Uczenie się klasyfikatorów ze strumieni danych

Jerzy Stefanowski

Instytut Informatyki
Politechnika Poznańska

Sem. PTS, UAM, Poznan, 14 grudnia 2016

Plan wykładu

Część 1: Wstęp

1. Klasyfikacja nadzorowana
   - Spojrzenie - systemy uczące się (ang. machine learning)
   - Wybrane algorytmy
2. Nowe rodzaje danych
3. Strumienie danych
   - Charakterystyka
   - Sposoby przetwarzania strumieni
   - Ocena klasyfikatorów
4. Zmienna definicja pojęć (concept drift)
5. Nowe algorytmy (Hoeffding Trees)

Część 2: zespoły klasyfikatorów dla zmiennych strumieni

Klasyfikacja nadzorowana

- Dane uczące *S* – przykłady ***x*** opisane atrybutami (cechami) *A* oraz etykietowane (ang. target value) y lub *Y*
  - należące do dwóch lub więcej *K* klas

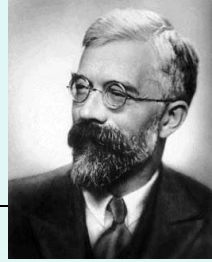
$$S = \left\{ \left(\mathbf{x}_i, c_i \right) \mid x_i \in A^p, c_i \in \{C_1, \ldots C_k\} \right\}_{i=1}^{N}$$

- Uczenie się → odkryć zależność między *A* and *Y* (*C*)

$$y_i = f(\mathbf{x}_i) = c_i$$

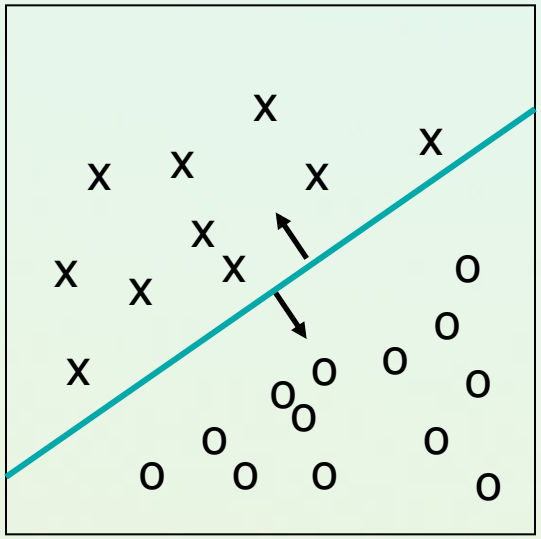
- Dążymy do minimalizacji błędów $y_i \neq c_i$
- Kontekst ML/DM to także:
  - Klasyfikacja porządkowa, wielo-etykietowa, złożone wyjścia y (structured outputs), klasyfikacja grafów oraz sieci, danych sekwencyjnych,
  - Inne paradygmaty: Dane częściowo-etykietowane, Aktywne uczenie się, „Transfer learning”,..

Różnorodność metod



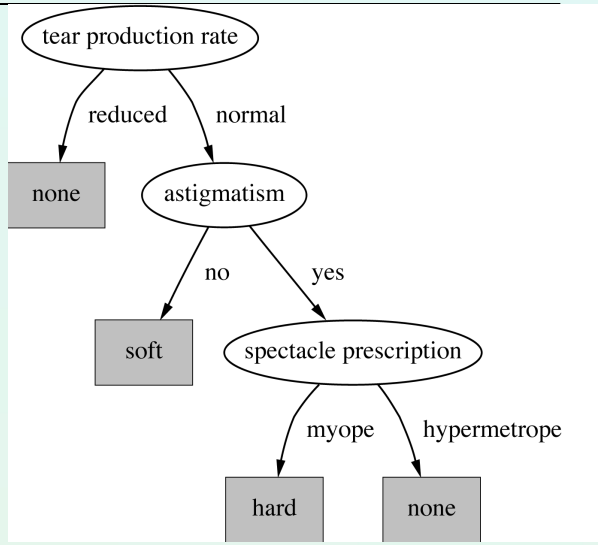
- **Metody symboliczne (reprezentacja wiedzy)**
  - Reguły
  - Drzewa decyzyjne
  - Podejścia logiczne (ILP, EBL)
- **Niesymboliczne**
  - Naïve Bayesian
  - K-najbliższych sąsiadów
  - Analiza dyskryminacyjna
  - Metoda wektorów wspierających (SVM)
  - Sztuczne sieci neuronowe
  - Klasyfikatory genetyczne
  - …
- **Podejścia złożone**
  - Zespoły klasyfikatorów

