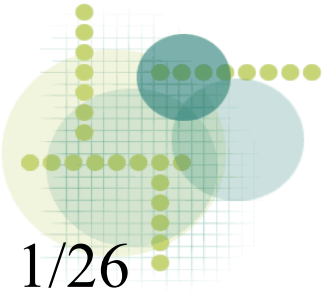
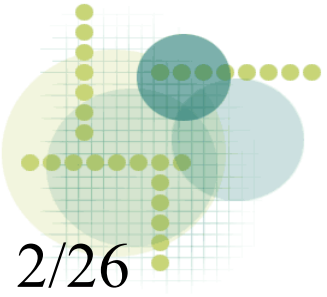
Kombinacja jądrowych estymatorów gęstości w klasyfikacji – kontynuacja prac

Mateusz Kobos, 25.03.2009
Seminarium Metody Inteligencji Obliczeniowej



Spis treści



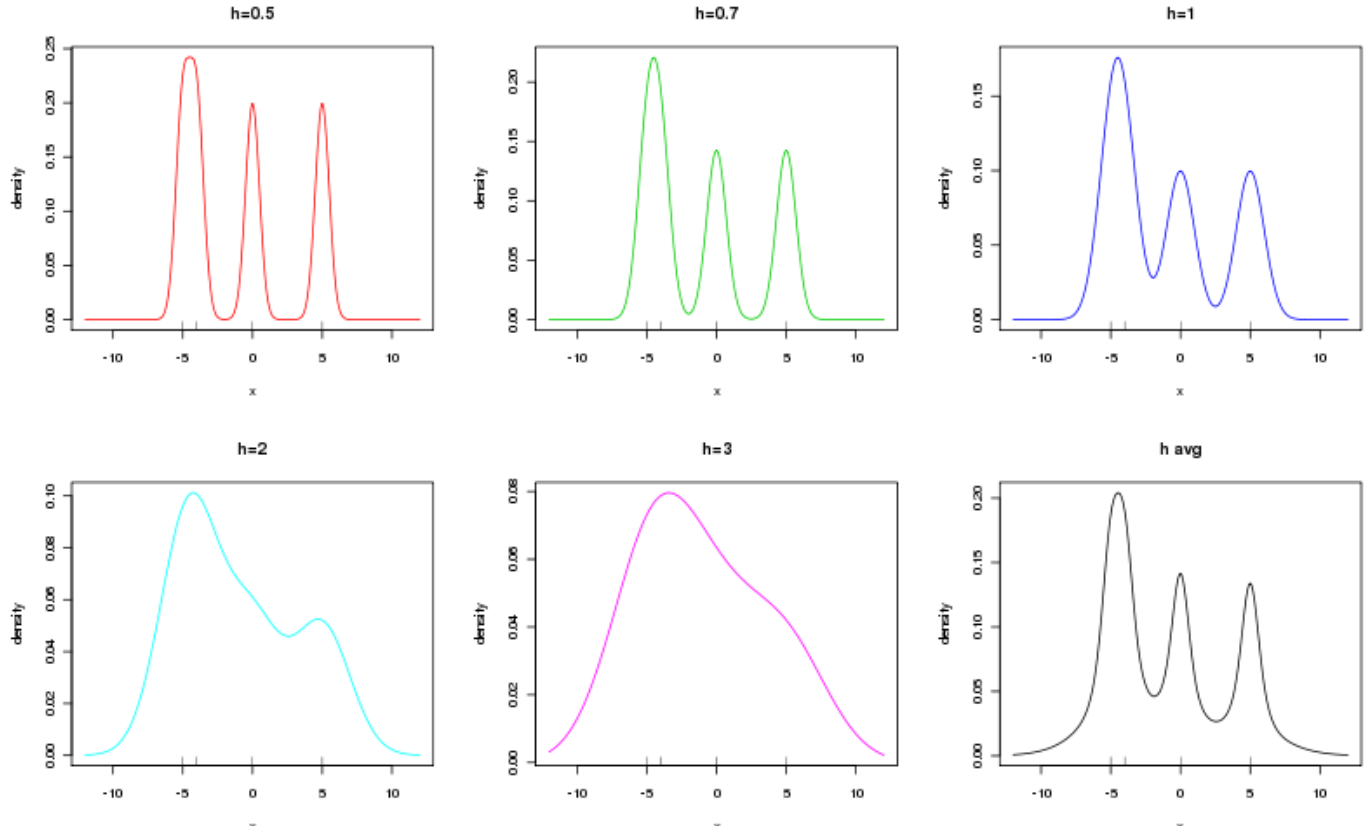
Działanie algorytmu - klasyfikacja/odtwarzanie

- Dla punktu testowego:
  - Estymuj gęstość każdej z klas punkcie (korzystając z parametrów obliczonych podczas nauki)
  - Zwróć etykietę klasy odpowiadającej największej gęstości (obliczamy korzystając ze wzoru Bayesa)

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

Czemu kombinacja estymatorów jądrowych?

