

Klasyfikacja strumieni danych z wykorzystaniem złożonych systemów klasyfikacji

Dr inż. Konrad Jackowski
Politechnika Wrocławska
Wydział Elektroniki
Katedra Systemów i Sieci Komputerowych

Agenda

- Przetwarzanie strumieni
 - Problem zmienności danych
 - Przykłady
 - Źródła zmian
 - Dynamika
 - Algorytmy przetwarzania zmiennych strumieni danych
 - Typy
 - Algorytmy złożone
 - Algorytm ewolucyjnej adaptacji klasyfikatora
 - Podstawowe założenia
 - Model
 - Algorytm
 - Ewaluacja
-

Data Mining

- Data Mining – techniki odkrywanie informacji z baz danych
 - Proces odkrywania ma na celu wyłowienie nietrywialnych, wcześniej nieznanych i praktycznie użytecznych informacji, wzorców, relacji ukrytych w bazach i niemożliwych do wyłuskania klasycznymi metodami raportowania
 - Identyfikacja nietypowych wzorców - Anomaly Detection
 - Odkrywanie zależności pomiędzy zmiennymi - Association Rule Learning
 - Odkrywanie grup i struktur danych wykazujących podobieństwo - Clustering
 - Identyfikacja wzorców atrybutów wykorzystywana w zadaniach klasyfikacji – Classification
-

Strumienie danych

- Strumień danych – sekwencja danych często powiązanych relacjami, generowanych i napływających do systemu w czasie, w stałych lub zmiennych odstępach czasu
 - Problemy z obsługą strumieni dotyczą
 - Dużego zapotrzebowania na przestrzeń do magazynowania danych
 - Zwiększonej mocy obliczeniowej niezbędnej do przetwarzania i analizy danych
-

Strumienie danych

- Klasyczne algorytmy nie mogą być w sposób prosty wykorzystane do przetwarzania danych strumieniowych
 - Większość z nich wymaga wielokrotnego przeglądu danych archiwalnych
 - Dane mogą być usuwane ze względu na problemy związane z ich przechowywaniem
 - Charakterystyka danych może ulegać zmianom
-

Zmienność w strumieniach danych

Concept Drift

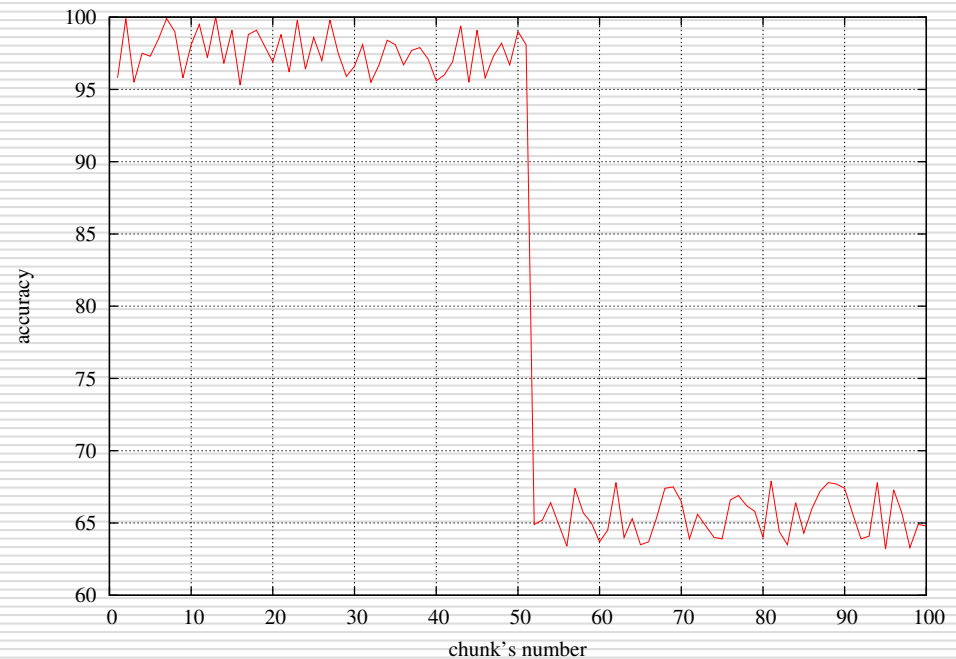
- ❑ Concept drift – to zjawisko pojawiania się zmian w charakterystyce danych strumieniowych
 - ❑ Przykład:
 - W systemach detekcji i filtracji SPAMu bazuje się na analizie treści. Z upływem czasu zmiany mogą ulec:
 - ❑ percepcja tego co uznawane jest za SPAM
 - ❑ sposoby formułowania wiadomości
-

Concept Drift

- Ewolucja metod fałszerstw finansowych
 - Zmiana preferencji klientów wynikająca z:
 - Pojawienia się nowej oferty
 - Sezonowości popularności wybranych produktów
 - Zmiany metod ataków na systemy informatyczne
 - Zmiana profilu klienta w systemach rekomendowania usług bankowych
 - Zmiana trendów wybranych wskaźników makroekonomicznych
-

Concept Drift

- Pojawienie się zmian charakterystyk danych powoduje, że klasyczne (statyczne) algorytmy analizy danych mogą z czasem stać się bezużyteczne
- W przypadku zadań klasyfikacji efektem może być drastyczne obniżenie poziomu poprawnych rozpoznań



Concept Drift

- Nowe zadania
 - Monitorowanie charakterystyk danych pod kątem pojawiania się w nich zmian –
Detekcja Dryftu
 - Dostrojenie systemu do nowych warunków
 - Podstawą przy projektowaniu systemów jest zrozumienie natury pojawiających się zmian
-