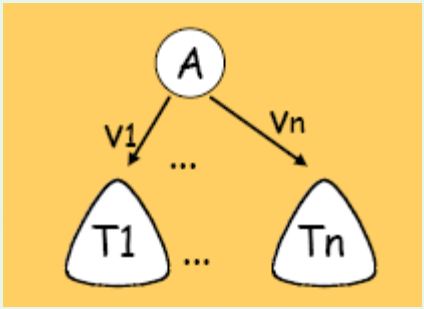
IF Sex = male AND Age > 46 AND
Number_of_painful_joints > 3 AND
Skin_manif. = psoriasis
THEN Diagnosis =
Crystal_induced_synovitis



Metody indukcji drzew decyzyjnych

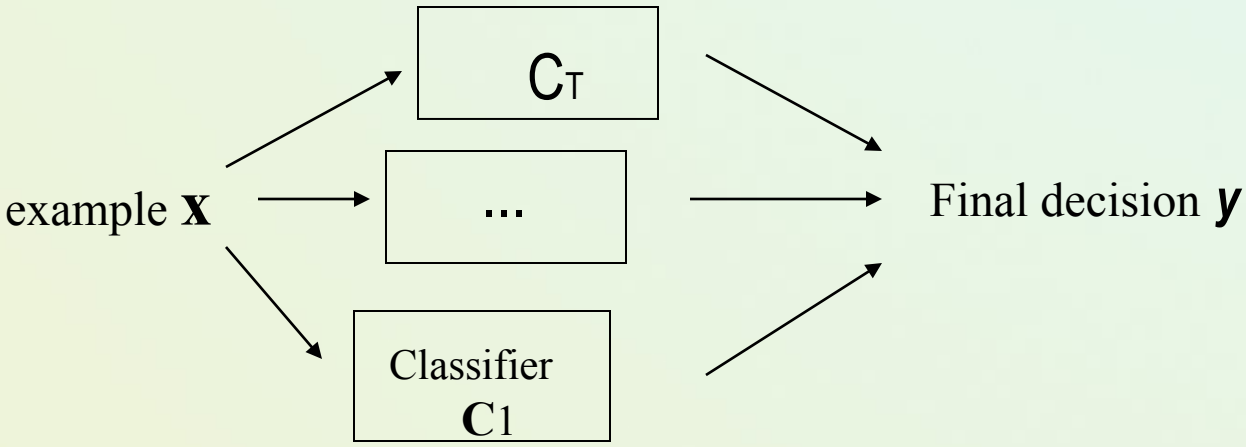
- Konstrukcja drzewa (rekurencyjna procedura)
  - Na początku wszystkie przykłady w węźle.
  - Rekurencyjnie dziel przykłady w oparciu o wybrane testy na wartościach atrybutu (kryterium wyboru najlepszego atrybutu).
  - Warunek zatrzymanie w węźle
- Upraszczanie drzewa - „Tree pruning”
  - Usuwanie poddrzew, które mogą prowadzić do błędnych decyzji podczas klasyfikacji (przeuczenie)
- Przykłady algorytmów: ID3, C4.5, CART,…

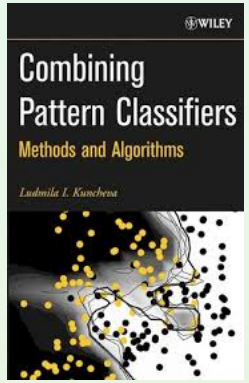




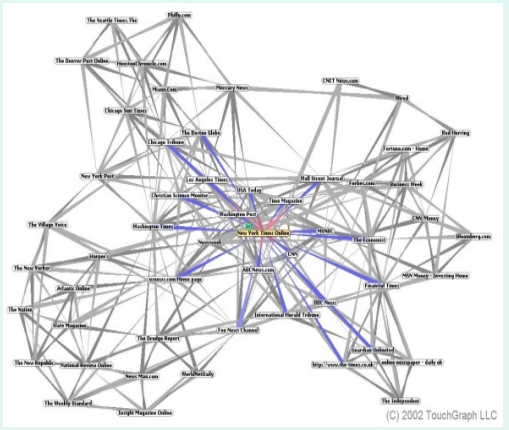
Zespoły klasyfikatorów (ang. multiple classifiers)

