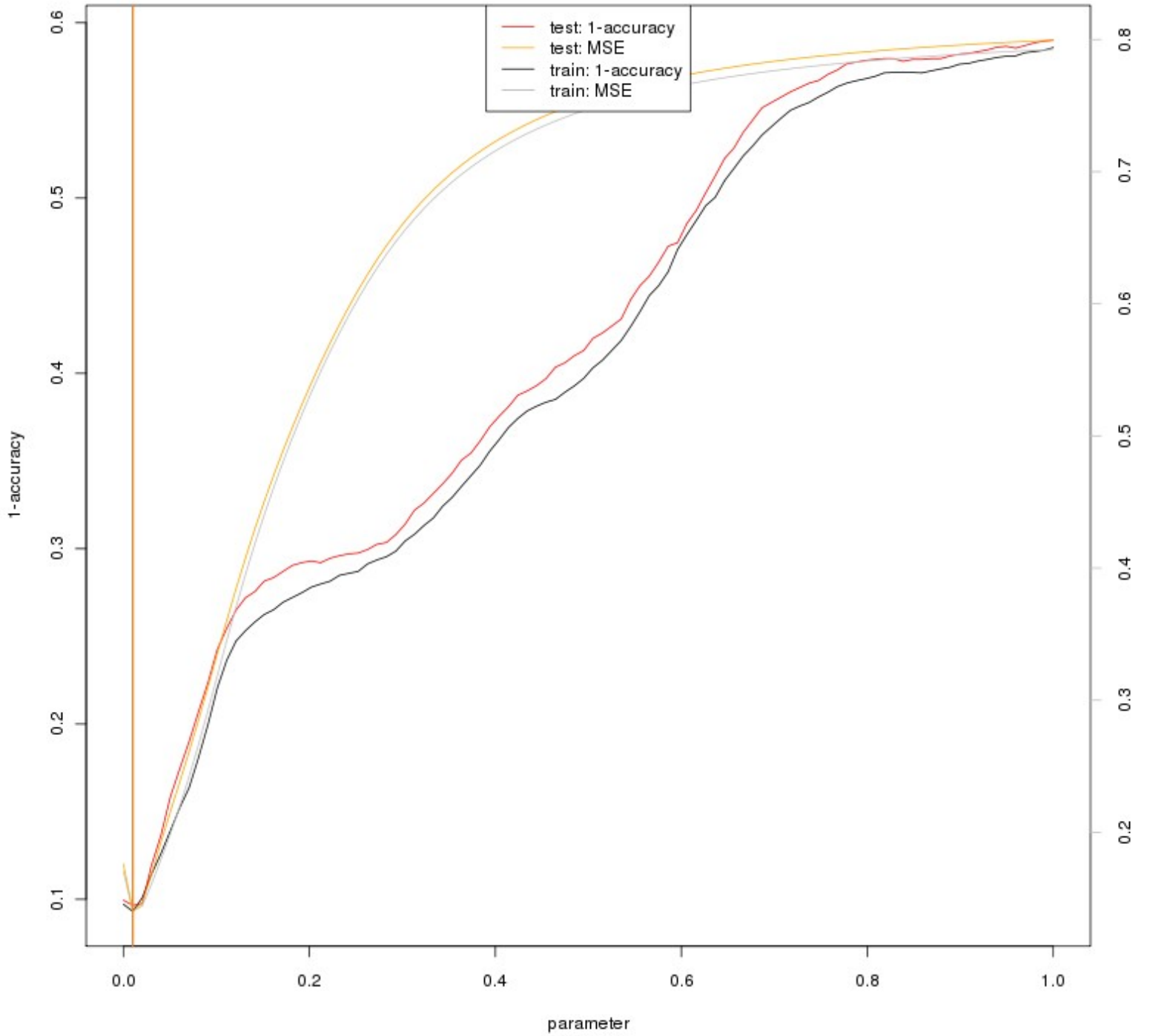

cv-boston_housing-repetition_0-fold_0 (test inaccuracy= 0.2745)
hRange=[0.2101054, 13.52839]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65