- Dokonujemy estymacji jądrowej dla różnych szerokości jądra i uśredniamy
- Różne szerokości jądra odpowiadają różnym „rozdzielczościom” spojrzenia na dane

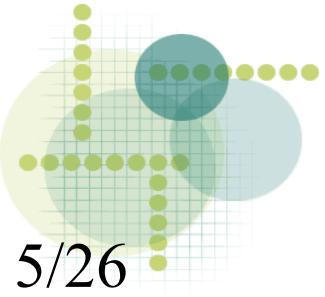


Kombinacja

- Szerokości jąder każdego z estymatorów są połączone za pomocą funkcji wykładniczej:

$$h_j(a) = h_{\min} + a^j(h_{\max} - h_{\min})$$

- Gdzie:
  - *h_j* – szerokość jądra estymatora *j*
  - *[h_min, h_max]* – przedział sensownych szerokości jądra
  - *a* należy do *[0, 1]*



Obliczanie przedziału [h_min, h_max]

- Metoda podobna do tej z [Ghosh06]: Badamy rozkład odległości w zbiorze danych, a następnie:
  - *h_max* = 99-ty percentyl odległości między punktami (podejście konserwatywne)
  - *h_min* – w przeciwieństwie do *h_max* musi być dobrane ostrożnie, bo:
    - We wzorze Bayes'a: Dla $h \to 0$, gęstość $p(x) \approx 0$ i wtedy dzielimy przez zero - niedobrze!

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

    - Dla $h \to 0$ estymacja gęstości jest niedobra

Obliczanie h_min

- Zabezpieczenia przed zbyt małymi wartościami h:
  - *1/ξ* części 1-go percentyla odległości między punktami
    - Dla jądra gaussowskiego: $\xi = \sqrt{F^{-1}_{\chi^2(d)}(0.99)}$
      - Gdzie: *d* – liczba wymiarów
    - To zabezpieczenie nie zawsze jest wystarczające
  - Wiemy, że w ekstremalnych sytuacjach (b. małe *h*, lub, równoważnie, obserwacja b. mocno odstająca) klasyfikacja za pomocą estymatora jądrowego zachowuje się jak alg. Nearest Neighbor (1-NN)
    - Dla *p(x)≈0* zwracamy wynik klasyfikacji 1-NN

Uczenie - ogólnie

- Minimalizacja brutalna średniokwadratowego (MSE) błędu cross-walidacyjnego (ze stratyfikacją) klasyfikacji

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

- 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 $\mathbf{x}$,
- $\mathbf{t}(\mathbf{x})[i]$ - actual value of point $\mathbf{x}$ probability of class ω_i

Uczenie - dokładniej

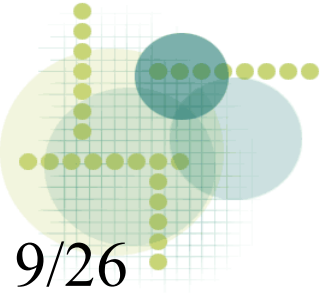
1)Losowa permutacja wszystkich instancji zbioru treningowego

2)Transformacja danych (np. standaryzacja, PCA)