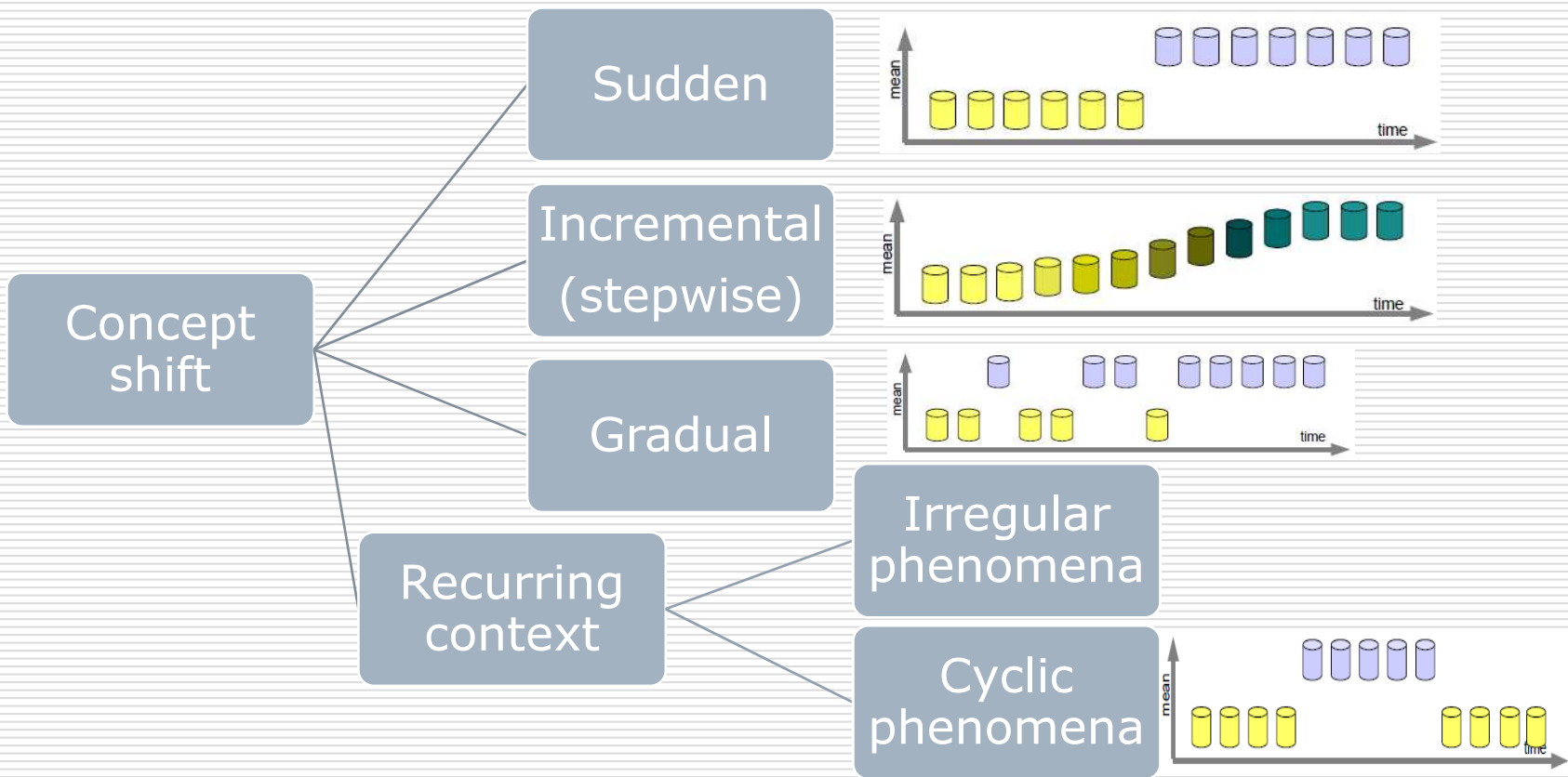
Concept Drift

Źródła zmian charakterystyk

- W zadaniach klasyfikacji zakłada się probabilistyczny model opisujących charakterystykę obiektów poddawanych analizie
 - Zmiany mogą ujawnić się w jednym lub wielu z następujących parametrów opisujących obiekty
 - Prawdopodobieństwo a-priori pojawiania się obiektów w klasach
 - Warunkowy (klasowy) rozkład atrybutów obiektów w przestrzeni cech
 - Wartość prawdopodobieństwa a-posteriori przynależności obiektu do danej klasy
-

Concept Drift

Dynamika zmian



Concept Drift

Dynamika zmian

- Możliwa reakcja na pojawienie się zmian zależy od ich dynamiki, np.:
 - W przypadku gwałtownej zmiany charakterystyk (sudden drift), zastąpienie zdezaktualizowanego klasyfikatora nowym pozwoli na błyskawiczne dostrojenie systemu do nowych warunków
 - W przypadku zmian rozciągniętych w czasie bardziej odpowiednia wydaje się stopniowa adaptacja klasyfikatora
 - W przypadku powtarzalności konceptów idealnie byłoby mieć możliwość ponownego wykorzystywania klasyfikatorów stworzonych w przeszłości
-

Concept Drift - Algorytmy

- Historia
 - STAGGER (Schlimmer i Granger, 1986)
 - IB3 (Aha, 1991)
 - Algorytmy FLORA (Widmer i Kubat, 1996)
 - Klasyfikacja algorytmów
 - Algorytmy detekcji dryftu
 - Online Learners
 - Algorytmy z mechanizmem „zapominania”
 - Ensemble classifiers
-

Algorytmy detekcji dryftu

- Detekcja dryftu jest podstawą podjęcia reakcji na pojawienie się zmian
 - Budowa nowego klasyfikatora
 - Adaptacja systemu do nowych warunków
 - Żądanie etykietowania zbioru danych
 - Nie wszystkie systemy przetwarzania danych wymagają detekcji dryftu
 - Niektóre systemy w sposób ciągły adaptują się do nadchodzących danych (Zliobaite:2010)
-

Algorytmy detekcji dryftu

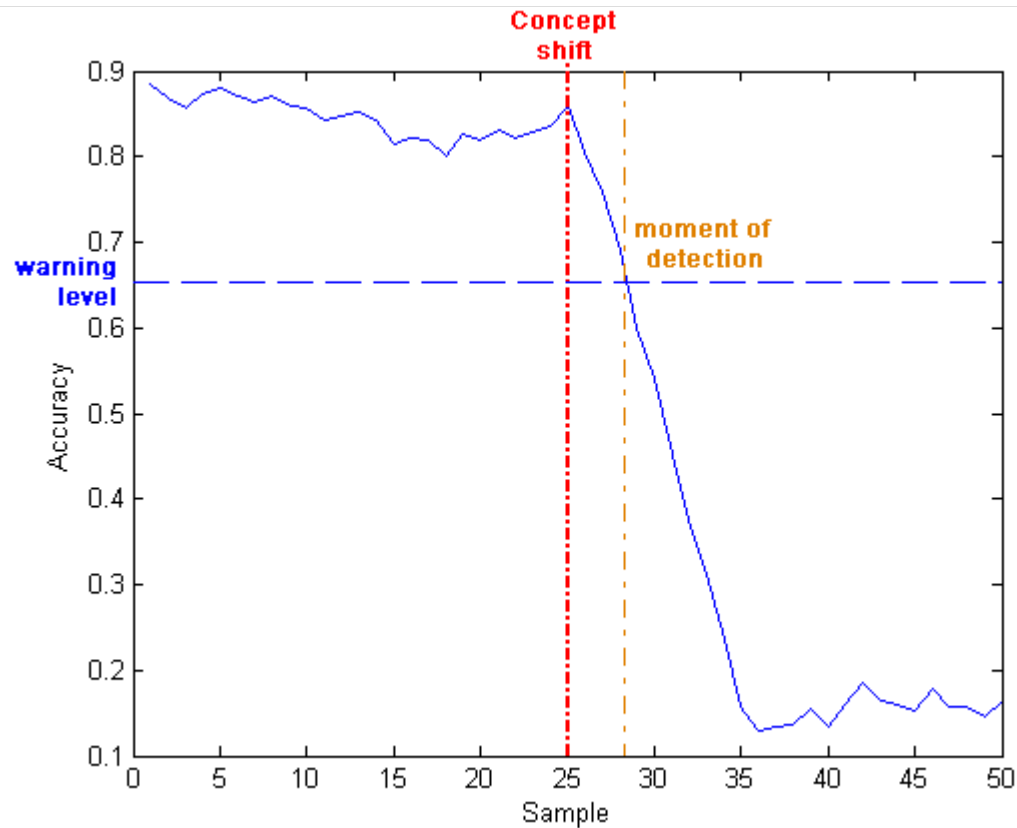
Możliwe podejścia

- Algorytmy nadzorowane
 - Dostępność informacji o klasach – udział eksperta
 - Monitorowanie zmian w jakości klasyfikacji
 - Monitorowanie warunkowych rozkładów obiektów w przestrzeni cech

 - + duża precyzja detekcji
 - - konieczność zaangażowanie eksperta
 - - w wielu praktycznych przypadkach nie ma dostępu do etykiet (np. z powodu szybkości przepływu danych w strumieniu) (Kifer:2004)
-

Algorytmy detekcji dryftu

Możliwe podejścia



Algorytmy detekcji dryftu

Możliwe podejścia

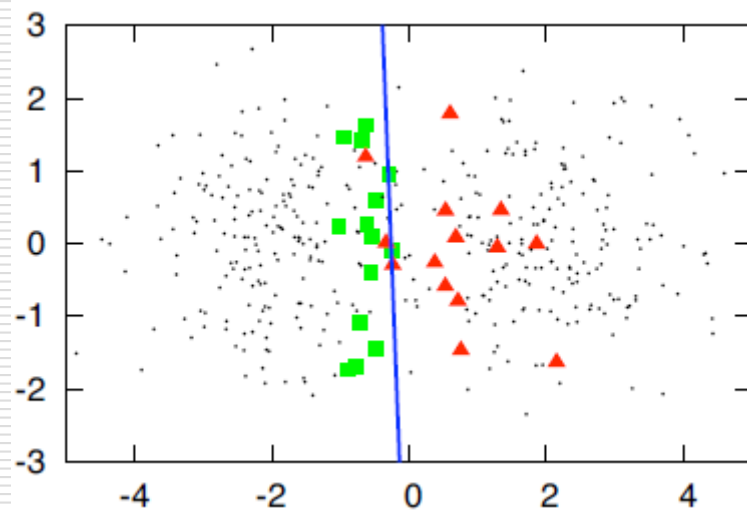
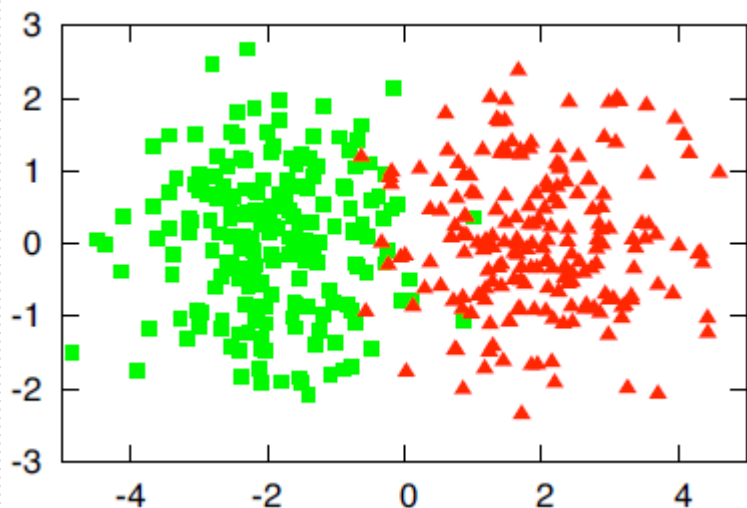
- Algorytmy częściowo nadzorowane (semisupervised)
 - Zakłada się ograniczony dostęp do wiedzy eksperta
 - Tylko część danych jest etykietowana
 - Uwzględnienie kosztów etykietowania w wyborze przykładów do oznaczenia
 - Algorytmy aktywnego uczenia