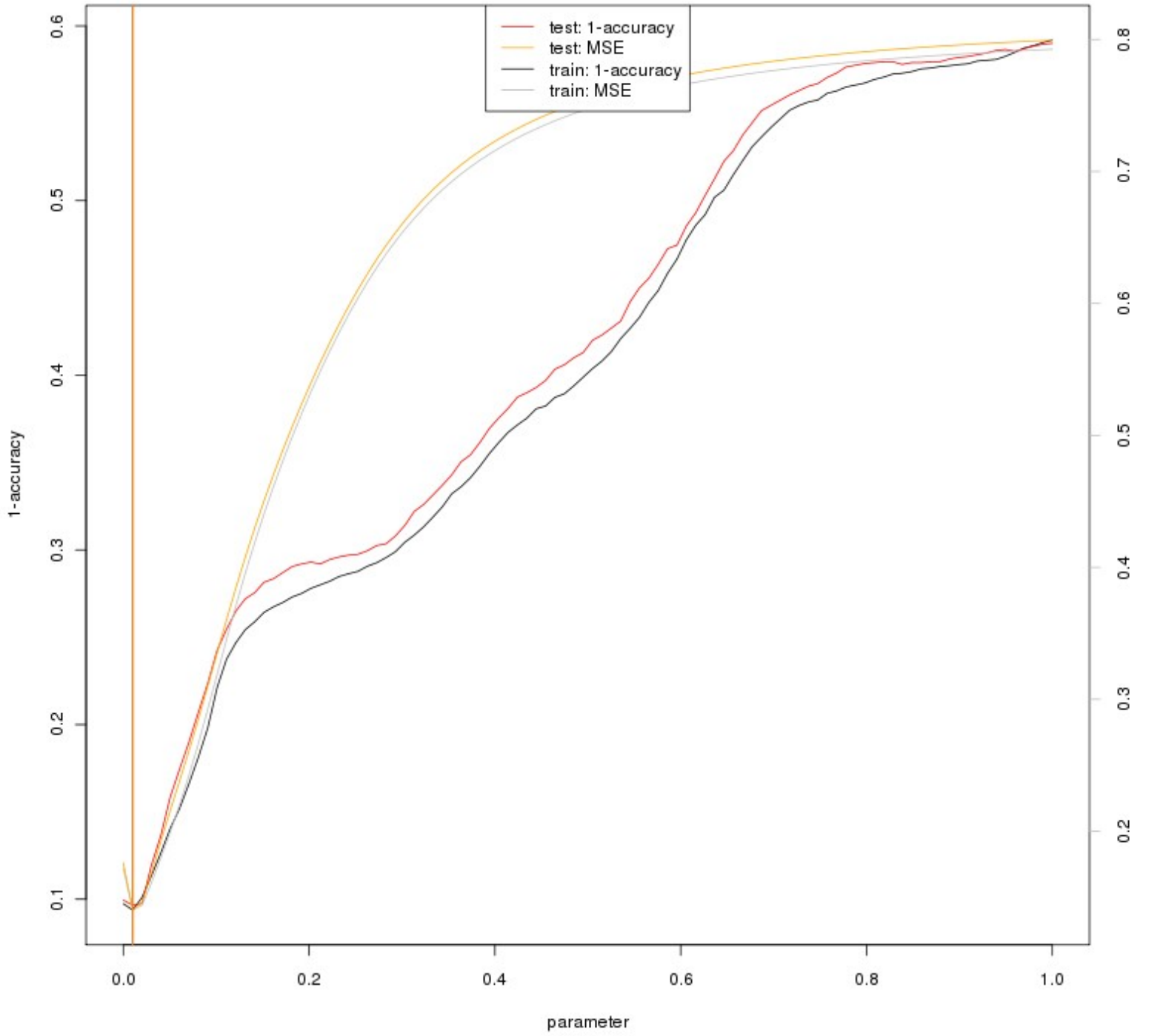

cv-bupa_liver-repetition_0-fold_0 (test inaccuracy= 0.3143)
hRange=[0.1959946, 11.01635]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.0
0.2
0.4
0.6
0.8
1.0

