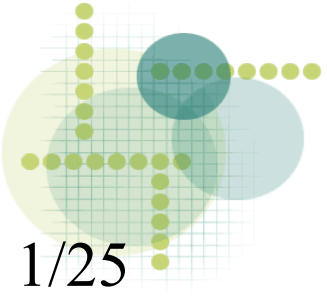
Kombinacja jądrowych estymatorów gęstości w klasyfikacji - testy na sztucznych danych

Mateusz Kobos, 25.11.2009
Seminarium Metody Inteligencji Obliczeniowej



Spis treści

- Dolne ograniczenie na wsp. wygładzania =0
- Testy na sztucznych danych
- E=k vs E<k

Opis algorytmu

- Klasyfikacja punktu *t*:
  - Estymuj gęstość w *t* za pomocą uśrednienia paru (aktualnie dwóch) estymatorów jądrowych
  - Użyj wzoru Bayesa, by uzyskać prawdopodobieństwa klas

$$d_B(\mathbf{x}) = \arg\max_{w_i} \hat{P}(\omega_i|\mathbf{x}) = \arg\max_{w_i} \frac{\hat{p}(\mathbf{x}|\omega_i)\hat{P}(\omega_i)}{\hat{p}(\mathbf{x})}$$

    - Gdy p(x)=0 (na skutek ograniczonej precyzji maszynowej), zwracamy prawdopodobieństwa z algorytmu 1-NN

Opis algorytmu

- Uczenie:
  - Minimalizuj błąd średniokwadratowy ze względu na parametry wygładzania estymatorów

$$\mathrm{MSE}(\hat{P}(\cdot), \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^{c} (\hat{P}(\omega_i|\mathbf{x}) - \mathbf{t}_i(\mathbf{x}))^2$$

  - Przy minimalizacji używamy pseudo-Newtonowskiego algorytmu L-BFGS-B [Zhu97]
    - Dla 2 estymatorów wybieramy punkt startowy:
      - x0=[1.1, 1]

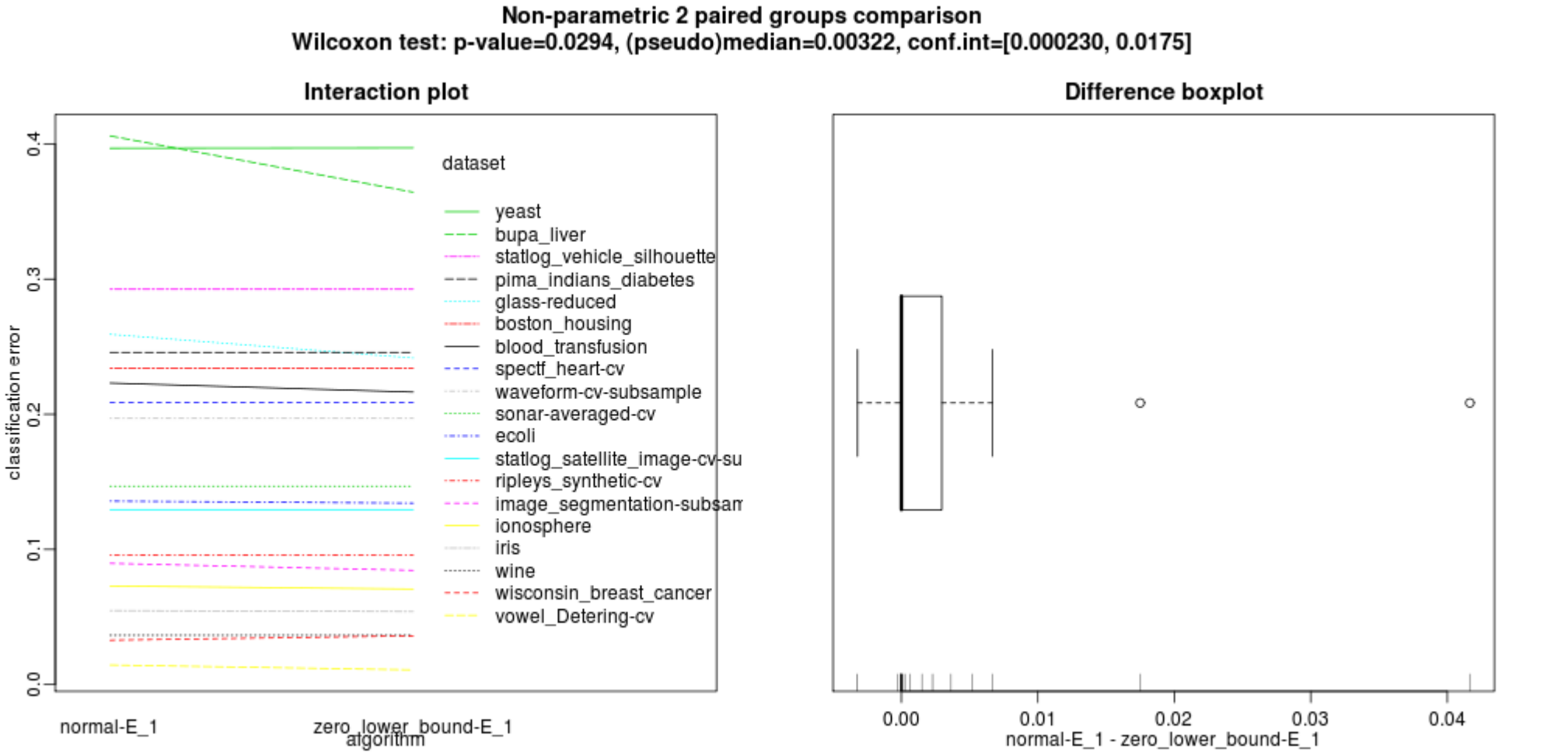
Dolne ograniczenie na wsp. wygładzania =0 vs. stara wersja

- Minimalizacja:
  - Przedziały poszukiwań współczynników wygładzania:
    - Górny: 99. percentyl odległości między przykładami
    - Dolny (**stara wersja**): 1. percentyl * 1/ (promień kuli zawierającej 99% masy rozkładu normalnego)
    - Dolny (**nowa wersja**): 0 (jeśli w którymś ze wzorów wystąpi 0 w mianowniku, to uruchamiamy alg. 1-NN)

Eksperyment

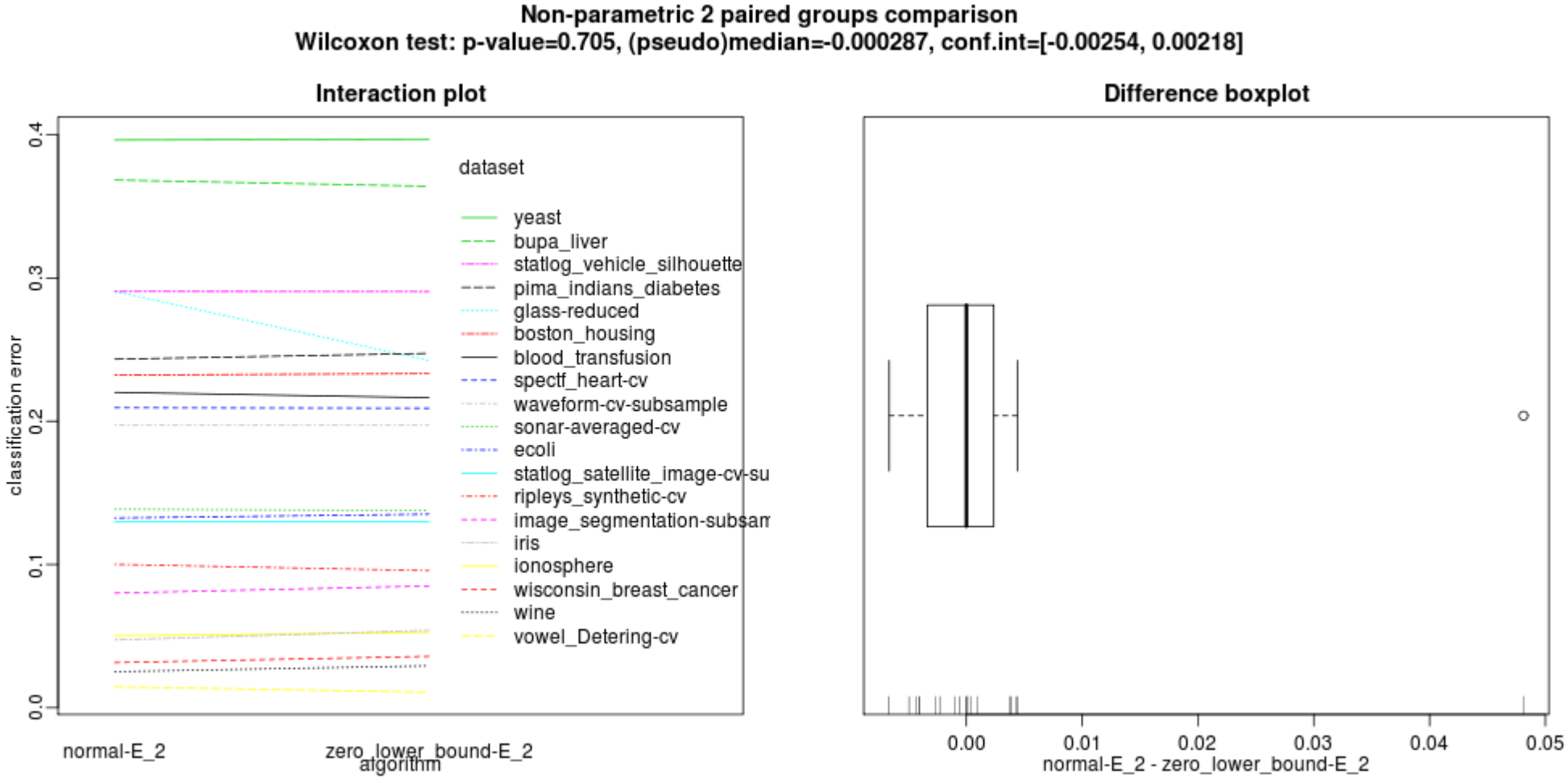
- Pytanie: Czy nowa wersja nie popsuła nic w wydajności algorytmu?
- Eksperyment:
  - Eksperyment: dla każdego algorytmu i zbioru danych: 10x 10-fold cross-validation i obliczamy średnią
    - Dla dużych zbiorów: image_segmentation, waveform, statlog_satellite_image używane są tylko 600-elementowe podzbiory
  - Test: test rangowy Wilcoxsona – zgodnie z opisem w [Demsar06]

E=1



jest istotna różnica - wersja z zerowym ograniczeniem jest lepsza,
ale w 8/19=42% przypadków nie ma różnicy między wynikami (są takie same).