- Zbiór indywidualnych klasyfikatorów, który predykcje są agregowane do jednej decyzji.
- Nazwy angielskie: ensemble methods, committees,…
- Cel → polepszyć zdolności predykcyjne
- Problemy:
  - Jak budować klasyfikatory składowe?
  - Jak agregować odpowiedzi?
- Wiele rozwiązań: Bagging, Boosting, Random Forests, …





Nowe źródła danych



Rozwój technologiczny

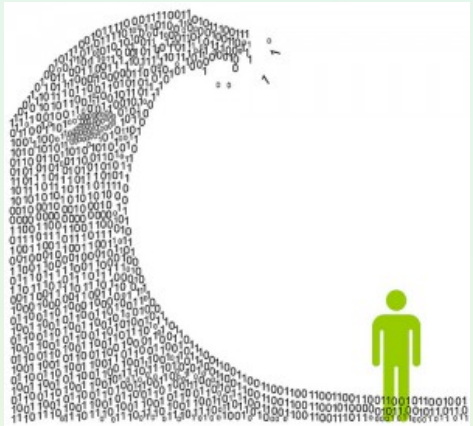
Rosnąca wielkość „tradycyjnych” repozytoriów

Nowe źródła danych i wzrost złożoności danych:

- Internet, media społecznościowe
- Sieci sensorów, urządzenia mobilne, dane generowane maszynowo

„In many fieds costs of data acquisition are now lower, much lower than costs of data analysis”

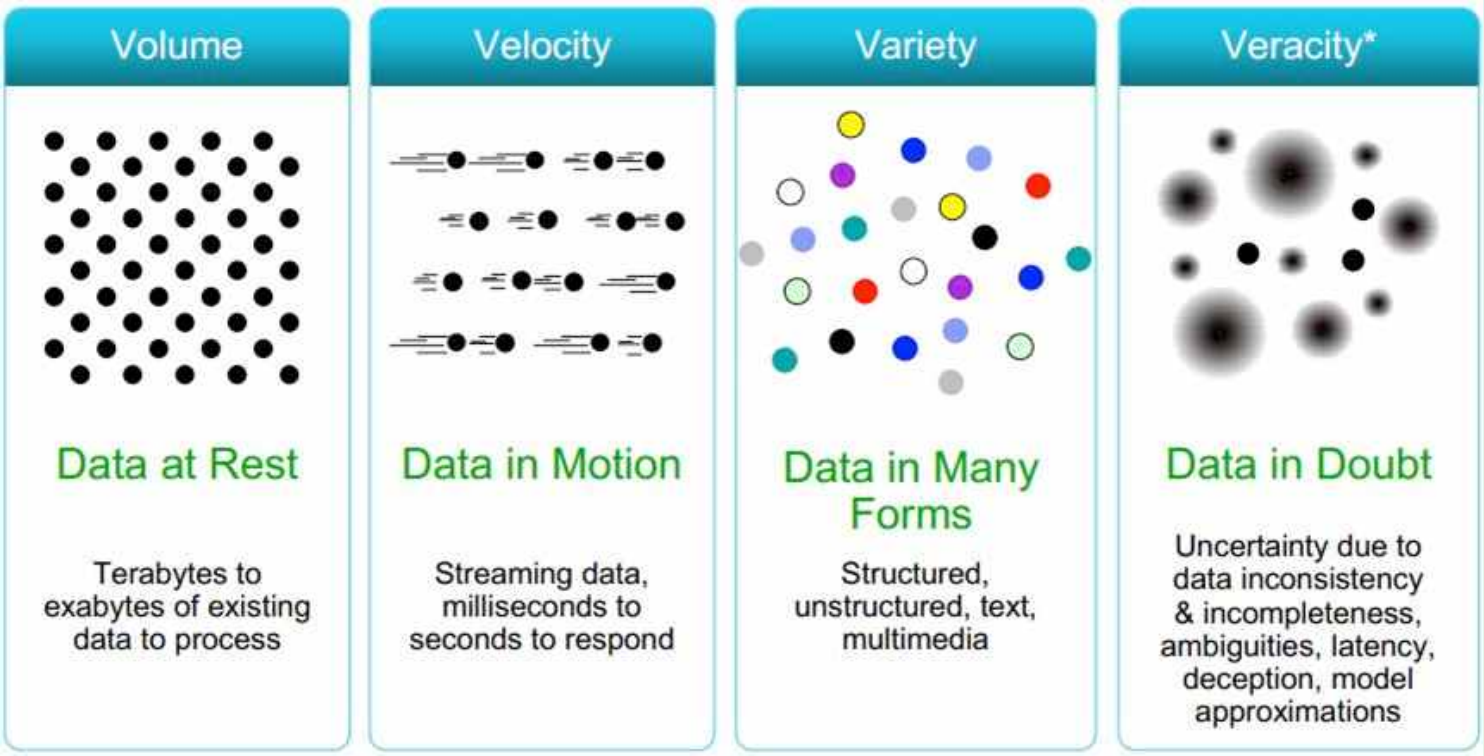
[Canadian Statistical Sci . Inst 2015]