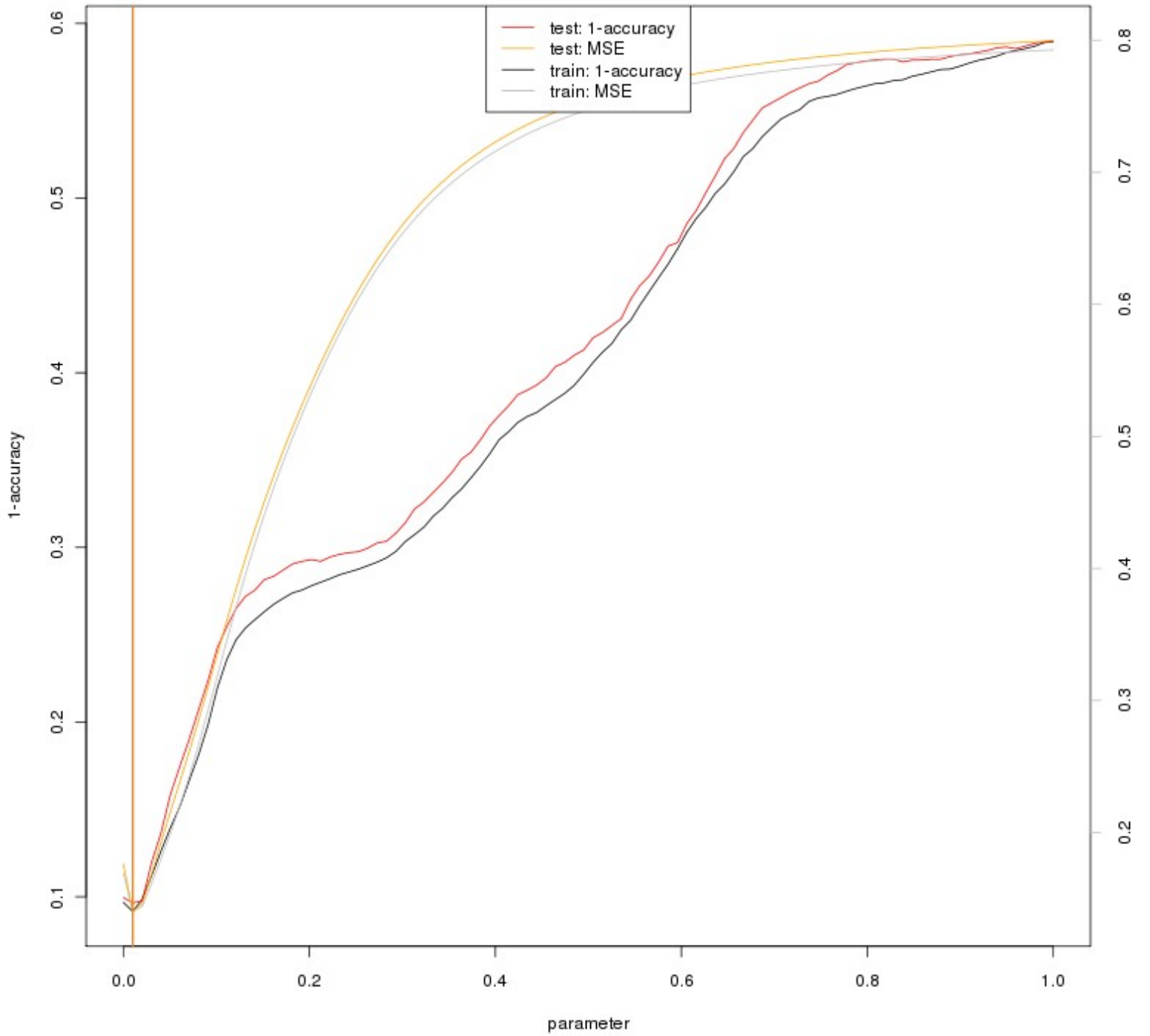

cv-glass-reduced-repetition_0-fold_0 (test inaccuracy= 0.1818)
hRange=[0.1014668, 14.67544]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7


cv-glass-reduced-repetition_1-fold_0 (test inaccuracy= 0.2273)
hRange=[0.1006569, 11.39634]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8


cv-glass-reduced-repetition_2-fold_0 (test inaccuracy= 0.1364)
hRange=[0.09443846, 11.26237]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


holdout-statlog_satellite_image-repetition_0 (test inaccuracy= 0.0965)
hRange=[0.1991850, 22.21845]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


holdout-statlog_satellite_image-repetition_1 (test inaccuracy= 0.0965)
hRange=[0.1991850, 22.21845]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

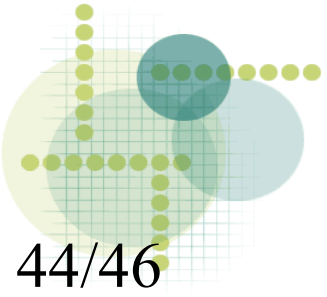

holdout-statlog_satellite_image-repetition_2 (test inaccuracy= 0.0965)
hRange=[0.1991850, 22.21845]
test: 1-accuracy
test: MSE
train: 1-accuracy
train: MSE
1-accuracy
MSE
parameter
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
0.2
0.4
0.6
0.8
1.0

Co należy zrobić

- Zastosować metody optymalizacyjne
- Lepsze określanie przedziału h
- Rozwiązać problem z jednym ze zbiorów (breast_cancer) – zbyt duża liczba przykładów o tych samych współrzędnych?
- Sprawdzić:
  - Preprocessing: PCA
  - Sprawdzić wersję ze stratified cross-validation
  - Inna liczba estymacji

Pomysły/modyfikacje

- Średnie zmiany
  - Używać innej funkcji zmniejszania się jąder
  - Używać innego jądra (np. p-Gaussian)
- B. duże zmiany
  - Dopasowywać wielkość/kształt jądra w zależności od położenia w przestrzeni



Literatura

[Lim00] Tjen-Sien Lin, Wei-Yin Loh, „A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms”, Machine Learning, 2000

[Ghosh06] Anil K. Ghosh, Probal Chaudhuri, and Debasis Sengupta, „Classification Using Kernel Density Estimates: Multi-scale Analysis and Visualization”, Technometrics, 2006

Dziękuję za uwagę!