E=2

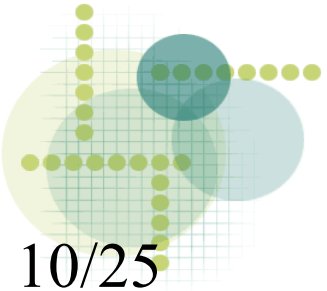


brak istotnej różnicy,
w 2/19=10% przypadków nie ma różnicy między wynikami (są takie same).

Wnioski

- Można stosować wersję z zerowym dolnym ograniczeniem na szerokość jądra, bo raczej nie ma statystycznie istotnych różnic między wynikami (raczej, bo dla porównania wersji E=1 wyszła różnica, ale i tak na korzyść wersji z zerowym ograniczeniem)
- Trzeba jeszcze sprawdzić dla tej modyfikacji, czy jest statystycznie istotna różnica w wynikach między E=1 i E=2

Testy na sztucznych danych



Notacja – na podstawie [Friedman97]

0.1 Notacja

- $\mathbf{x} = (x_1, \ldots, x_n)$ – atrybuty wejściowe
- y – atrybut wyjściowy
- $T = \{\mathbf{x}_i, y_i\}_1^N$ – zb. uczący o rozmiarze N
- W uczeniu z nadzorem:
  - $y \in R^1$ – regresja
  - $y \in \{0, 1\}$ – klasyfikacja binarna
- $f(\mathbf{x}) = Pr(y = 1|\mathbf{x})$
  - here: $f(\mathbf{x}) = \frac{\pi_1 p_1(\mathbf{x})}{\pi_0 p_0(\mathbf{x}) + \pi_1 p_1(\mathbf{x})} \in [0, 1]$
- $\hat{f}(\mathbf{x}) \in [0, 1]$ – nasza estymacja $f(\mathbf{x})$
- $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon \in \{0, 1\}$
  - ϵ – szum losowy (rozkład dwumianowy)
- $\hat{y}(\mathbf{x}) \in \{0, 1\}$ – nasza estymacja etykiety klasy

0.2 Regresja (MSE)

$$\mathrm{MSE}(\hat{f}(\mathbf{x}), y(\mathbf{x})) = E_T[y(\mathbf{x}) - \hat{f}(\mathbf{x}|T)]^2 = E_T[f(\mathbf{x}) - \hat{f}(\mathbf{x}|T)]^2 + E_\varepsilon[\varepsilon|\mathbf{x}]^2$$

$$\mathrm{MSE}(\hat{f}(\mathbf{x}), f(\mathbf{x})) = E_T[f(\mathbf{x}) - \hat{f}(\mathbf{x}|T)]^2 = [f(\mathbf{x}) - E_T\hat{f}(\mathbf{x}|T)]^2 + E_T[\hat{f}(\mathbf{x}|T) - E_T\hat{f}(\mathbf{x}|T)]^2$$

Czyli nieformalnie:

$$\mathrm{MSE}(\hat{f}, y) = \mathrm{MSE}(\hat{f}, f) + \mathrm{variance(noise)} = \Big(\mathrm{bias}(f, E\hat{f})\Big)^2 + \mathrm{variance}(\hat{f}) + \mathrm{variance(noise)}$$

0.3 Klasyfikacja (błąd klasyfikacji)

Błąd klasyfikacji uśredniony po wszystkich przykładach uczących T rozmiaru N (dla uproszczenia piszemy f zamiast $f(\mathbf{x})$):

$$Pr(\hat{y} \neq y) = |2f - 1|Pr(\hat{y} \neq y_B) + Pr(y_B \neq y)$$

Gdzie:

- $Pr(\hat{y} \neq y_B)$ – **boundary error** – prawdopodobieństwo, że nasz klasyfikator zwróci inna etykietę klasy niż klasyfikator Bayesowski
- $Pr(y_B \neq y)$ – błąd klasyfikatora Bayesowskiego (błąd, którego nie można zredukować)

Założenia przykładów

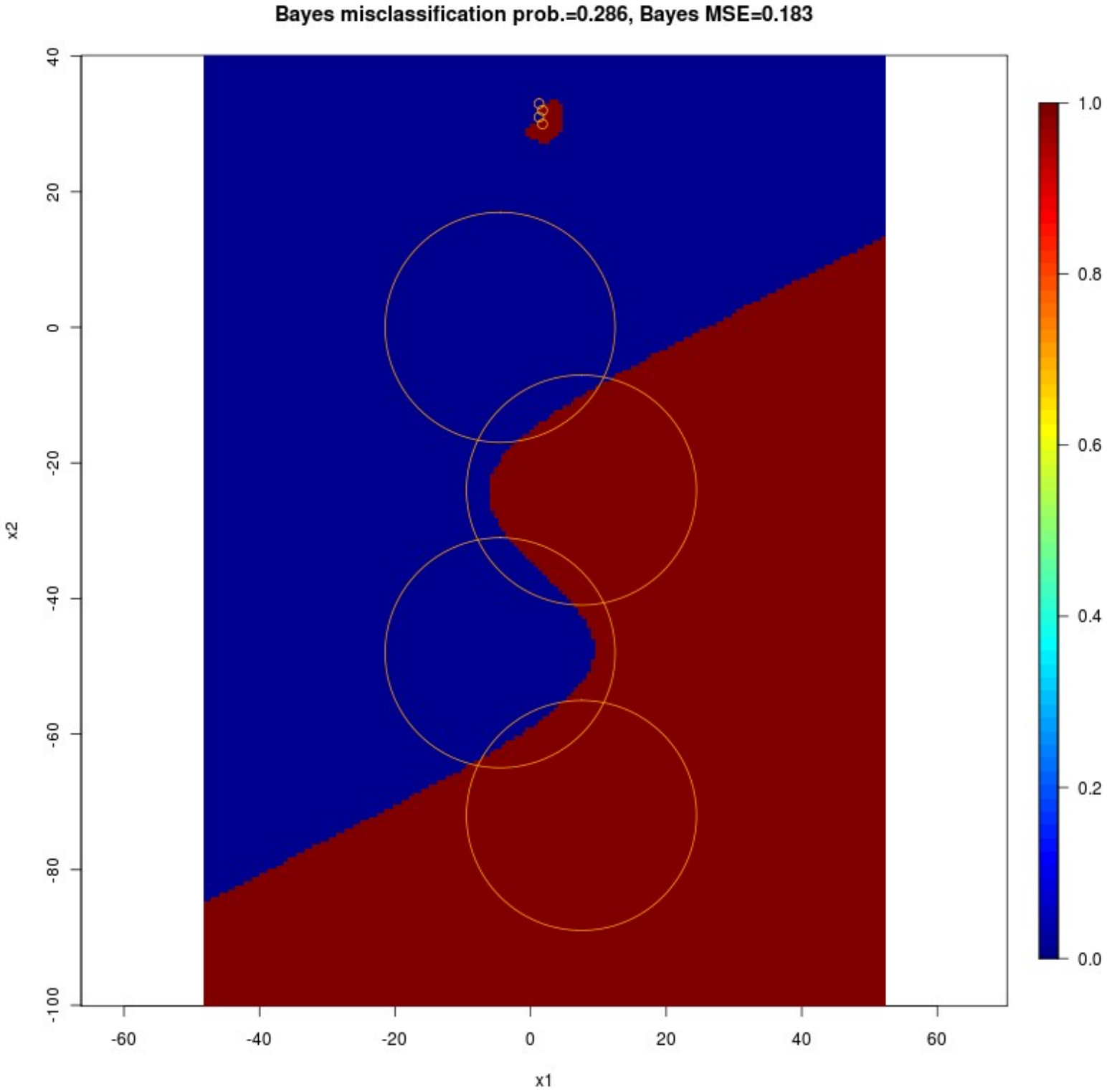
- **dla obu klas dobieramy taki sam współczynnik wygładzania** (dla *ex10-twisted* da się to łatwo uzasadnić, bo klasy są symetryczne)
- **optymalizujemy na zbiorze testowym** - czyli w idealnych warunkach, gdy wszystkie dokonywane przez nas szacunki w algorytmie są sensowne (przybliżanie błędu klasyfikacji za pomocą MSE, optymalizacja numeryczna (sensownie dobrany punkt startowy, nietrafienie w minimum lokalne), szacowanie błędu na zbiorze testowym za pomocą CV), możemy zbliżyć się do takich wyników

Parametry przykładów

- Wielkość zb. testującego: 2000
- Wielkość zb. uczącego: 50-3200
    - Liczba zbiorów uczących, po których uśredniamy: 20
- h_min=0, h_max – dobierane ręcznie dla każdego przykładu
- Dla zb. uczącego o rozmiarze 50, h_max było czasami tak dobierane, by minimum było w środku wykresu błędu a nie na brzegu
- Siatka wykresu błędu: 100x100