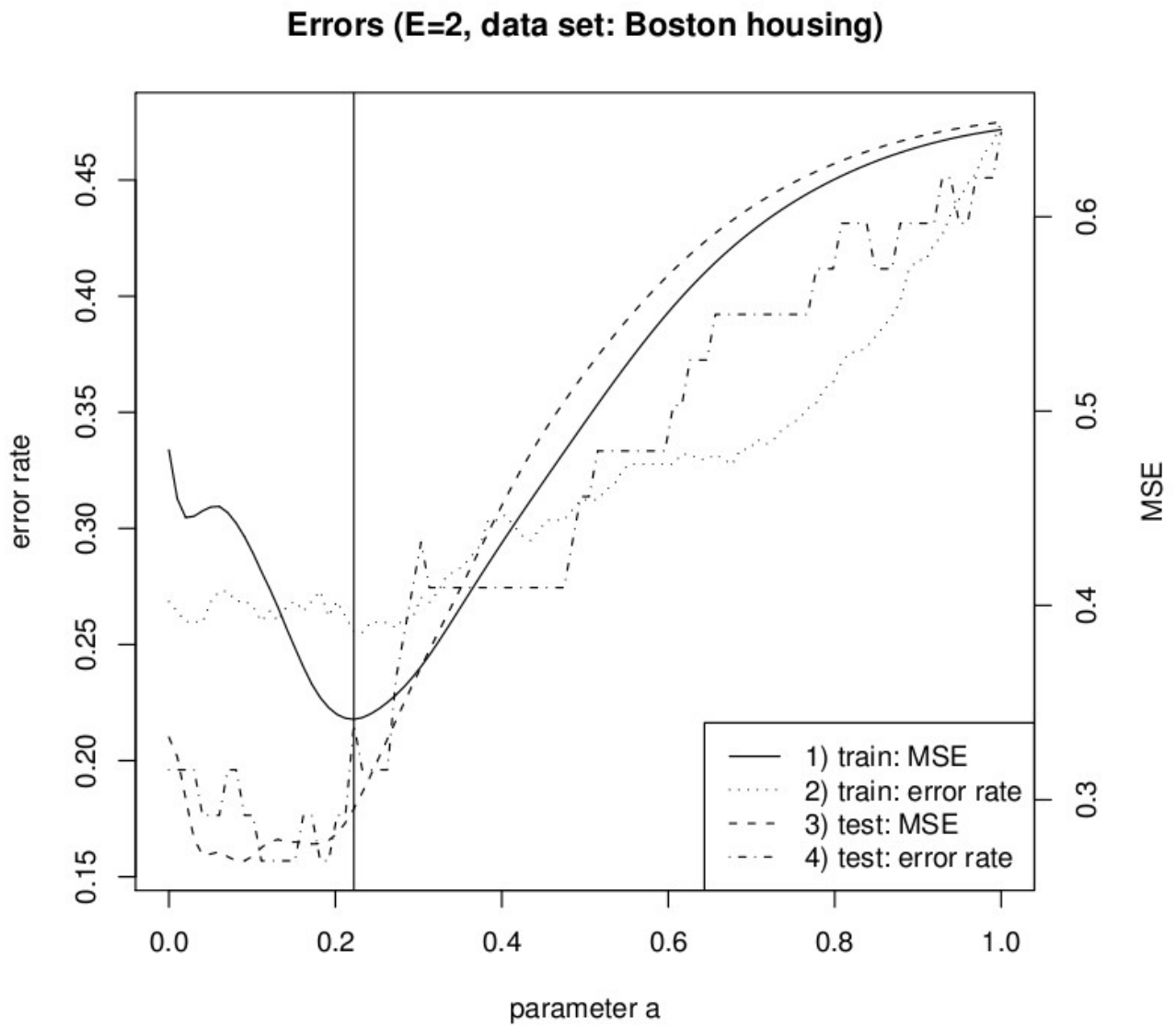
3)Obliczenie *[h_min, h_max]*

4)Poszukiwanie optymalnego a za pomocą metody brutalnej – optymalizacji grid-search cross-walidacyjnej (ze stratyfikacją) funkcji MSE

- sprawdzanych jest 100 wartości funkcji



Minimalizacja MSE - przykład



Eksperymenty - metodologia

- 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
  - Cross-validation: Bez zdefiniowanego zbioru testowego: 10-fold stratified cross-validation
- Wersja algorytmu:
  - Transformacja: standaryzacja, PCA whitening