 - + duża precyzja detekcji
 - +/- ograniczone koszty związane z etykietowaniem
-

Algorytmy detekcji dryftu

Możliwe podejścia

- Detektor dryftu wykorzystujący algorytmy aktywnego uczenia (Kurlej, Woźniak, 2011)



Algorytmy detekcji dryftu

Możliwe podejścia

- Algorytmy nienadzorowane
 - Brak etykiet
 - Brak miar jakości klasyfikacji
 - Monitorowanie właściwości obiektów w strumieniu
 - Rozkłady obiektów w przestrzeni cech
 - Rozkład klastrów
 - Testy statystyczne
 - + brak kosztów etykietowania
 - + natychmiastowe przetwarzanie danych
 - - konieczność przyjęcia założeń odnośnie rozkładu obiektów w przestrzeni cech
 - - problemy z detekcją zmian rozkładów warunkowych w klasach
-

Online algorithms

- Algorytmy przetwarzające przychodzące dane w czasie rzeczywistym.
 - Wymagania (Domingos:2003)
 - Przetwarzanie danych w chwili ich pojawienia się - brak konieczności gromadzenia danych
 - Jednorazowe przetwarzanie każdego przychodzącego obiektu w procesie uczenia – brak konieczności ponownego przeglądania danych
 - Przetwarzanie danych w ograniczonym czasie i z wykorzystaniem ograniczonej ilości zasobów
 - W przypadku przerwania procesu przetwarzania danych w dowolnym momencie, jakość klasyfikatora powinna odpowiadać jakości klasyfikatora uczonego wsadowo na podstawie wszystkich dotychczas pojawiających się przykładów
-

Online algorithms - przykłady

- Zaadaptowane klasyczne algorytmy klasyfikacji
 - Naive Bayes
 - Neural Networks
 - Nearest Neighbour
 - Algorytmy złożone
 - Concept-adapting Very Fast Decision Tree - CVFDT (Hulten, 2001)
 - Generowanie alternatywnych drzew lub gałęzi drzewa decyzyjnego na podstawie nowych danych
 - Dynamiczne zastępowanie zdezaktualizowanych drzew ich nowymi wersjami w przypadku gdy osiągają lepsze wyniki
-

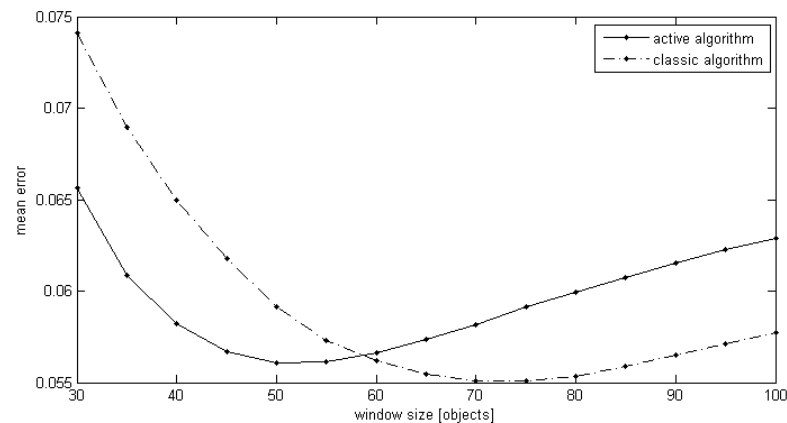
Sliding Window

- ❑ Algorytmy wykorzystujące w procesie uczenia zbiór uczący zawierający jedynie pewną pulę ostatnich (najnowszych) przykładów – „okno”
 - ❑ Algorytmy bazujące na założeniu, że „najnowsze” przykłady w najlepszy sposób opisują bieżący *kontekst*
 - ❑ Rozmiar okna wpływa na dynamikę zachowania klasyfikatora
-

Sliding Window

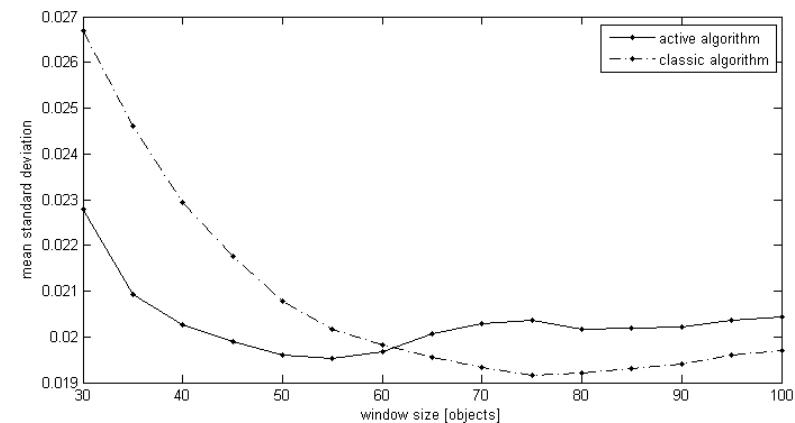
❑ Małe okno

- ❑ duża dynamika (szybka reakcja na zmiany)
- ❑ Większa podatność na zakłócenia
- ❑ Niższa stabilność procesu



❑ Duże okno

- ❑ Wyższa inercja (wolniejsze dostosowywanie się do nowych warunków)
- ❑ Większa odporność na szumy
- ❑ Wyższa stabilność

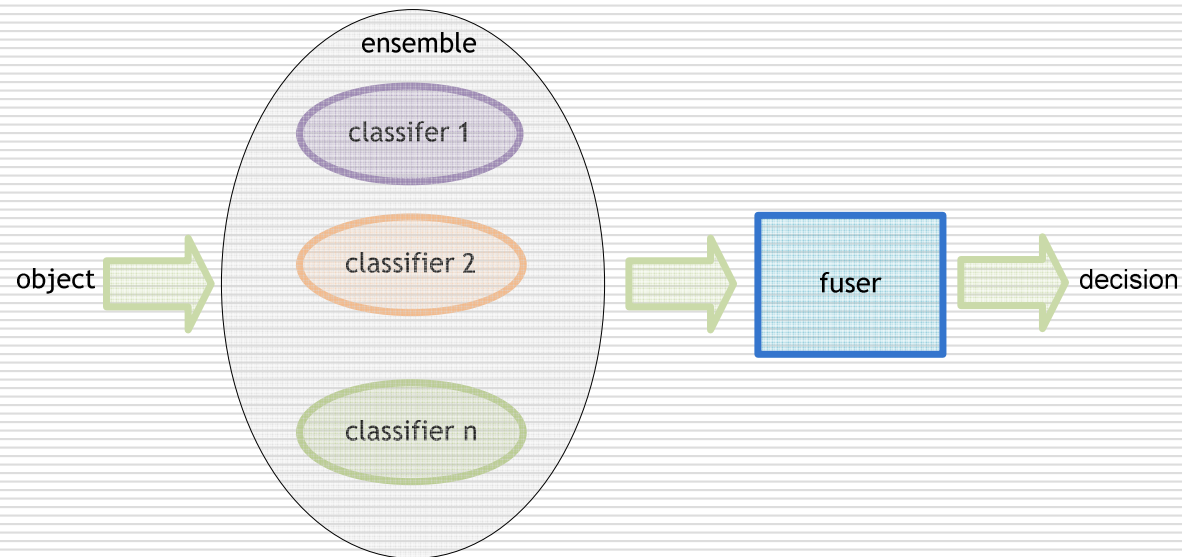


Sliding Window

- Zaawansowane algorytmy „okna”
 - Algorytmy o zmiennej szerokości okna
 - FLORA2 (Widmer,1996)
 - ADWIN2 (Bifet,2006)
 - Algorytmy z wieloma oknami (Lazarescu, 2003)
 - Algorytmy ważone (Klinkenberg,1998),(Koychev,2000)
 - Wszystkie obiekty w oknie otrzymują wagi wprost proporcjonalne do ich „wieku”
 - Algorytmy wykorzystujące bagging i boosting (Bifet,2009),(Chu,2004)
-