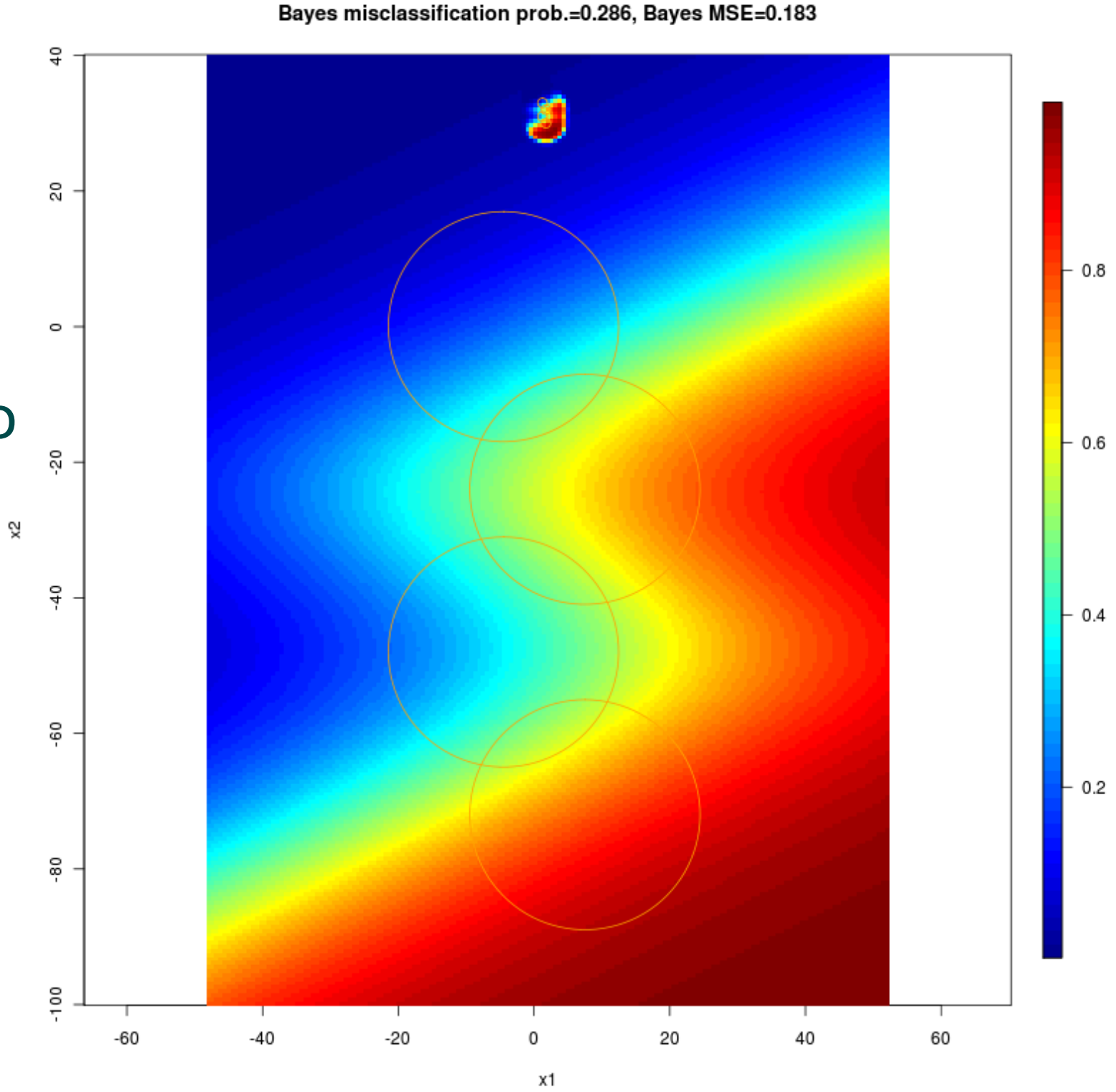
ex10

- Regiony decyzyjne klas. Bayesowskiego
- duże i małe zagęszczenie



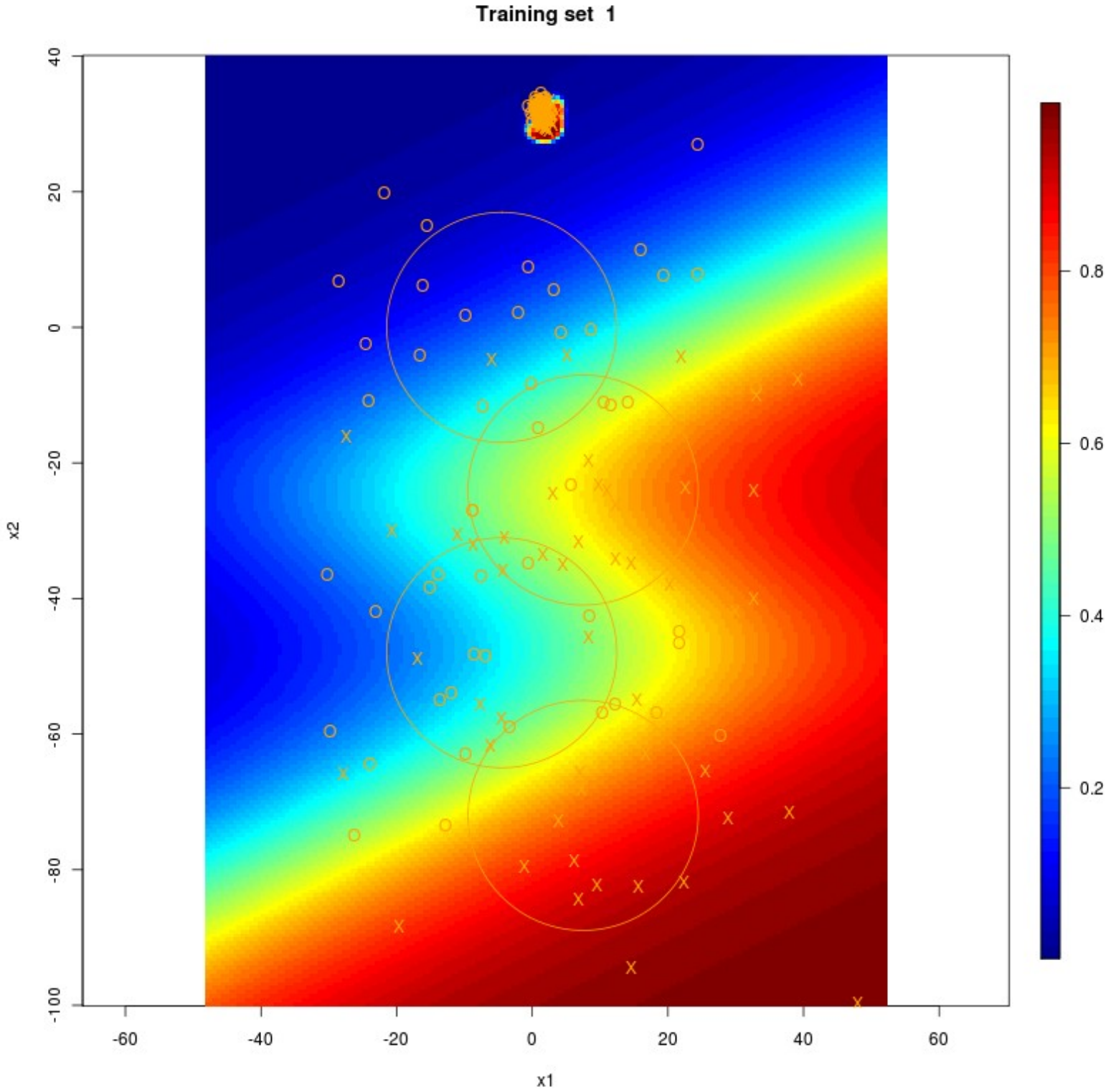
ex10

- Prawdziwe prawd. przynależności do klas



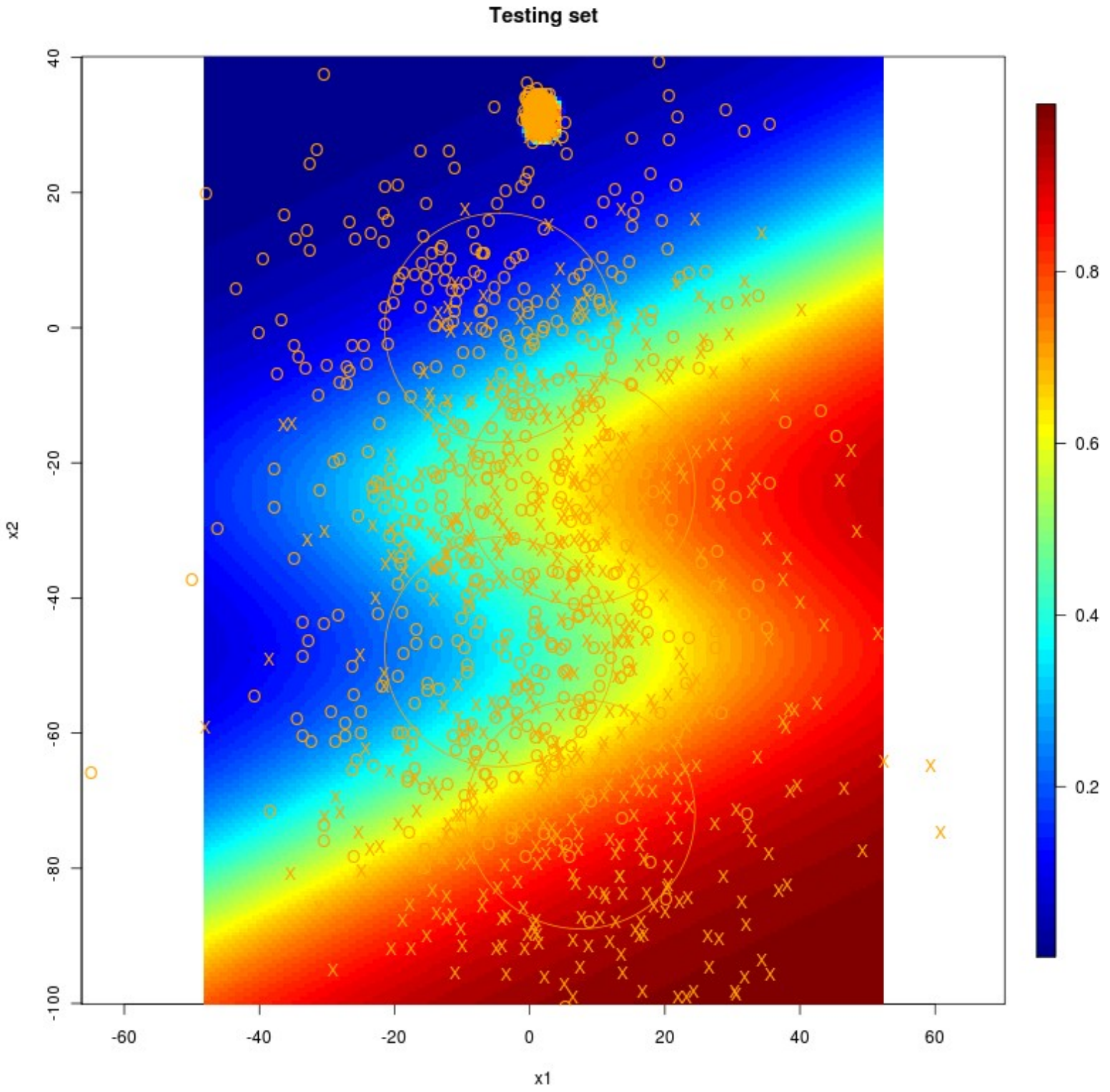
ex10

- Zbiór trenujący

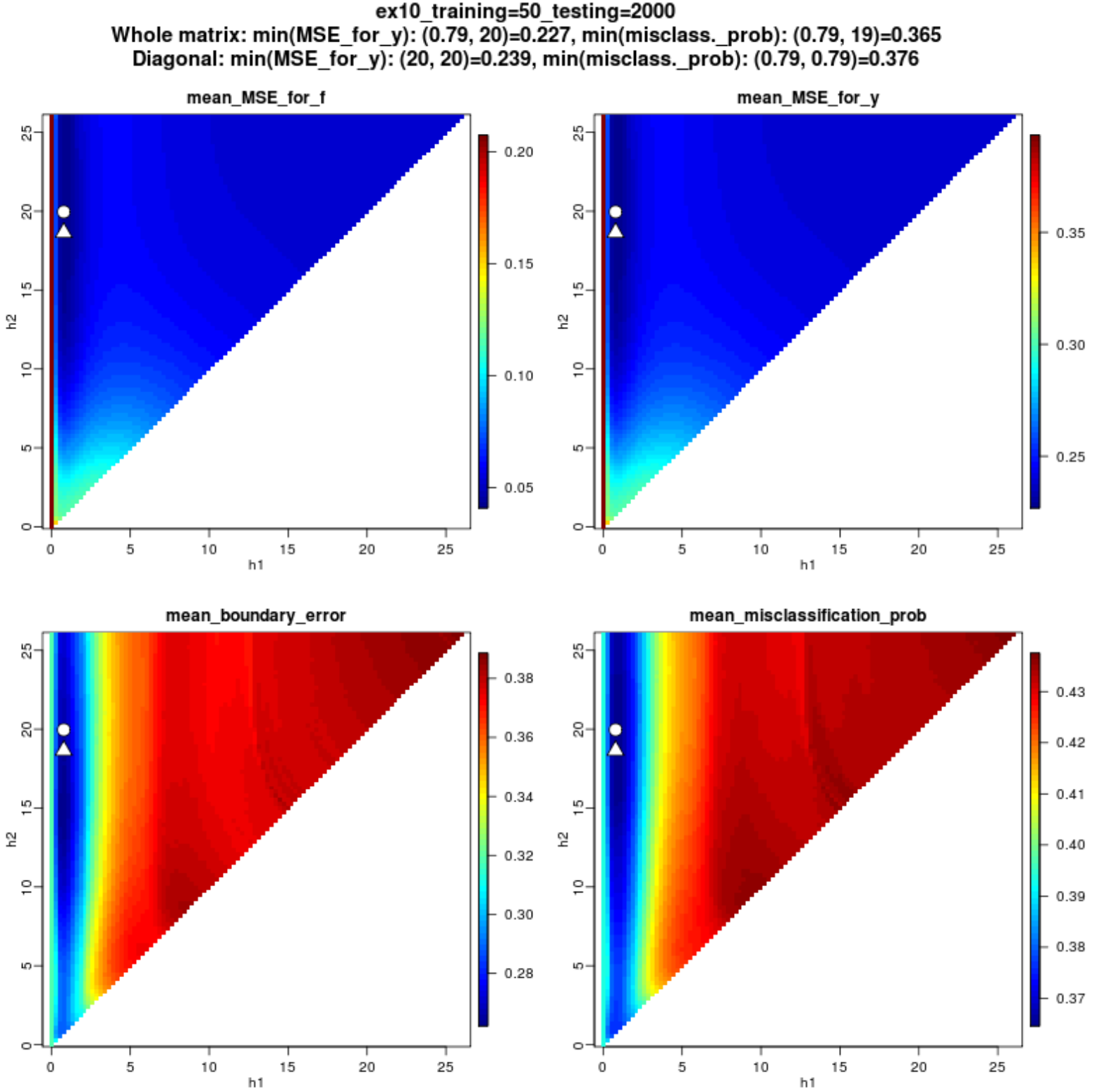


ex10

- Zbiór testujący

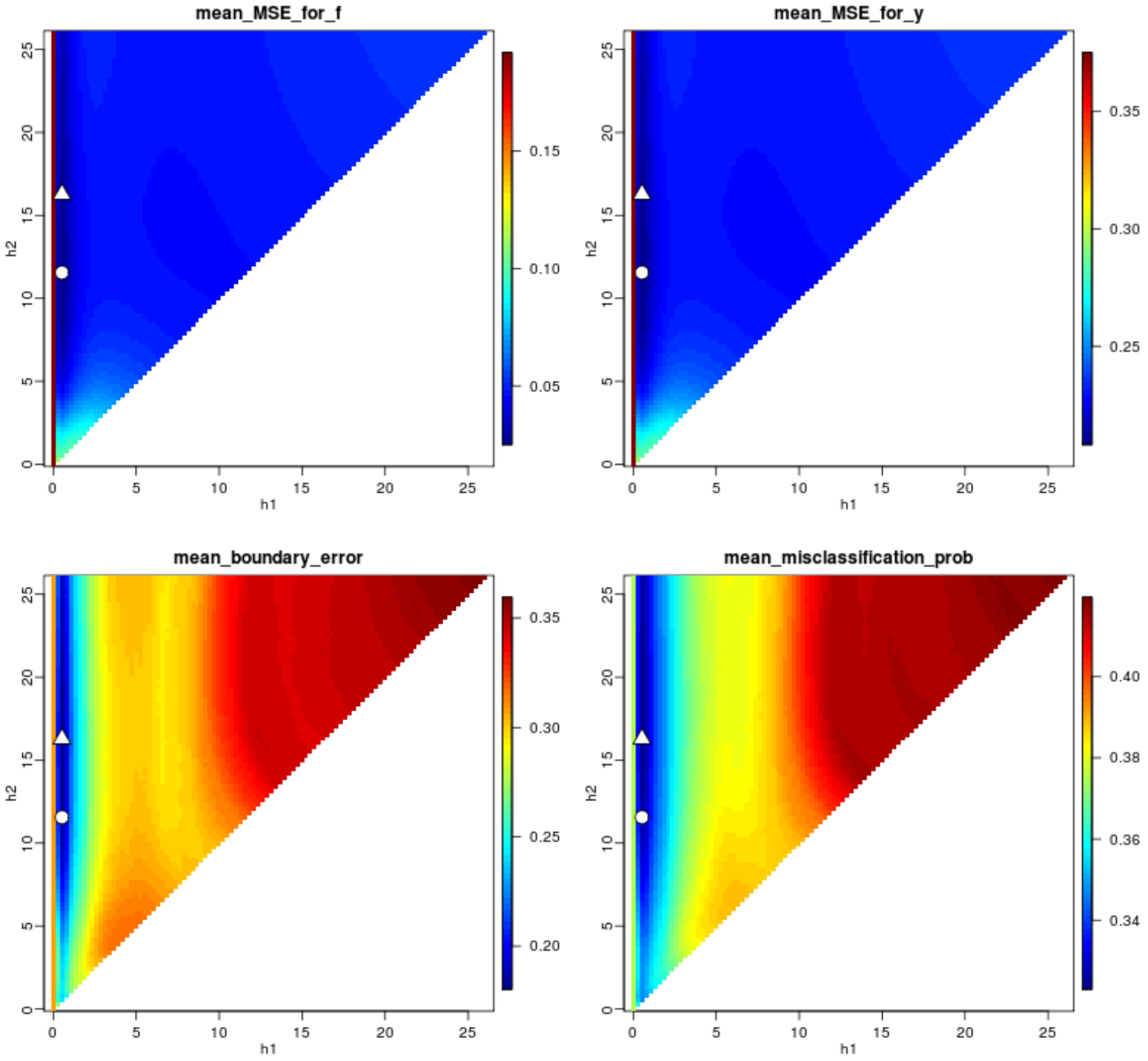


ex10

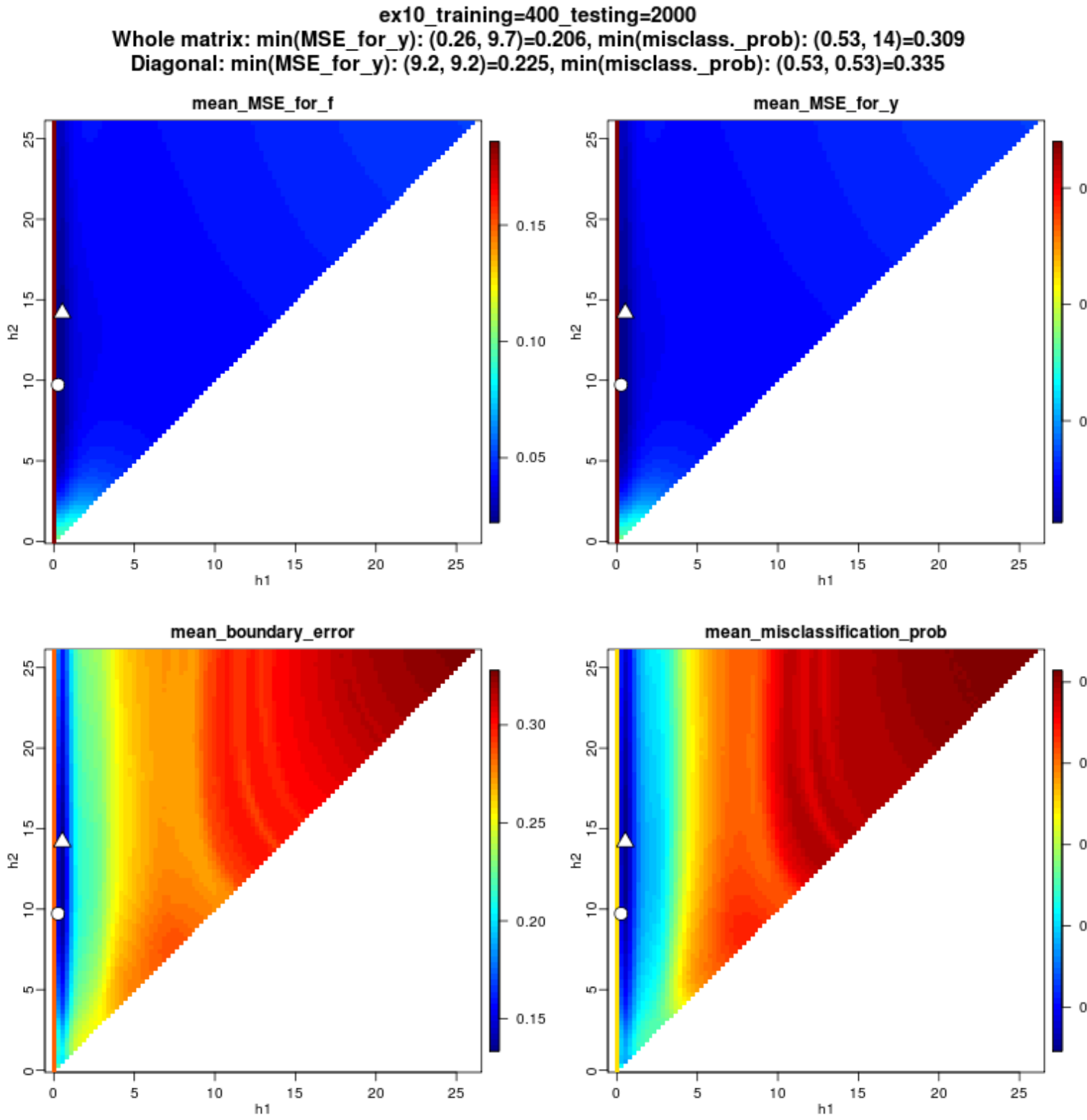