Eksperymenty - wyniki

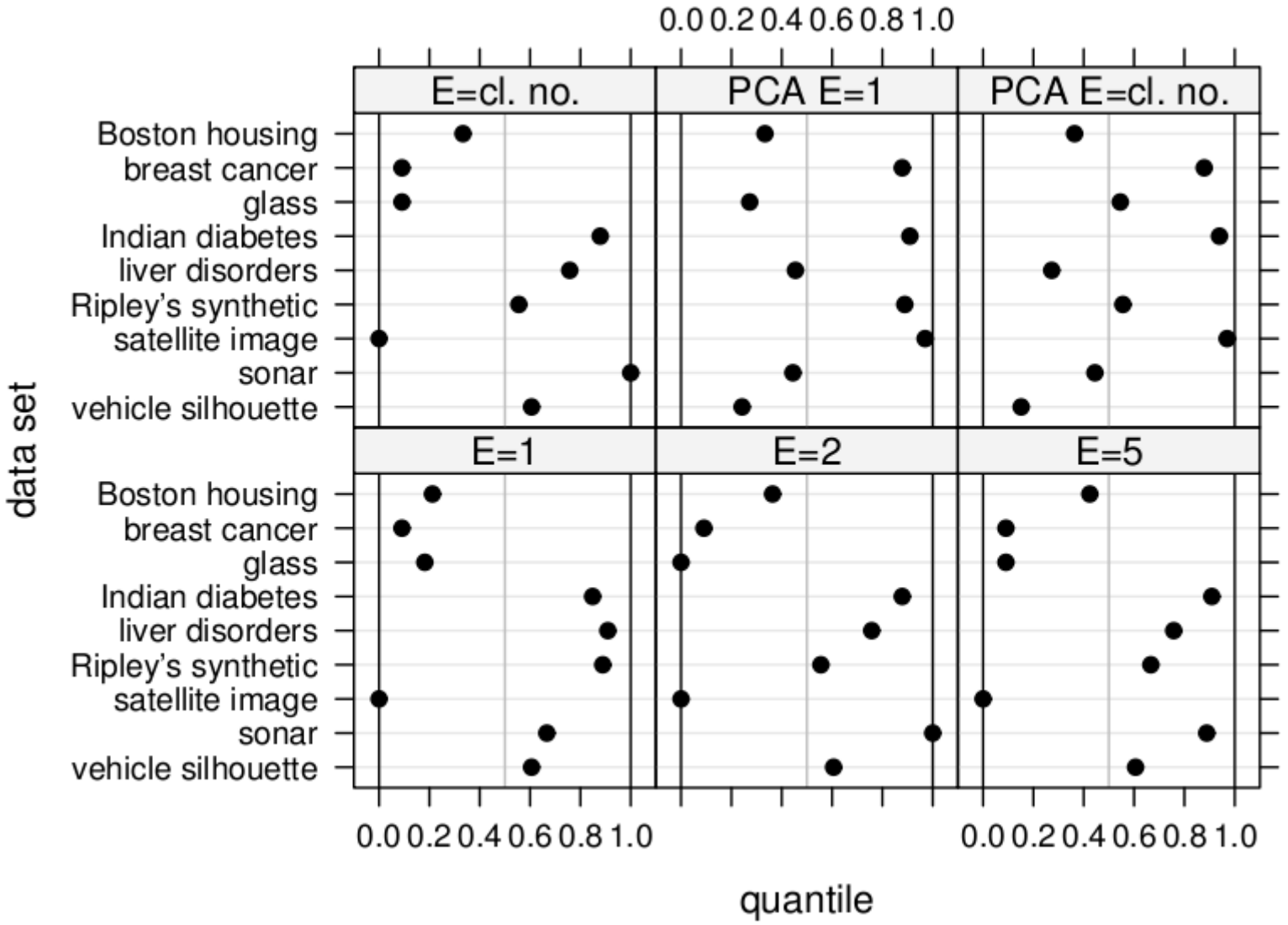
name	classes no	attributes no	instances	source
wisconsin breast cancer	2	9	683	[Lim00]
BUPA liver disorders	2	6	345	[Lim00]
PIMA indians diabetes	2	7	532	[Lim00]
Boston housing	3	13	506	[Lim00]
StatLog satellite image	6	36	6435	[Lim00]
StatLog vehicle silhouettes	4	18	846	[Lim00]
Ripley's synthetic	2	2	1250	[Ghosh06]
Sonar averaged	2	20	208	[Ghosh06]
Glass (reduced)	6	5	214	[Ghosh06]

data set	best lit.	E=1	E=2	E=5	E=cl.no.	PCA E=1	PCA E=cl.no.
Boston housing	.221	.239	.247	.249	.245	.243	.247
breast cancer	.0278	.0323	.0323	.0323	.0323	.0661	.0661
glass	.236	.252	.233*	.247	.247	.271	.308
Indian diabetes	.221	.251	.256	.259	.256	.264	.269
liver disorders	.279	.405	.365	.365	.365	.321	.310
Ripley's synthetic	.090	.105	.094	.097	.094	.105	.094
satellite image	.098	.097*	.095*	.095*	.096*	.292	.288
sonar	.135	.194	.222	.216	.222	.181	.181
vehicle silhouette	.145	.286	.285	.284	.286	.208	.205

Wnioski

- Są różnice względem E=1 (chociaż niewielkie)
  - Wersje *E=2, E=5, E=cl.no*. lepsze od wersji *E=1*
  - Wersje *PCA E=1, PCA E=cl.no.* gorsze od wersji *E=1*
- *E=2* uzyskuje 2 wyniki lepsze niż literaturowe

Porównanie z wynikami literaturowymi



Wnioski

- Dla wszystkich wersji co najmniej 4/9 wyników należy do górnych 50% wyników literaturowych
- Są różnice względem E=1 (chociaż niewielkie)
  - Wersje *E=2, E=cl.no*. lepsze od wersji *E=1*
  - Wersje E=5, *PCA E=1, PCA E=cl.no.* gorsze od wersji *E=1*
- Wniosek końcowy: wersja *E=2* wydaje się najlepsza