Ensembles

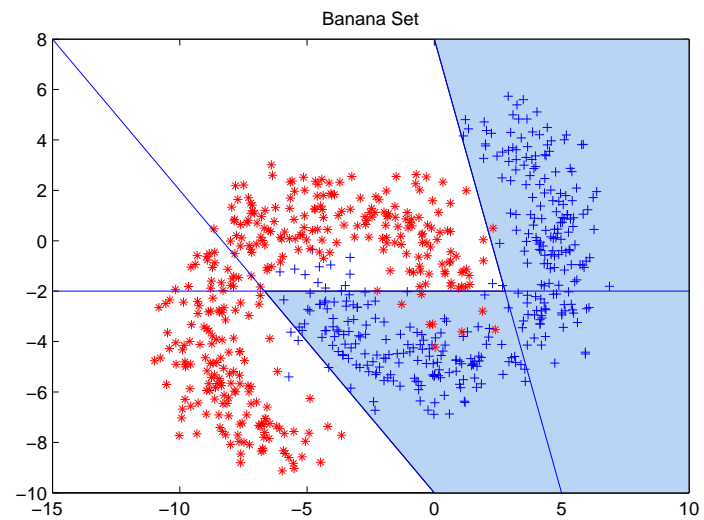
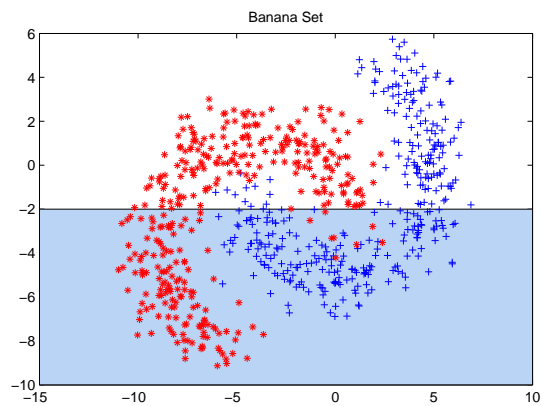
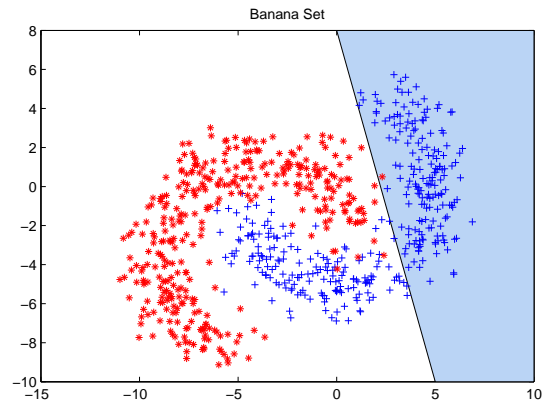
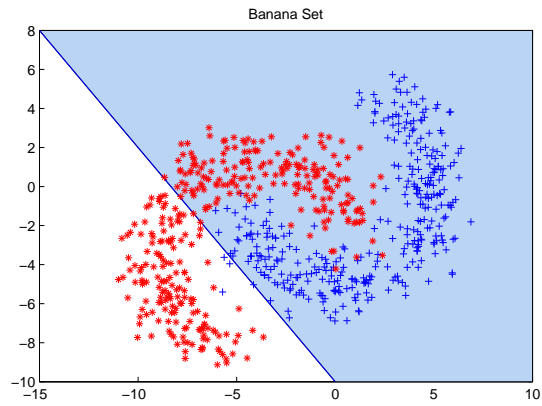
- ❑ Algorytmy złożone (Ensemble classifiers) – algorytmy wykorzystujące pulę klasyfikatorów elementarnych
- ❑ Fuzja wiedzy podchodzących ze zbioru klasyfikatorów pozwala na istotne podniesienie jakości klasyfikacji



Ensembles

- Warunek sukcesu fuzji klasyfikatorów
 - Wiedza zawarta w klasyfikatorach musi się uzupełniać
 - Klasyfikatory powinny być zróżnicowane
-

Ensembles



Ensembles

- Warunek sukcesu fuzji klasyfikatorów
 - Wiedza zawarta w klasyfikatorach musi się uzupełniać
 - Klasyfikatory powinny być zróżnicowane
 - W warunkach statycznych zróżnicowanie puli może wynikać z
 - Różnych modeli klasyfikatorów
 - Losowości wynikającej z modelu klasyfikatora
 - Wykorzystania różnych podzbiorów cech
 - Podziału zbioru uczącego
 - W warunkach dryftu zróżnicowanie może dodatkowo wynikać z reprezentacji różnych konceptów
-

Model fuzji

Fuzja odpowiedzi

- Fuzja odpowiedzi – etykiet zwracanych przez klasyfikatory elementarne –
 - Głosowanie większościowe
 - Głosowanie ważone
 - + prostota implementacji
 - + możliwość fuzji heterogenicznego zbioru klasyfikatorów
 - + intuicyjność metody
 - - zignorowanie informacji o „zaufaniu” wobec decyzji poszczególnych klasyfikatorów
-

Model fuzji

Fuzja funkcji dyskryminujących

- Fuzja funkcji dyskryminujących
 - Estymatory prawdopodobieństw a-posteriori
 - Funkcje przejścia w sieciach neuronowych

 - Funkcje dyskryminujące jako miara wsparcia/pewności odnośnie podejmowanej decyzji
 - Klasyfikatory zwracają wartości funkcji dla każdej z możliwych klas
-

Model fuzji

Fuzja funkcji dyskryminujących

- Proste algorytmy agregujące:
 - min,
 - max,
 - avg,
 - prod,
 - ...
 - + prostota implementacji
 - + brak fazy uczenia klasyfikatora złożonego
 - - ograniczone pole zastosowań
 - - relatywnie niska skuteczność
-

Ważona fuzja funkcji dyskryminujących

- Ważona fuzja funkcji dyskryminujących
 - Wagi zależne od klasyfikatorów
np. wagi wprost proporcjonalne do jakości klasyfikacji klasyfikatora elementarnego
 - Wagi zależne od klasyfikatorów i wartości wektora cech
 - Wagi zależne od klasyfikatorów i klasy
 - Wagi zależne od klasyfikatorów, wektora cech i klasy
-

Ważona fuzja funkcji dyskryminujących

- Ważona fuzja funkcji dyskryminujących
 - Wagi zależne od klasyfikatorów
np. wagi wprost proporcjonalne do jakości klasyfikacji klasyfikatora elementarnego
 - Wagi zależne od klasyfikatorów i wartości wektora cech
 - + Uzależnienie wpływu klasyfikatora na efekt klasyfikacji w zależności od jego jakości
 - + Efektywne przeciwdziałanie negatywnemu wpływowi przetrenowania klasyfikatorów elementarnych
 - - Pracochłonna implementacja algorytmu uczenia
-

Concept Drift

Ensembles - Strategie

- Dynamic combiners –
 - Fuzja klasyfikatorów trenowanych w procesie poprzedzającym budowę klasyfikatora złożonego (Littlestone:1994),(Jacobs:1991)
 - Wadą rozwiązania jest konieczność znajomości wszystkich kontekstów w fazie tworzenia puli
 - Inkrementacyjna aktualizacja klasyfikatorów elementarnych w puli (Oza:2000), (Kolter:2007), (Bifet:2011), (Bifet:2009), (Rodriguez:2008)
 - System jest złożony z klasyfikatorów onlinowych
 - Podejście to bywa nazywane Online Ensemble
 - Aktualizacja struktury systemu
 - Wraz z nadejściem nowej porcji danych wszystkie składowe systemu są za ich pomocą oceniane
 - Najsłabszy klasyfikator może być zastąpiony nowym, trenowanym na bieżących danych (Kolter:2003), (Jackowski:2013)
-