ex10_training=50_testing=2000
Whole matrix: min(MSE_for_y): (0.79, 20)=0.227, min(misclass._prob): (0.79, 19)=0.365
Diagonal: min(MSE_for_y): (20, 20)=0.239, min(misclass._prob): (0.79, 0.79)=0.376
mean_MSE_for_f
mean_MSE_for_y
mean_boundary_error
mean_misclassification_prob
h1
h2

ex10

ex10_training=200_testing=2000
Whole matrix: min(MSE_for_y): (0.53, 12)=0.208, min(misclass._prob): (0.53, 16)=0.323
Diagonal: min(MSE_for_y): (11, 11)=0.229, min(misclass._prob): (0.53, 0.53)=0.349

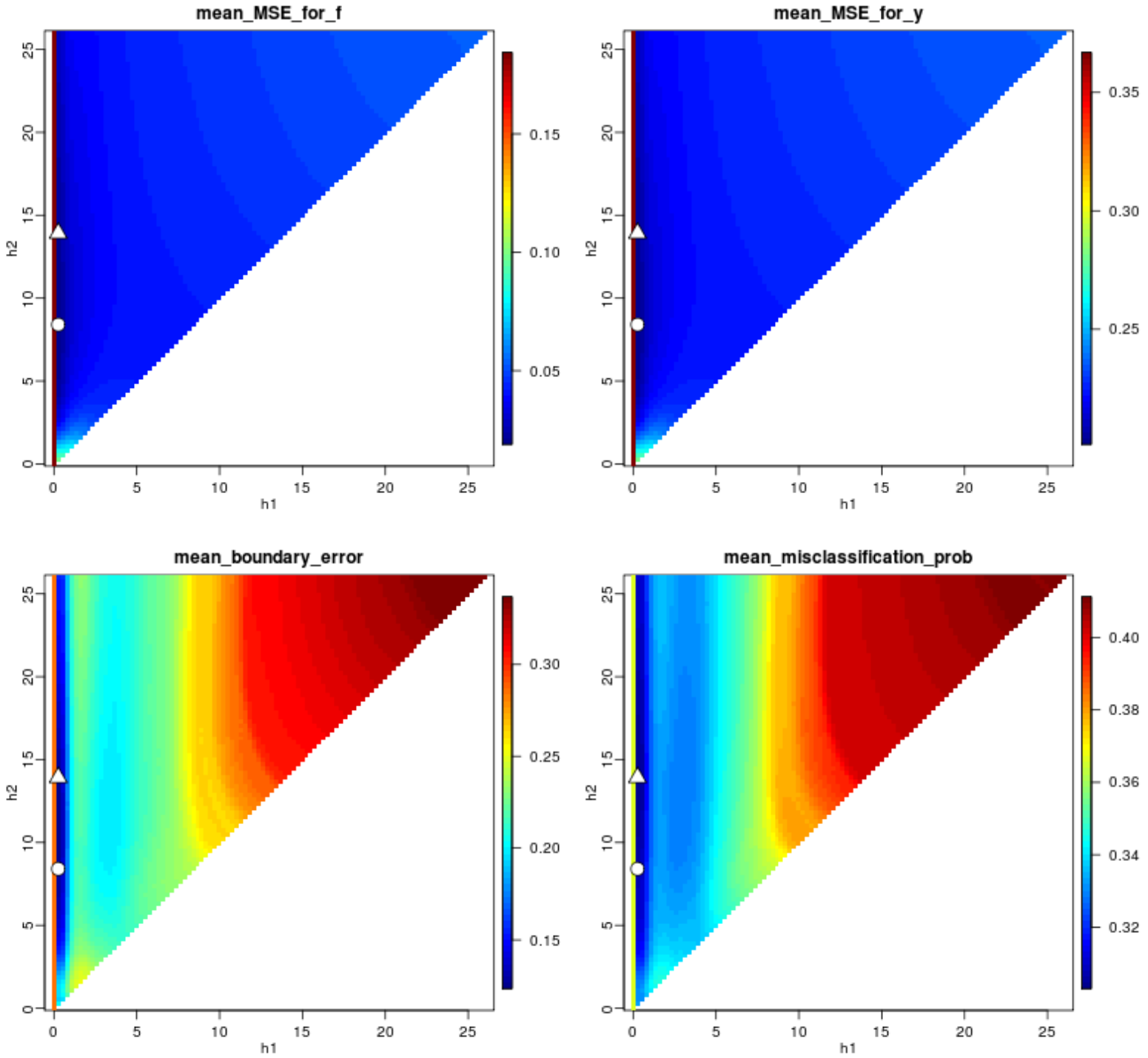


ex10


ex10_training=400_testing=2000
Whole matrix: min(MSE_for_y): (0.26, 9.7)=0.206, min(misclass._prob): (0.53, 14)=0.309
Diagonal: min(MSE_for_y): (9.2, 9.2)=0.225, min(misclass._prob): (0.53, 0.53)=0.335
mean_MSE_for_f
mean_MSE_for_y
mean_boundary_error
mean_misclassification_prob
h1
h2
0
5
10
15
20
25
0.05
0.10
0.15
0.20
0.25
0.30