Pytania oraz pomysły/modyfikacje

- Pytania:
  - Czy nauka algorytmu z 1 estymatorem w tej wersji jest nowa?
- Małe zmiany
  - Używać skali logarytmicznej dla parametru *a*
- 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

Optymalizacja z wykorzystaniem pochodnej

- Algorytm optymalizacyjny: L-BFGS-B [Zhu97]
  - pseudo-Newtonowski, znajduje optimum lokalne
  - Przeznaczony do rozwiązywania dużych, nieliniowych problemów optymalizacyjnych, z ograniczoną pojemnością pamięci
  - Każda ze zmiennych może być ograniczona ($l \leq x_i \leq u$), tutaj: $0 \leq a \leq 1$
  - W każdym kroku wymaga jednoczesnego obliczenia wartości funkcji i pochodnej
- Co optymalizujemy: cross-walidacyjną (ze stratyfikacją) funkcję błędu średniokwadratowego

Zalety i wady optymalizacji z wykorzystaniem pochodnej

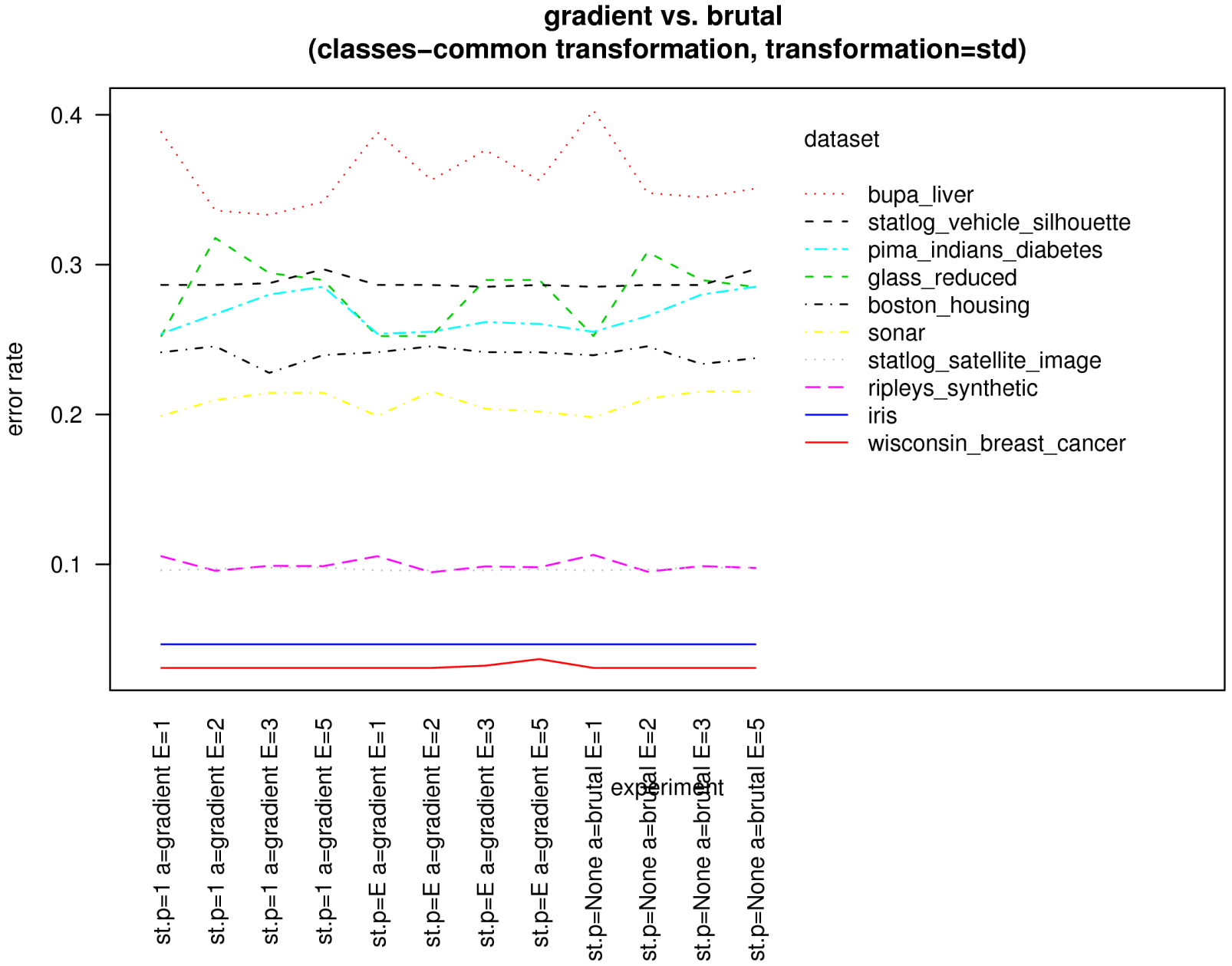
- Zalety
  - Poszukiwanie minimum trwa o ok. 75% krócej niż w metodzie brutalnej
- Wady
  - Trzeba ustalać punkt startowy,
  - Znajdowane jest minimum lokalne
- Własności:
  - Liczba ewaluacji funkcji i gradientu to średnio ok. 8.4