Concept Drift

Ensembles – Przykładowe algorytmy

- Streaming Ensemble Algorithm - SEA (Street:2001)
 - Accuracy Weighted Ensemble – AWE (Wang:2003)
 - Podstawowe założenia
 - Pula klasyfikatorów o stałym rozmiarze
 - Grupowanie nadchodzących danych w blokach
 - Przetwarzanie wsadowe
 - Ocena klasyfikatorów w puli na podstawie dokładności klasyfikacji przykładów z bloku danych
 - Odświeżanie puli – zastępowanie klasyfikatorów nowymi
 - SEA - fuzja z wykorzystaniem głosowania większościowego
 - AWE – fuzja z wykorzystaniem głosowania ważonego, gdzie waga odpowiada jakości klasyfikatora w klasyfikacji aktualnych danych
-

AWE - pseudokod

AWE algorithm

1. Input:
 2. S: Data Chank
 3. K: Size of comittee
 4. C: classifier pool
 5. Output:
 6. C
 7. Begin
 8. train C0 on S;
 9. compute error rate of C0 via cross validation on S;
 10. calculate w0 for C0 using MSEi;
 11. for *each classifier* Ci in C do
 12. apply Ci on S to derive MSEi
 13. calculate wi based on MSEi;
 14. C ← K of the top weighted classifiers in $C \cup \{C0\}$;
 15. return C;
 16. End
-

Dyskusja

- ❑ Algorytmy oparte na zapominaniu mają ograniczone zastosowanie w przypadku możliwości ponownego pojawienia się konceptu
 - ❑ Rozwiązania z pulą klasyfikatorów pozwalają na reprezentację kontekstów w postaci klasyfikatorów elementarnych
 - ❑ Możliwość ponownego wystąpienia kontekstu wymusza rezygnację z usuwania „starych” klasyfikatorów z puli
 - ❑ Komitet głoszący nie powinien zawierać nieaktualnych klasyfikatorów
 - ❑ Ważenie wkładu klasyfikatorów w podejmowanie decyzji pozwala na uwzględnienie stopnia ich aktualności
 - ❑ Ustalenie wag na podstawie jakości klasyfikatora jest intuicyjne i proste w implementacji, ale algorytmy takie mają tendencję do wpadania w minima lokalne
 - ❑ Alternatywą jest dobór wag na podstawie tak by zoptymalizować jakość pracy całego systemu
-

Dyskusja

- Zachowanie wiedzy – raz zdobyta wiedza może okazać się przydatna w przypadku ponownego pojawienia się kontekstu
 - Ewolucyjna adaptacja – stopniowe dostrajanie systemu pozwala na uniknięcie niestabilności wynikających z naturalnych fluktuacji danych
 - Ważona fuzja funkcji dyskryminujących
 - Uwzględnienie aktualności wiedzy klasyfikatorów
 - Uwzględnienie stopnia pewności decyzji klasyfikatorów
 - Ograniczenie liczebności komitetu głosującego
 - Ograniczenie czasu obliczeń
 - Ograniczenie wpływu słabych/nieaktualnych klasyfikatorów
 - Pewna detekcja dryftu
 - Rozróżnienie dryftu od naturalnych fluktuacji strumienia
-

Założenia systemu EAE

Evolutionary Adapted Ensemble

- ❑ System przechowuje wiedzę odnośnie kontekstów w postaci puli dostępnych klasyfikatorów elementarnych
- ❑ Rozmiar puli nie jest ograniczony
- ❑ Składowe puli nie są z niej usuwane
- ❑ Decyzja systemu jest podejmowana kolektywnie
- ❑ Rozmiar komitetu jest stały i ograniczony
- ❑ Decyzja jest podejmowana na podstawie ważonej fuzji funkcji dyskryminujących
- ❑ System w sposób ciągły nadzoruje i aktualizuje skład komitetu oraz wagi poszczególnych klasyfikatorów
- ❑ Nowe klasyfikatory są tworzone w przypadku obniżenia jakości klasyfikacji
- ❑ Algorytm uczenia to złożony problem optymalizacyjny mający na celu minimalizację błędu klasyfikacji systemu
- ❑ W procesie uczenia wykorzystano algorytm ewolucyjny

EAE Model

□ Pula klasyfikatorów

$$\Pi^\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_K\}$$

□ Komitet głosujący

$$\Xi^\Psi = \{c_1, c_2, \dots, c_E\}$$

□ Wagi klasyfikatorów

$$W^\Psi = \{w_1, w_2, \dots, w_E\}$$

□ Formuła decyzyjna

$$\bar{\Psi}(x) = \arg \max_{i=1}^M \sum_{e=1}^E w_e d_{e,i}(x)$$

□ Zbiór uczący

$$LS = \{(x_1, j_1), (x_2, j_2), \dots, (x_N, j_N)\}$$

□ Funkcja celu procesu uczenia

$$Q^{\bar{\Psi}} = \frac{1}{N} \sum_{n=1}^N L(\bar{\Psi}(x_n), j_n) = \frac{1}{N} \sum_{n=1}^N L\left(\arg \max_{i=1}^M \sum_{e=1}^E w_e d_{c_e,i}(x), j_n\right)$$

EAE

Algorytm optymalizacyjny

- ❑ Uczenie z wykorzystaniem algorytmów ewolucyjnych
- ❑ Model rozwiązania zakodowany w postaci chromosomu

$$Ch = \begin{cases} W^\Psi \\ \Xi^\Psi \end{cases} = \begin{cases} [w_1, w_2, \dots, w_E] \\ [c_1, c_2, \dots, c_E] \end{cases}$$

- ❑ Przetwarzanie populacji osobników reprezentujących możliwe rozwiązania
 - ❑ Zachowanie całej populacji po ukończeniu procesu uczenia
 - ❑ Najlepszy osobnik z populacji jest rozwiązaniem na danym etapie przetwarzania strumienia
 - ❑ Proces jest kontynuowany po odebraniu kolejnego zestawu danych ze strumienia
-

EAE

Inicjalizacja populacji

- Populacja reprezentuje możliwe rozwiązania
 - Sposób inicjalizacji populacji zależy od etapu przetwarzania strumienia danych
 - W przypadku otrzymania pierwszego pakietu danych (początek strumienia) chromosomu osobników populacji są wypełniane wartościami losowymi
 - W przypadku przetwarzania kolejnych pakietów danych, populacja jest wypełniona osobnikami zachowanymi z ostatniego przebiegu algorytmu (uczenie poprzedniego pakietu danych)
 - W przypadku, gdy komitet głosujący nie jest wypełniony, tworzy się nowy klasyfikator elementarny i dodaje go do komitetu z losową wagą
-

EAE

Inicjalizacja populacji

```
1. Function Init Population
2. Input:
3.     Previous population,
4.     Pool Learning Set
5.     Pool
6.     Ensemble Size
7.     Population Size
8. Output:
9.     Population
10. Begin
11.     If (Previous population is empty)
12.         Create Population
13.     Else
14.         Population = Previous population
15.     EndIf
16.     If (size of Pool < Ensemble Size)
17.         Create New Classifier
18.         Train Classifier with Pool Learning Set
19.         Add Classifier to Pool
20.         For each Individual in Population
21.             Increment size of  $\Xi$  and  $W$ ;
22.             Set Individual. $\Xi$ (last) to New Classifier
23.             Set Individual. $W$ (last) to random number
24.         EndFor
25.     EndIf
26. End
```

EAE

Funkcja dopasowania

- Podstawą oceny osobników w populacji jest funkcja dopasowania
- W przypadku EAE osobniki reprezentujące możliwe konfiguracje złożonego klasyfikatora są oceniane za pomocą funkcji odwrotnie proporcjonalnej do funkcji błędu

$$F = 1 - Q^{\bar{\Psi}} = 1 - \frac{1}{N} \sum_{n=1}^N L \left(\arg \max_{i=1}^M \sum_{e=1}^E w_e d_{C_{e,i}}(x), j_n \right)$$

- Wartość dopasowania osobnika jest kluczowa w procesie selekcji osobników do procesów krzyżowania oraz tworzenia populacji potomnej
 - Osobniki o wysokim dopasowaniu mają większe szanse na przekazanie swoich genów osobnikom potomnym w procesie krzyżowania
-