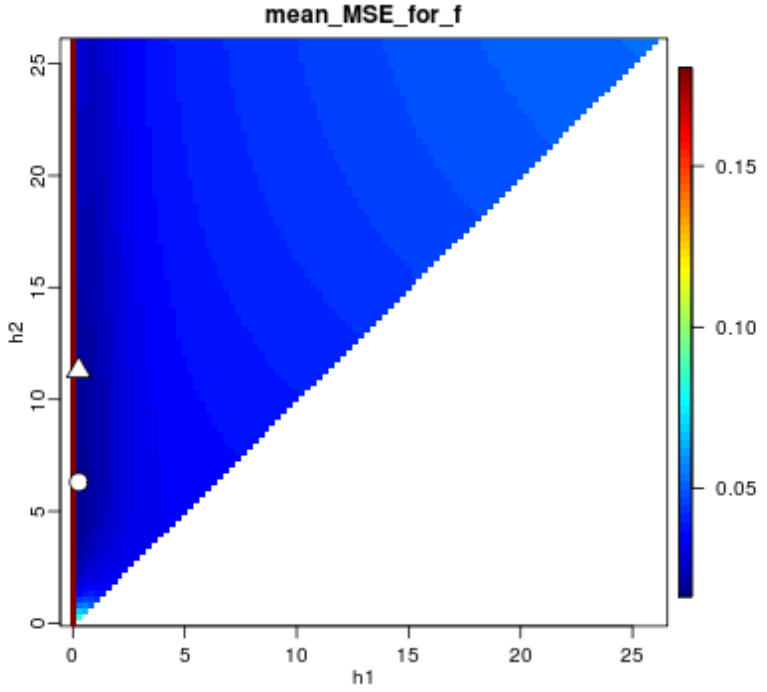
ex10

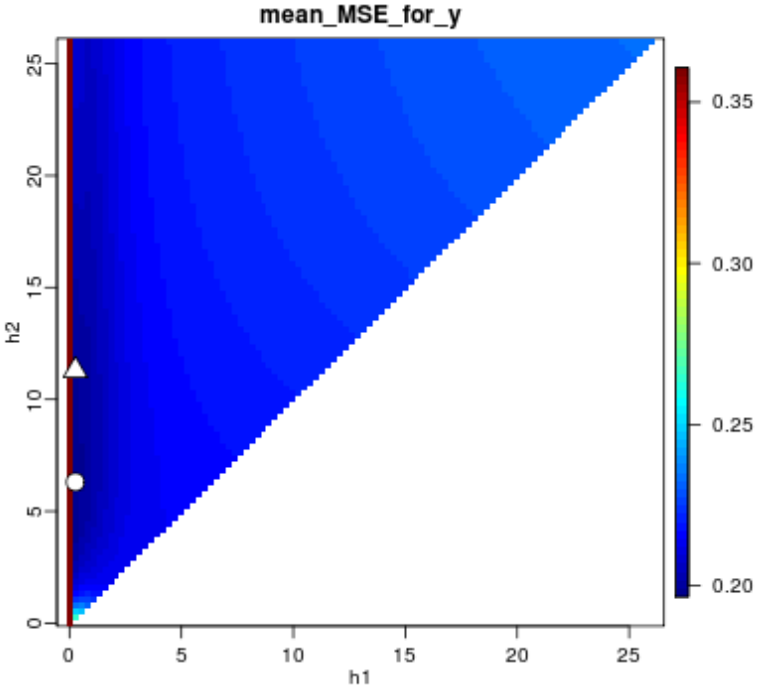
ex10_training=800_testing=2000
Whole matrix: min(MSE_for_y): (0.26, 8.4)=0.201, min(misclass._prob): (0.26, 14)=0.303
Diagonal: min(MSE_for_y): (7.1, 7.1)=0.224, min(misclass._prob): (0.53, 0.53)=0.33

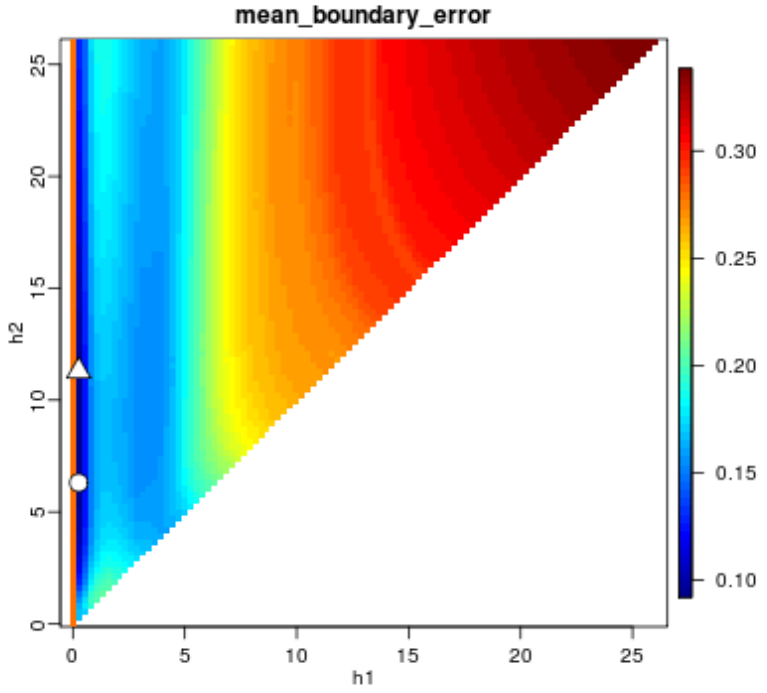


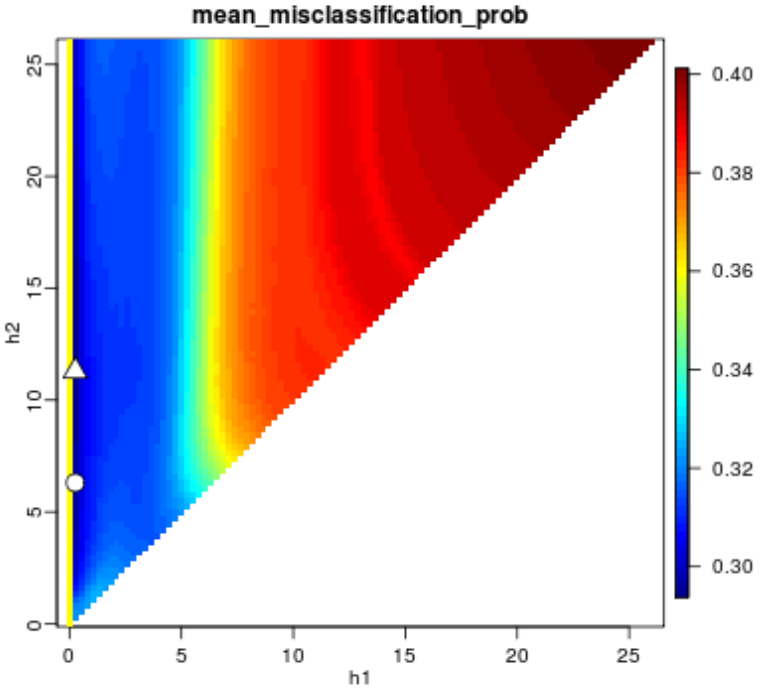
ex10

ex10_training=3200_testing=2000
Whole matrix: min(MSE_for_y): (0.26, 6.3)=0.197, min(misclass._prob): (0.26, 11)=0.294
Diagonal: min(MSE_for_y): (2.4, 2.4)=0.213, min(misclass._prob): (3.9, 3.9)=0.315