Wzrost przekonania o roli Big Data

Wiele V – charakterystyka Big Data

3 V → „High volume, velocity and variety” [Doug Laney 2001]



2012 IBM Solutions presentation

Kolejne V’s stopniowo dodawano do definicji (Value, Variability, …)

Velocity – czas, prędkość i zmienność

- Dane-generowane z dużą szybkością i wymagają odpowiednio szybkiego przetwarzania
- „Online Data Analytics”
- Spóźnione decyzje → nieakceptowane straty
- Przykłady
  - e-marketing → oferty we właściwym czasie
  - Monitorowanie stanu zdrowia
  - Decyzje finansowe
  - Wykrywanie defraudacji, zagrożeń
- Wiele problemów → strumienie danych
  - Przyrostowe i wydajne obliczeniowo przetwarzanie
  - Zmienność rozkładów danych

Potrzebne nowe algorytmy eksploracji danych !

Co to jest strumień danych?

- *"A **data stream** is a potentially unbounded, ordered sequence of data items, which arrive continuously at high-speeds"*

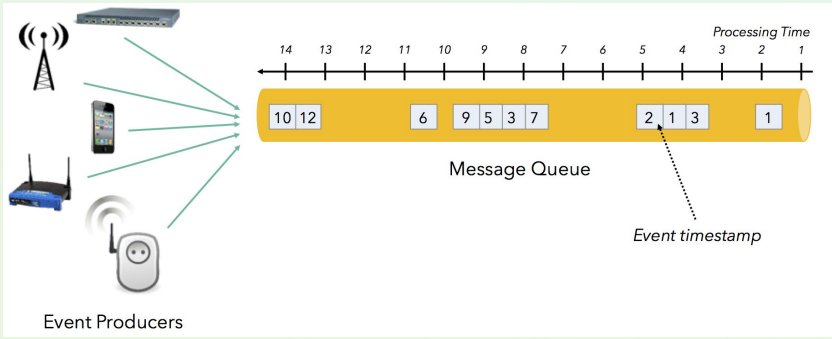
  Springer Encyclopedia of Machine Leaning

- *"It is impossible to control the order in which items arrive, nor is it feasible to locally **store** a stream in its entirety"*
- Struktura danych ≠ audio lub video data
- Masywne dane, pojawiające się z dużą prędkością

Timestamp	Puis. A (kW)	Puis. R (kVAR)	U 1 (V)	I 1 (A)
…	…	…	…	…
16/12/2006-17:26	5,374	0,498	233,29	23
16/12/2006-17:27	5,388	0,502	233,74	23
16/12/2006-17:28	3,666	0,528	235,68	15,8
16/12/2006-17:29	3,52	0,522	235,02	15
…	…	…	…	…