EAE

Krzyżowanie

- ❑ W fazie przetwarzania populacji następuje wymiana danych zapisanych w chromosomach dwóch losowo wybranych osobników
 - ❑ Celem krzyżowania jest propagowanie cech „silnych” osobników w populacji potomnej
 - ❑ W EAE wykorzystano standardowy operator krzyżowania dwupunktowego
 - ❑ Krzyżowanie dotyczy tych samych składowych chromosomu (wagi i indeksy) w obydwu osobnikach
 - ❑ Dopuszcza się występowanie jednego klasyfikatora w komitecie wielokrotnie
 - Jeżeli redundancja wpłynie negatywnie na jakość rozwiązania, osobnik taki zostanie wyeliminowany w procesie selekcji
-

EAE

Mutacja

- Mutacja to proces wprowadzania pewnych losowych zmian w kod chromosomu
 - Mutacja zazwyczaj dotyczy znikomej części populacji
 - Ma na celu podtrzymanie pewnego poziomu zróżnicowania populacji
 - Mutacja wag
 - Dodanie szumu Gausowskiego
 - Normalizacja wektora wag
 - Mutacja indeksów składu komitetu
 - Wymiana losowo wybranego indeksu na inny z puli dostępnych
 - Prawdopodobieństwo wylosowania klasyfikatora jest wprost proporcjonalne do jego jakości
-

EAE

Mutacja

```
1. Function Mutation
2. Input:
3.   Parents
4.   Pool
5. Output:
6.   Children
7. Begin
8.   For each Individual in Parents
9.     Child = Individual
10.    Select randomly index in  $\Xi$ 
11.    Select new Classifier from Pool
        With Roulette Selection
12.    Set Child. $\Xi$ (index) to new classifier
13.    Set Child. $W$ (index) to random number
14.    Normalize Child. $W$ 
15.   EndFor
16. End
```

EAE

Detekcja przetrenowania

- ❑ Duża elastyczność modelu klasyfikatora EAE sprzyja możliwości jego przetrenowania
 - ❑ Utrata zdolności generalizacji w naturalny sposób obniża jakość jego pracy
 - ❑ Podstawą działania jest zbiór walidacyjny – nie wykorzystywany w procesie uczenia
 - ❑ Detekcja przetrenowania ma miejsce w czasie każdej iteracji procesu ewolucyjnego
 - ❑ Detektor zapamiętuje ostatnią z populacji, w której nie wykryto przetrenowania
 - ❑ Po zakończeniu procesu uczenia, populacja ta jest zwracana jako wynik działania algorytmu uczenia EAE
-

EAE

Detekcja przetrenowania

```
1. Function Detect Overtraining
2. Input:
3.   Population
4.   Not over-fitted population
5.   Last Result
6. Output:
7.   Not over-fitted population
8.   Last Result
9. Begin
10.  Get Best from Population
11.  Set Result to Evaluate on Validating Set
12.  If (Result < Last Result)
13.    Last Result = Result
14.    Set Not over-fitted population = Population
15.  EndIf
16. End
```

EAE

Detekcja dryftu konceptu

- ❑ Procedura detekcji dryftu ma za zadanie rozpoznanie momentu pojawienia się zmian
 - ❑ Wykorzystano detektor nadzorowany
 - ❑ Podstawą oceny jest monitorowanie wartości jakości klasyfikacji
 - ❑ Detektor ma za zadanie odróżnić odchylenia w jakości wynikających z chwilowych fluktuacji od pojawienia się prawdziwego dryftu
 - ❑ Zdefiniowano dwa poziomy alarmowe odchylenia
 - Poziom dryftu
 - Poziom szumu
 - ❑ Pojawienie się szumu może spowodować utworzenie nowego klasyfikatora
 - ❑ Pojawienie się dryftu powoduje wymianę całego komitetu głosującego
-

EAE

Detekcja dryftu konceptu

```
1. Function Detect drift
2. Input:
3.   Population
4.   Pool Learning Set
5.   Ensemble Training Set
6.   Ensemble Size
7.   Last Result
8.   Drift Threshold
9.   Noise Threshold
10.  Update Counter
11. Output:
12.  Last Result
13.  Max Generation
14. Begin
15.  Get Best from Population
16.  Set Result to Evaluate on Ensemble Training Set
17.  If (Result > Last Result + Noise Threshold)
18.    Update Counter = 1
19.  EndIf
20.  If (Result > Last Result + Drift Threshold)
21.    Update Counter = Ensemble Size
22.  EndIf
23.  If (Update Counter > 0)
24.    Create Classifier
25.    Train Classifier with Pool Learning Set
26.    Add Classifier to Pool
27.    Set Max Generation to 2 * Max Generation
28.    Decrement Update Counter
29.  else
30.    Last Result = Result
31.  EndIf
32. End
```

EAE

Główna procedura algorytmu

- Algorytm jest sterowany zestawem parametrów
 - Rozmiar populacji
 - Frakcja krzyżowania/mutacji
 - Rozmiar elity
 - Warunki zakończenia uczenia
 - Procedura jest uruchamiana po skompletowaniu paczki danych o zadanym rozmiarze
 - Podział zbioru na:
 - Uczący dla klasyfikatorów elementarnych
 - Uczący algorytmu EAE
 - Walidacyjny
-

EAE

Główna procedura algorytmu

```
1. Program EAE
2. Input
3.   Data Chunk
4.   Previous Population
5. Output
6.   Winner
7.   Population
8. Begin
9.   Split Data Chunk into
       Pool Training Set
       Ensemble Training Set
       Validating Set
10.  Init population
11.  Repeat Generation until Max Generation
12.    Evaluate population on Ensemble Training Set;
13.    Select Elite
14.    Select Parents for mutation and crossover
15.    Mutate
16.    Crossover
17.    Create child population
18.    Detect Overtraining
19.    If (last Generation)
20.      Detect Drift
21.    EndIf
22.  EndRepeat
23.  Set Winner to Best from Population;
24. End
```

EAE

Ewaluacja systemu

- Cele oceny
 - Ocena zdolności adaptacyjnych
 - Ocena odporności na przetrenowanie
 - Ocena wpływu rozmiaru puli na czas przetwarzania
 - Analiza porównawcza z innymi metodami
 - Środowisko
 - Matlab
 - PRTools
-

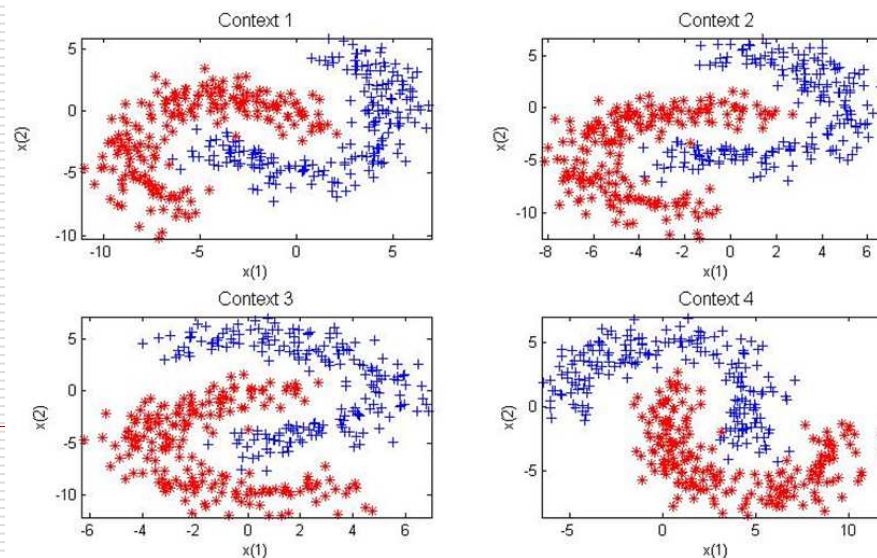
EAE

Ewaluacja systemu

□ Zbiory danych

■ Zbiory ze sztucznie generowanym dryftem

	Contexts	Drift extent (rotation angle)	Chunks per context	Ensemble size	Chunk size (instances)
Series 1	2	1.57	5	5	300
Series 2	4	0.78			
Series 3	6	0.52			