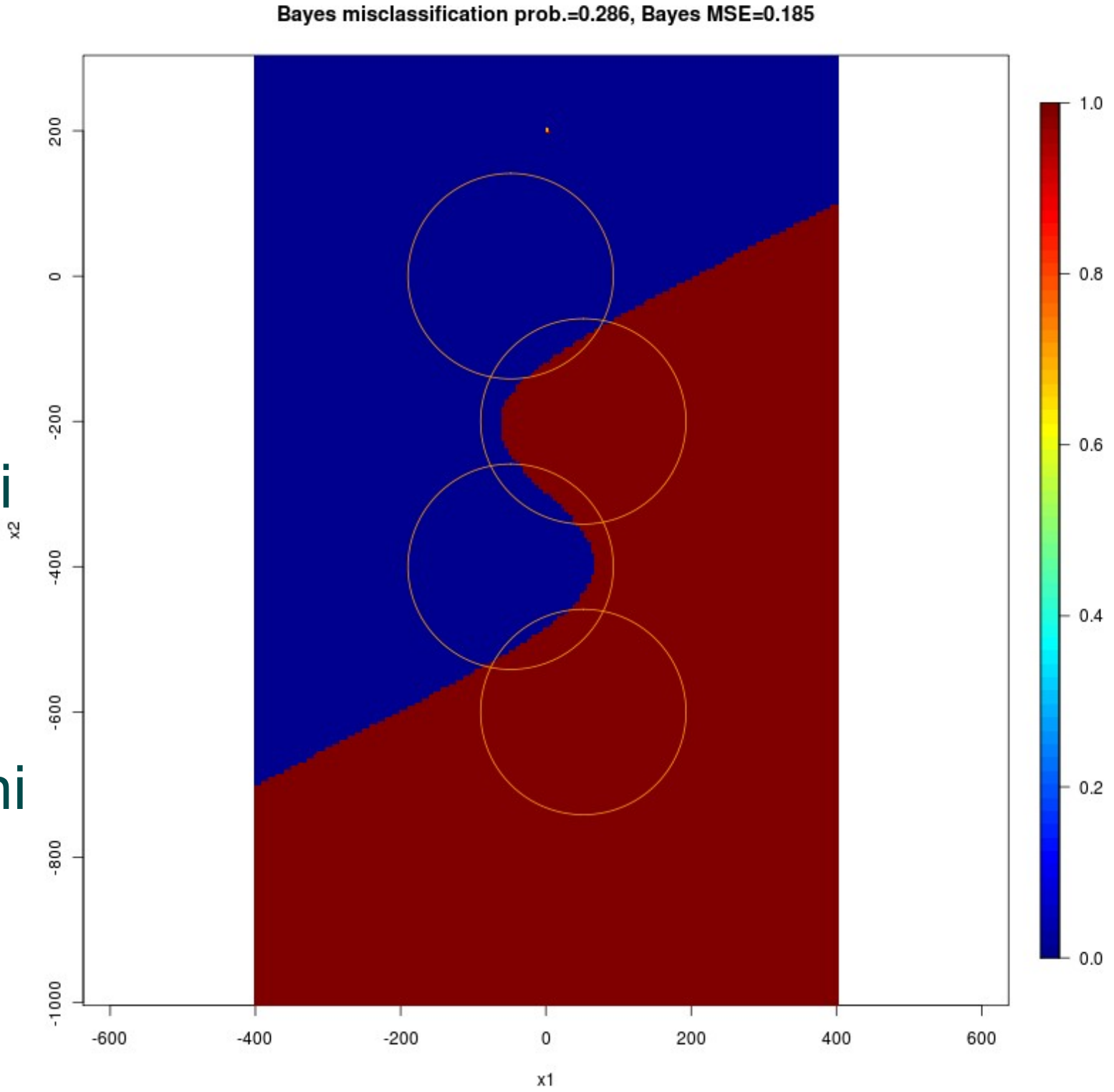




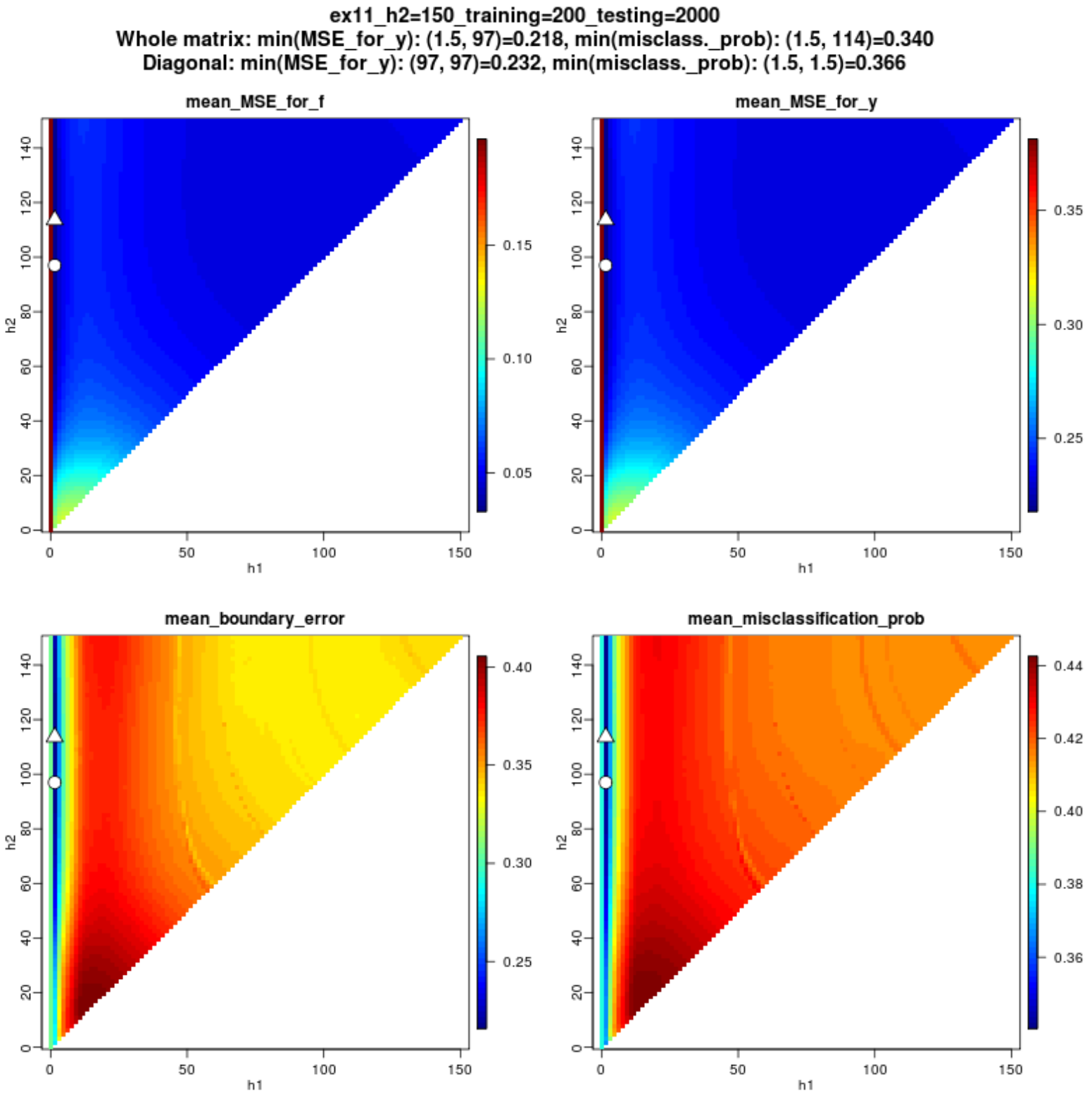


ex11

- Regiony decyzyjne klas. Bayesowskiego
- duże i b. małe zagęszczenie

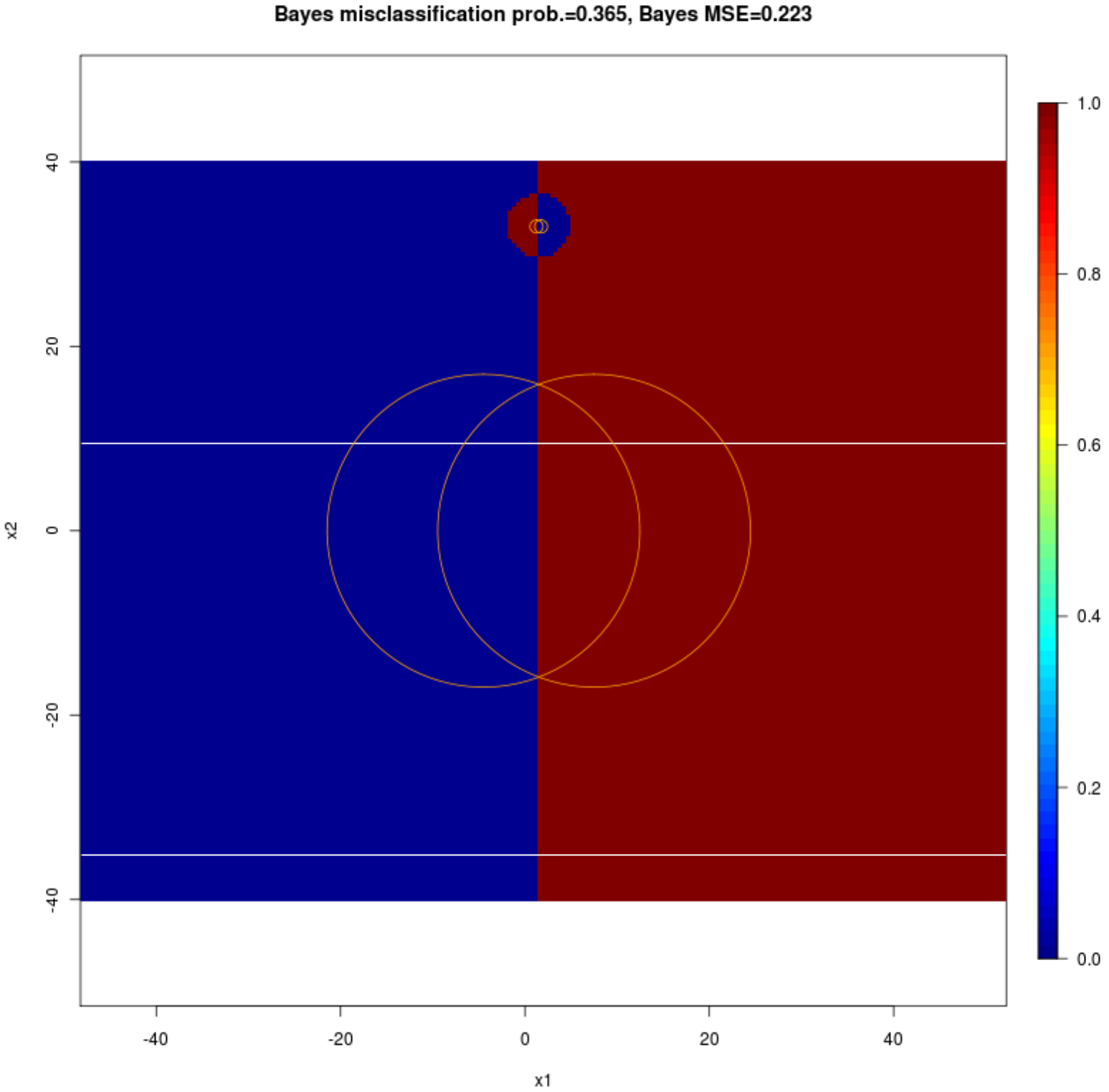


ex11

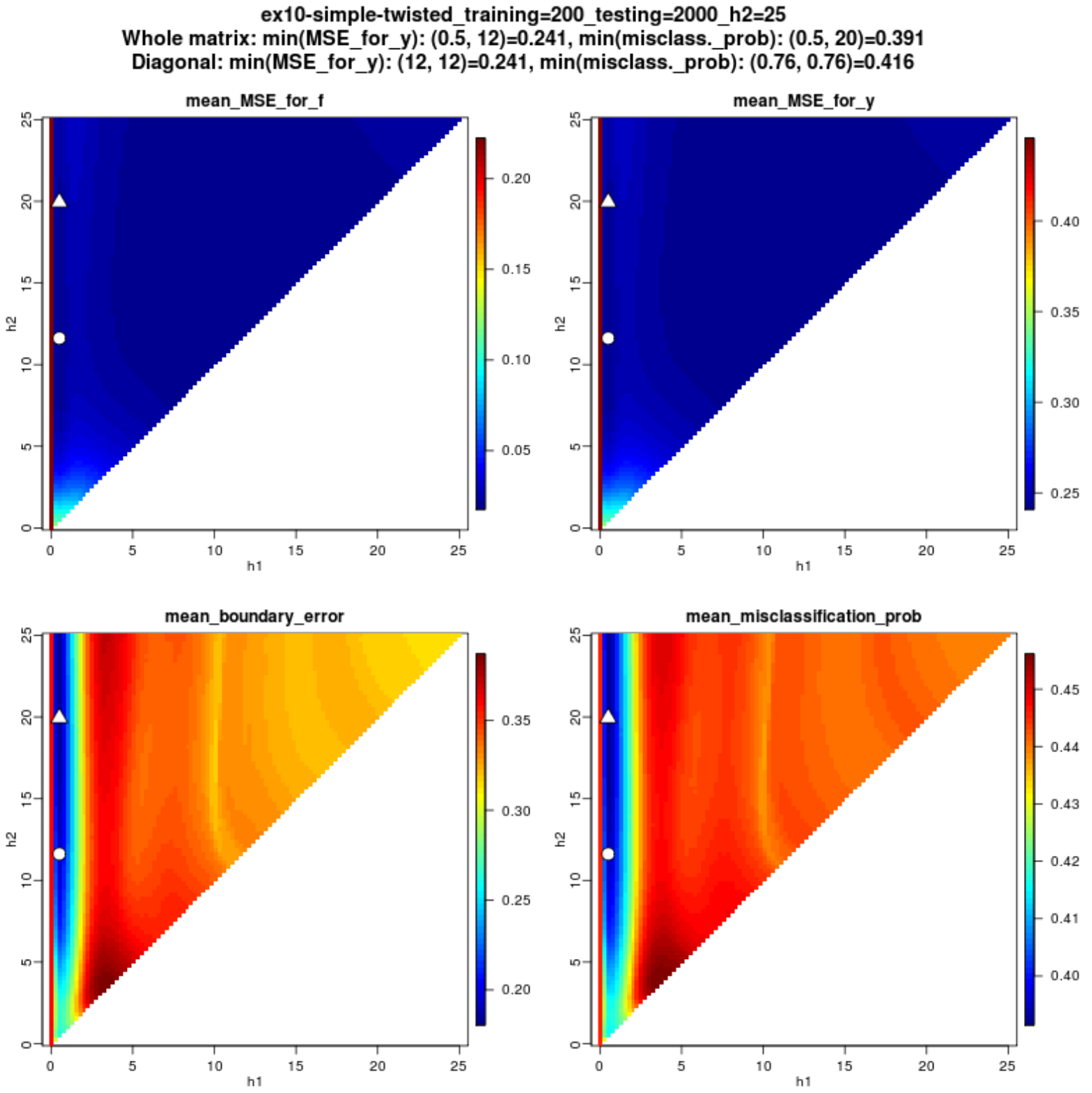

ex11_h2=150_training=200_testing=2000
Whole matrix: min(MSE_for_y): (1.5, 97)=0.218, min(misclass._prob): (1.5, 114)=0.340
Diagonal: min(MSE_for_y): (97, 97)=0.232, min(misclass._prob): (1.5, 1.5)=0.366
mean_MSE_for_f
mean_MSE_for_y
mean_boundary_error
mean_misclassification_prob
h1
h2

ex10-twisted

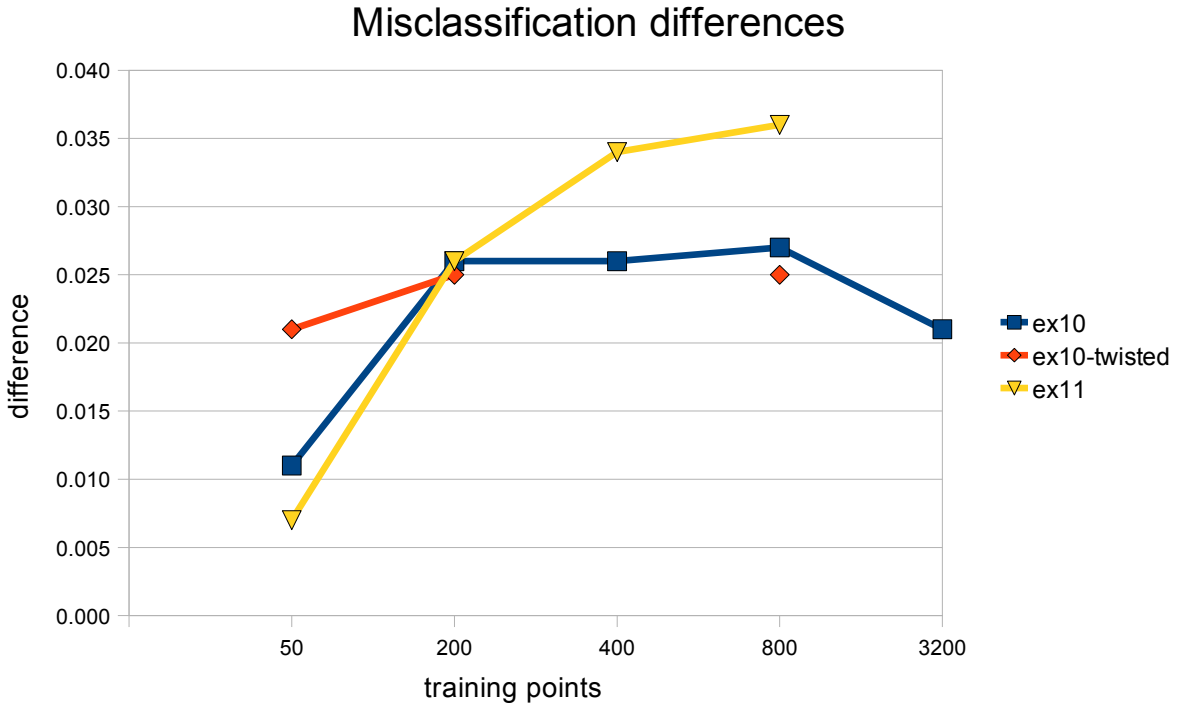
- Regiony decyzyjne klas. Bayesowskiego
- Uproszczona wersja ex10, klasy są symetryczne

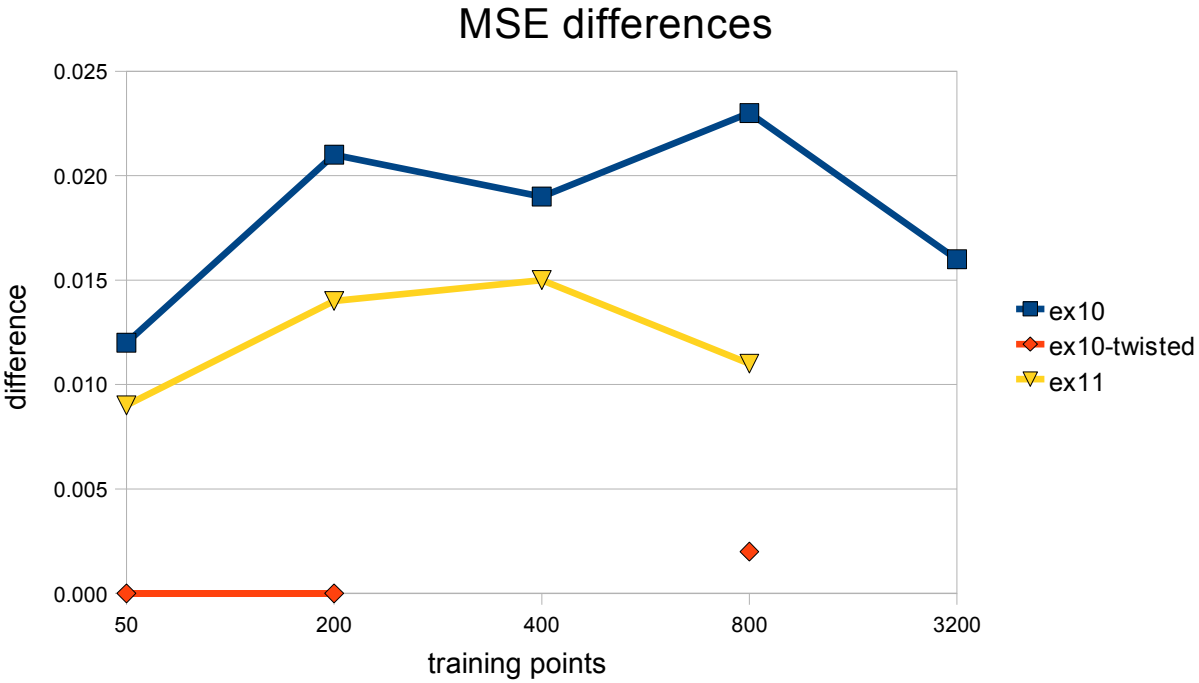


ex10-twisted



Zestawienie wyników





Wnioski

- można dobrać takie sztuczne przykłady, by była różnica między przypadkiem, gdy E=1 i E=2 (na korzyść E=2)
- własności tych przykładów: granica klas przebiega przez obszar o dużej i małej gęstości
    - hipotetyczne wyjaśnienie: dla obszarów o dużej gęstości dominuje mała szerokość jądra (wpływ szerokich jąder jest tu wszędzie podobny, więc nie ma większego znaczenia) a dla obszarów o małej - duża (bo wpływ małego jądra szybko wygasa)

Wnioski cd.

- Przy większej różnicy w gęstości regionów (ex11 vs. Ex10) różnica w jakości klasyfikacji wydaje się być bardziej znacząca (lepsza)