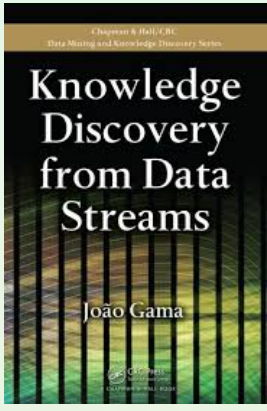
Przykładowe dziedziny zastosowań

- Sieci sensorów
- Analiza ruchu sieci
- Media społeczne (WWW)
- Zachowanie użytkowników sklepów internetowych
- Obserwacja rynków finansowych
- Filtrowanie informacji lub wiadomości sieciowych
- Telemechanika (energetyka)
- Sterowanie obiektami przemysłowymi
- Inteligentne miasta i IoT (Internet przedmiotów)
- Systemy monitorowania
- Opieka nad osobami chorymi, niepełnosprawnymi
- …



Charakterystyka strumieni danych

- Dane dostępne sekwencyjnie (potencjalnie asynchronicznie) potencjalnie nieograniczone
- Braku kontroli systemu nad porządkiem pojawiania się przykładów
- Gotowość modelu do natychmiastowej analizy i predykcji
  - „Any time classifier”
- Krytyczny czas przetwarzania i odpowiedzi
- Jednorazowy odczyt elementu danych (ang. single scan algorithm)
  - Brak możliwości zapamiętania całego strumieni
- Zmienność rozkładów danych



Charakterystyka strumieni danych - cd

- Źródła generują dane w postaci nieprzerwanego strumienia
  - Trudności z przechowywaniem wszystkich elementów danych
  - Konieczność przetwarzania szybkiego strumienia w ograniczonym czasie
  - Przyrostowe działanie algorytmów
- Zmienność rozkładów danych → niestacjonarne środowiska

	Statyczne	Strumień
No. odczytów	wielokrotne	pojedyncze
Czas	nie jest krytyczny	ograniczony
Uzycie PAO	b. elastyczne	ograniczone
Rezultaty	dokładne	przybliżone
Przetwarz. Rozproszne	zwykle nie	tak

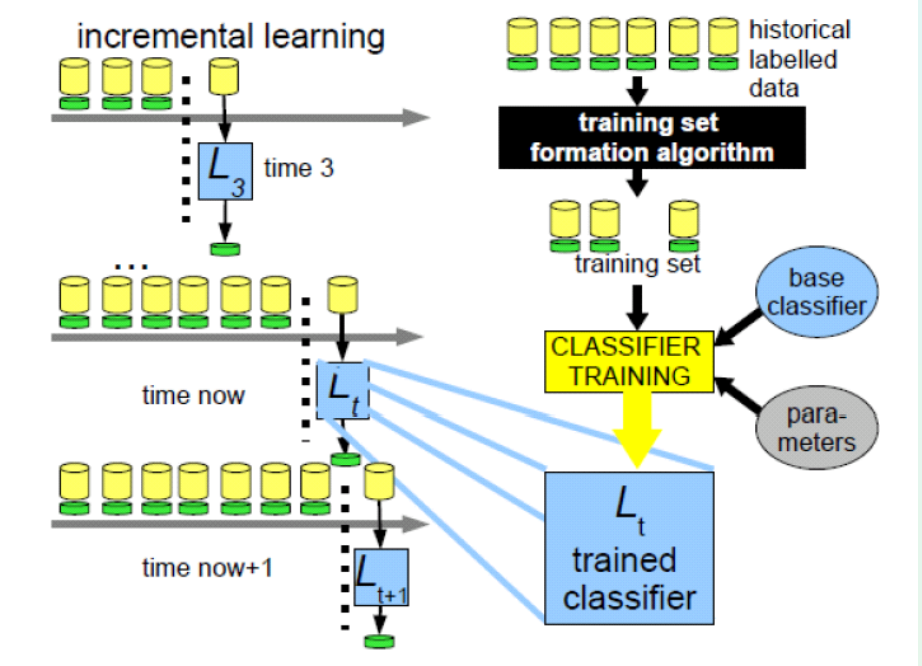
Nowe wyzwania:

- Próbkowanie danych, tworzenie zastępczych reprezentacji (histogramy)
- Grupowanie danych (analiza skupień)
- Analiza zbiorów częstych, reguł asocjacyjnych, wzorców sekwencyjnych