Aktualne problemy z tym podejściem

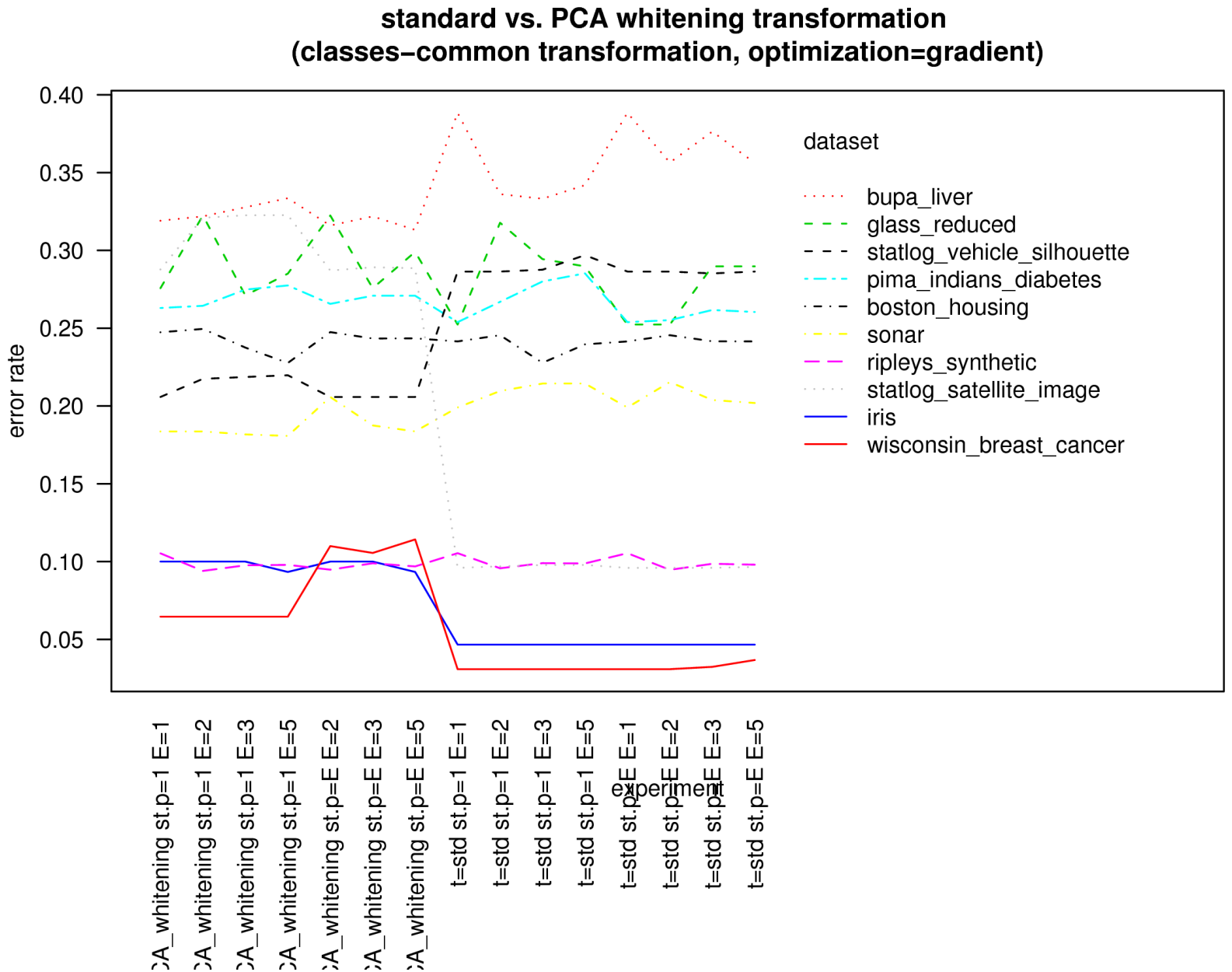
- Czasami program wyrzuca wyjątek (dzielenie przez 0)
- Dobieranie odpowiedniego punktu startowego
  - Aktualnie: h1=1, h1=E (były też testy h1=h_min, h1=h_max)
  - Pomysł: Zastosować jedną z popularnych metod dobierania punktu optymalnego dla estymacji gęstości za pomocą estymatorów jądrowych (Sheater-Jones, smoothed sross-validation)