EAE

Ewaluacja systemu

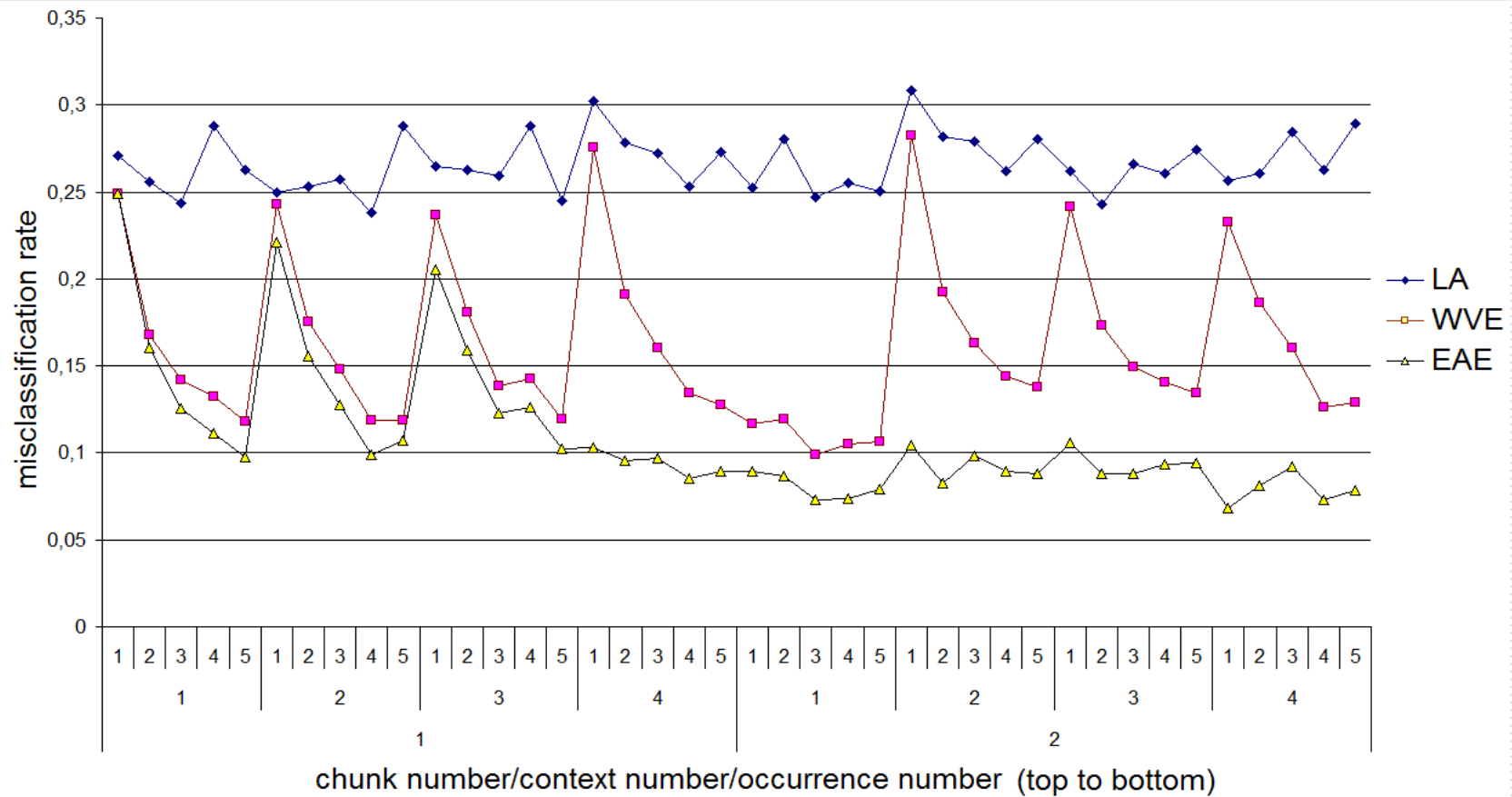
□ Zbiory danych

- Zbiory ze sztucznie generowanym dryftem
- Zbiory benchmarkowe (repozytorium UCI)

Name	Liver Disorders	Auto MPG	Pima Indians Diabetes	Breast Cancer	Biomed	Electricity demand	SPAM
Instances	345	398	768	699	699	45312	699
Classes	2	2	2	2	2	2	2
Attributes	6	6	8	9	5	6	9 324
Concept Drift	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Real	Real