E=k vs. E<k

- Czy E=k zawsze da nie gorsze rozwiązanie niż E<k?
  - na pewno dla danego E=m zawsze można znaleźć E>m postaci $E = i\,m,\ i \in \{2,3,\ldots\}$

    które da wynik nie gorszy

Przyszłość

- Określić własności algorytmu – uzasadnienie, że użycie więcej niż 2 estymatorów zmniejsza błąd (najlepiej: klasyfikacji)
    - Wstępne wyniki dla różnicy gęstości klas (bias i variance estymacji)
- Startować alg. optymalizacyjny z mniej arbitralnego punktu
    - np. dla każdej klasy obliczyć wsp. wygładzania h_AMISE, który minimalizuje „Asymptotic Mean Integrated Squared Error” estymacji gęstości, przy założeniu, że rozkład jest normalny (ew. metoda Sheater-Jones)
- Transformacja każdej z klas oddzielnie
- Dobierać liczbę użytych estymatorów
- Porównać z algorytmami: Naive Bayes, k-NN

Literatura

[Demsar06] Demsar J. „Statistical Comparisons of Classifiers over Multiple Data Sets”, Journal of Machine Learning Research, 2006

[Friedman97] Friedman, J. H., „On Bias, Variance, 0/1-loss and the Curse-of-Dimensionality”, Data Mining and Knowledge Discovery 1, pp. 55-77, 1997

[Zhu97] C. Zhu, R. H. Byrd and J. Nocedal. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization (1997), ACM Transactions on Mathematical Software, Vol 23, Num. 4, pp. 550 - 560.

Dziękuję za uwagę!