Wstępne testy dla wersji z pochodną

- Metodologia:
  - Zaproponowana w [Demsar06] – porównywanie wielu algorytmów na wielu zbiorach danych:
    1) Sprawdzamy, czy rezultaty wszystkich algorytmów są takie same: Friedman test
    2) Jeśli nie, to sprawdzamy, które są różne:
       - Porównując z algorytmem bazowym: Holm test lub Bonferroni-Dunn test
       - Porównując wszystkie ze wszystkimi: Nemenyi test
  - Wykonałem tylko krok 1)
- Testowano różne parametry: liczba estymatorów (*E*), punkty startowe (*st.p.*), algorytmy optymalizacyjne (*a*), transformacje danych (*t*), typ transformacji (*type*)

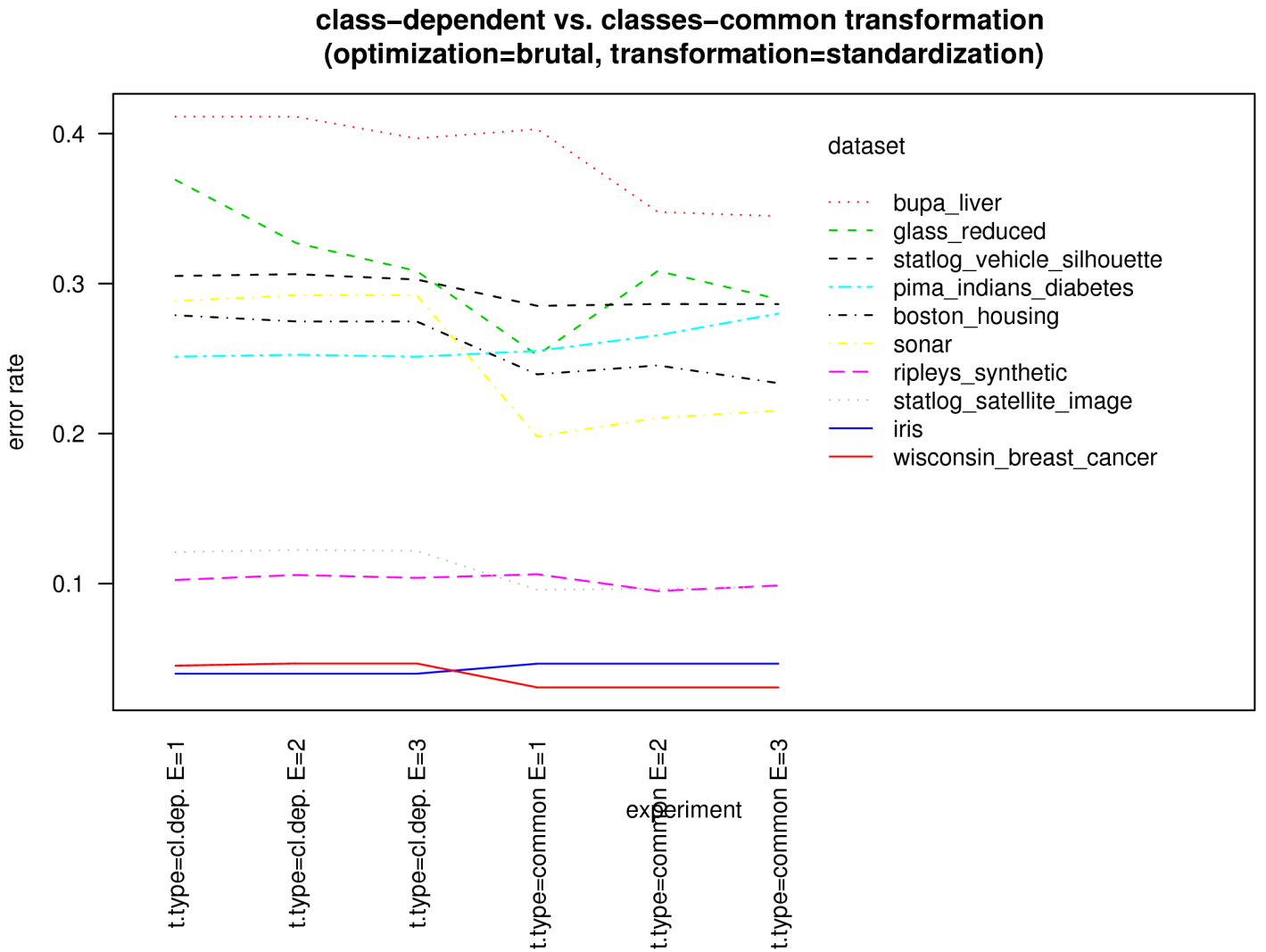


- Friedman test: p-value=0.8217