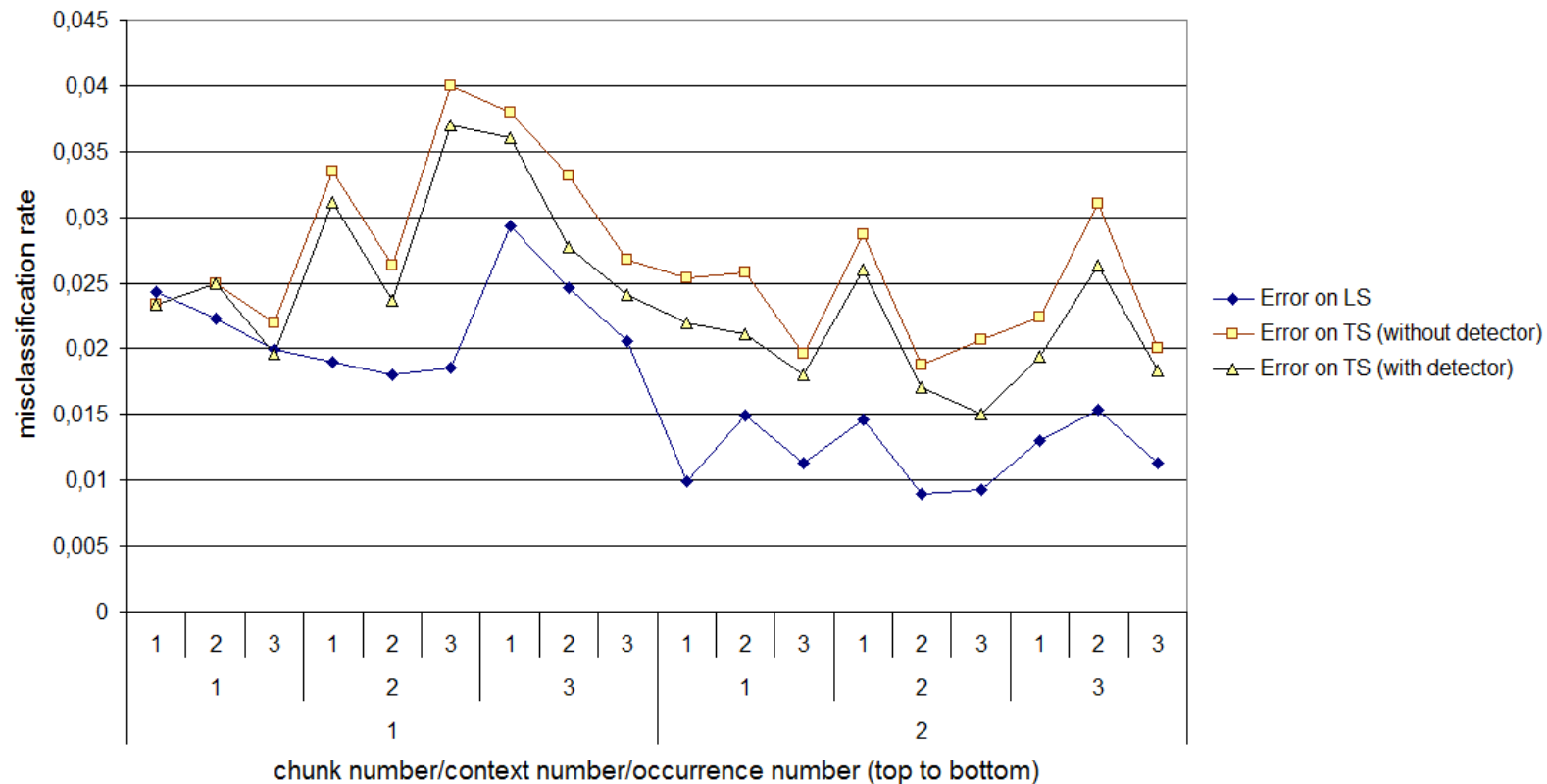
Ewaluacja systemu

Stabilność i czułość



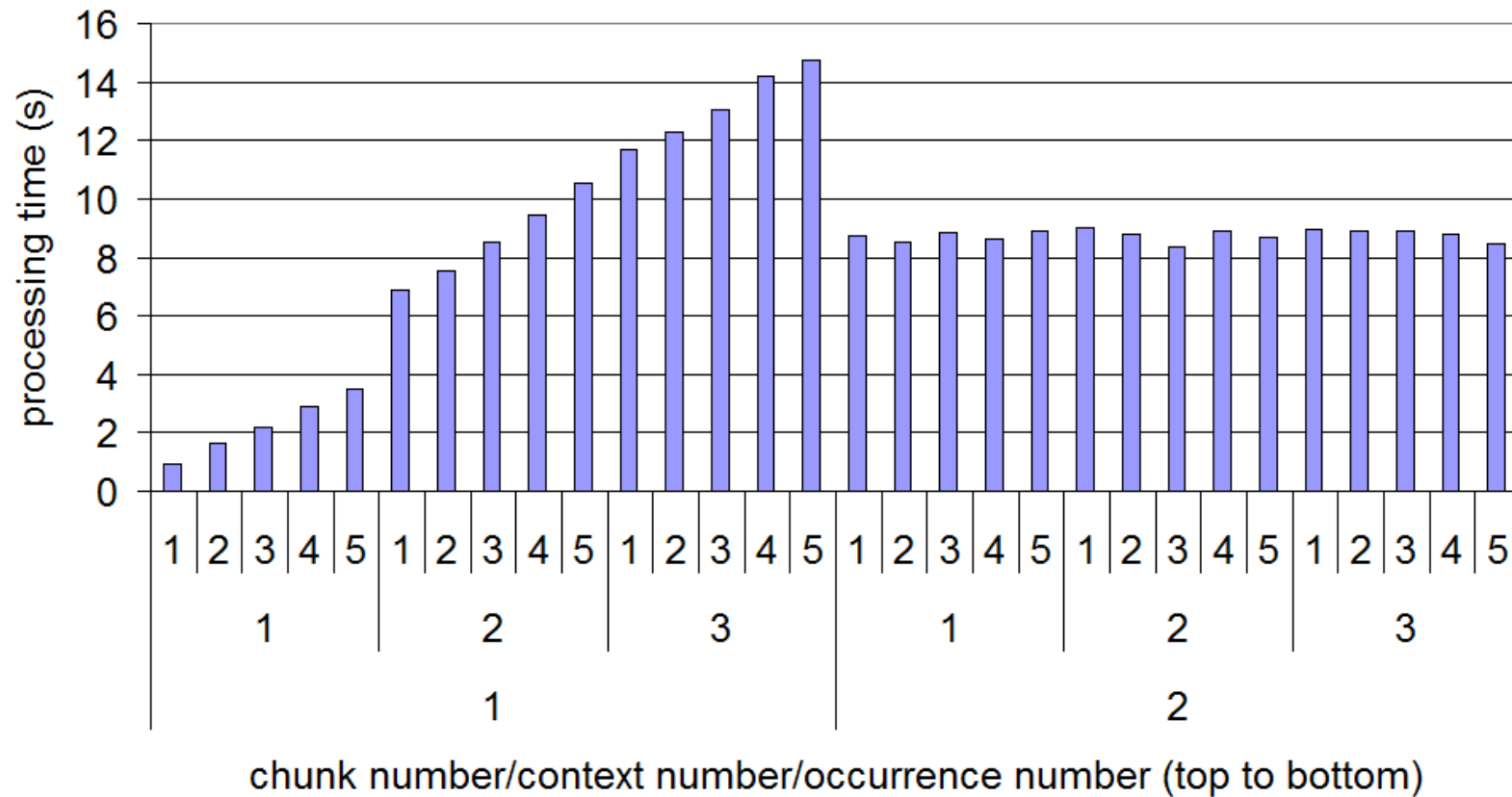
Ewaluacja systemu

Odporność na przetrenowanie



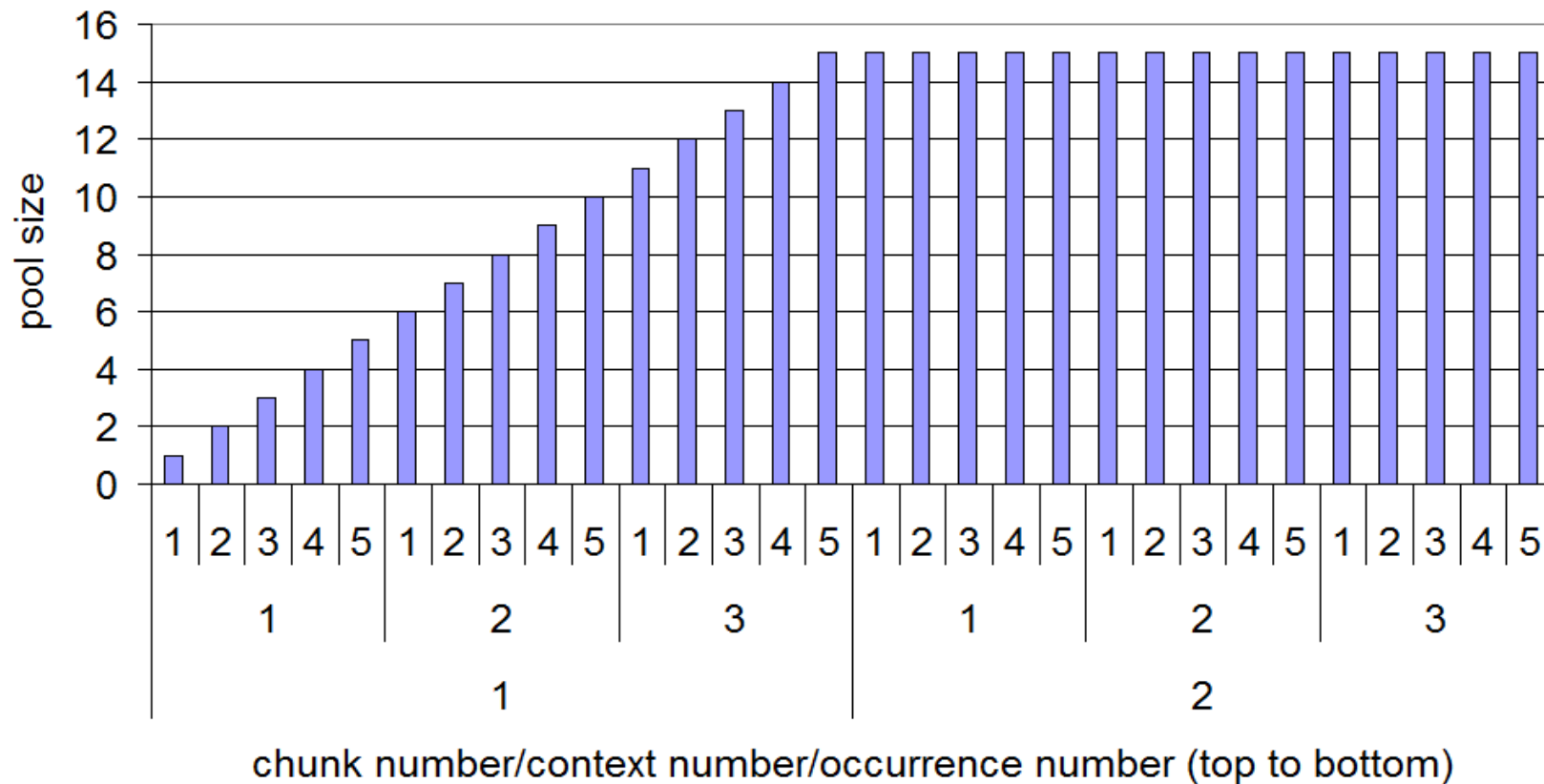
Ewaluacja systemu

Czas procesu uczenia



Ewaluacja systemu

Czas procesu uczenia



Ewaluacja systemu

Testy benchmarkowe

Dataset		AWE	AHT	LA	EAE
Electricity demand (90 chunks)	Average Error	0,2673	0,2489	0,2624	0,2342
	Error Variance	0,0007	0,0009	0,0009	0,0060
SPAM (23 chunks)	Average Error	0,1967	0,2006	0,1783	0,1002
	Error Variance	0,0031	0,0052	0,0030	0,0154

Ewaluacja systemu

Propozycje dalszych prac

- ❑ Wykorzystanie innych algorytmów optymalizacyjnych w procesie uczenia EAE
 - ❑ Dekompozycja przestrzeni cech na obszary kompetencyjne w celu eksploracji lokalnych specjalizacji klasyfikatorów w puli
 - ❑ Wykorzystanie innych modeli fuzji
 - ❑ Wykorzystanie innych algorytmów detekcji dryftu
-