  - Brak istotnych różnic między algorytmami



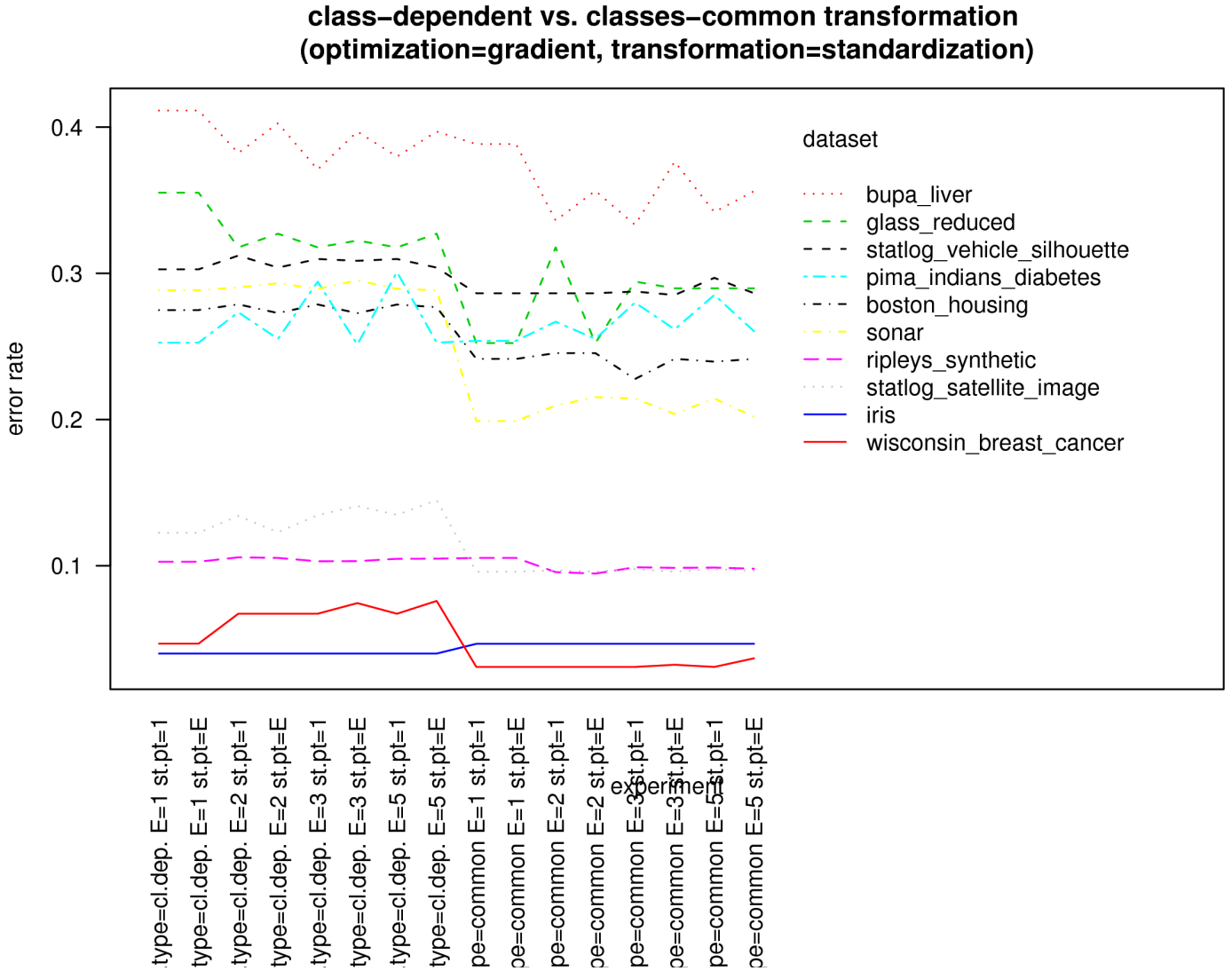
- Friedman test: p-value=0.9797
  - Brak istotnych różnic między algorytmami

Porównanie typów transformacji danych 1



- Friedman test: p-value=0.0433
  - Istnieją istotne różnice między algorytmami na poziomie istotności 0.05

Porównanie typów transformacji danych 2



- Friedman test: p-value=3.163e-05
  - Istnieją istotne różnice między algorytmami na poziomie istotności 0.01

Literatura

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

[Lim00] Tjen-Sien Lim, 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

[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!