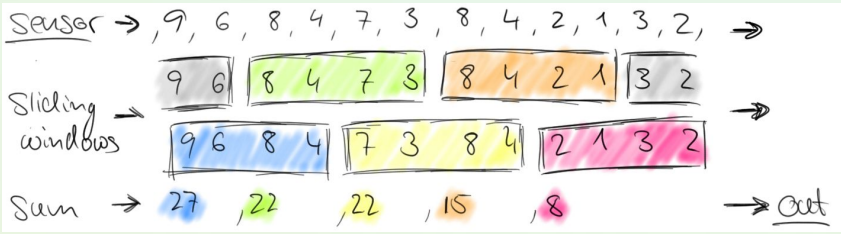
Także klasyfikacja i modele predykcyjne

Poprzednie badania nad alg. przyrostowymi

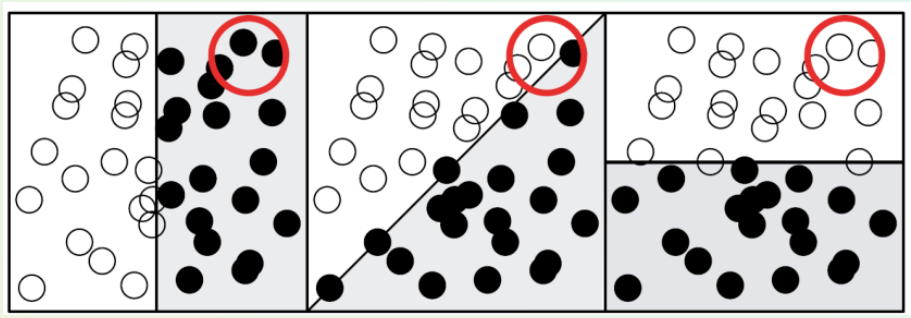
- “Incremental learning vs. batch”
  - Sztuczne sieci neuronowe
  - Rozszerzenia k-NN (Aha’s IBL)
  - Klasyfikatory Bayesowskie
- Analiza szeregów czasowych
- Przyrostowe wersji reprezentacji symbolicznych
  - Drzewa decyzyjne ID5 (Utgoff)
  - Reguły AQ11 PM (Michalski)
- Grupowanie – COBWEB (D.Fisher); BIRCH, ..
- Wybór przykładów (partial memory)
  - Windowing (Sliding vs. landmark)
  - Sampling for k-means like clustering algorithms
- Lecz – nie spełniają wymagań wobec danych strumieniowych!

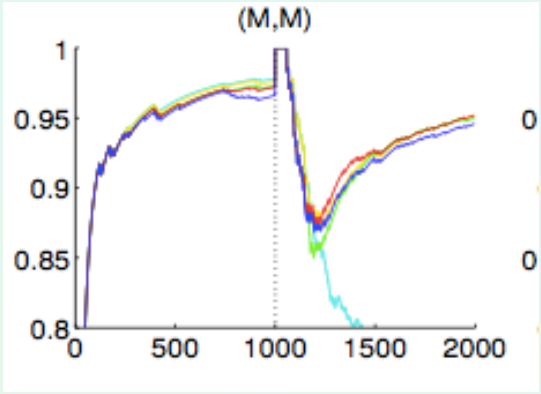


Rysunek I Zliobaite



Zmienne strumienie danych (concept drift)





Zmienność danych (także definicji klas) wraz z upływem czasu tzw. concept drift

Zmiany te mają negatywny wpływ na trafność klasyfikacji

Statyczne algorytmy eksploracji danych nie mogą być stosowane!

Nowe wymagania do algorytmów – wydajne obliczeniowo + zdolności reakcji na zmiany

Przykłady zmienności w strumieniach

- Wykrywanie “niechcianych wiadomość”
  - “Old spam is not a new spam”
  - Twórcy spamu poszukują nowych rozwiązań
- Analizowanie preferencji klientów
- Zmienne rynki finansowe
- Filtrowanie informacji oraz systemy rekomendacyjne
  - Co to jest interesujący produkt (książka, muzyka)?
- Wykrywanie niewłaściwych zachowań (ang. intruder detection)
- Przewidywania cen oraz zapotrzebowania.

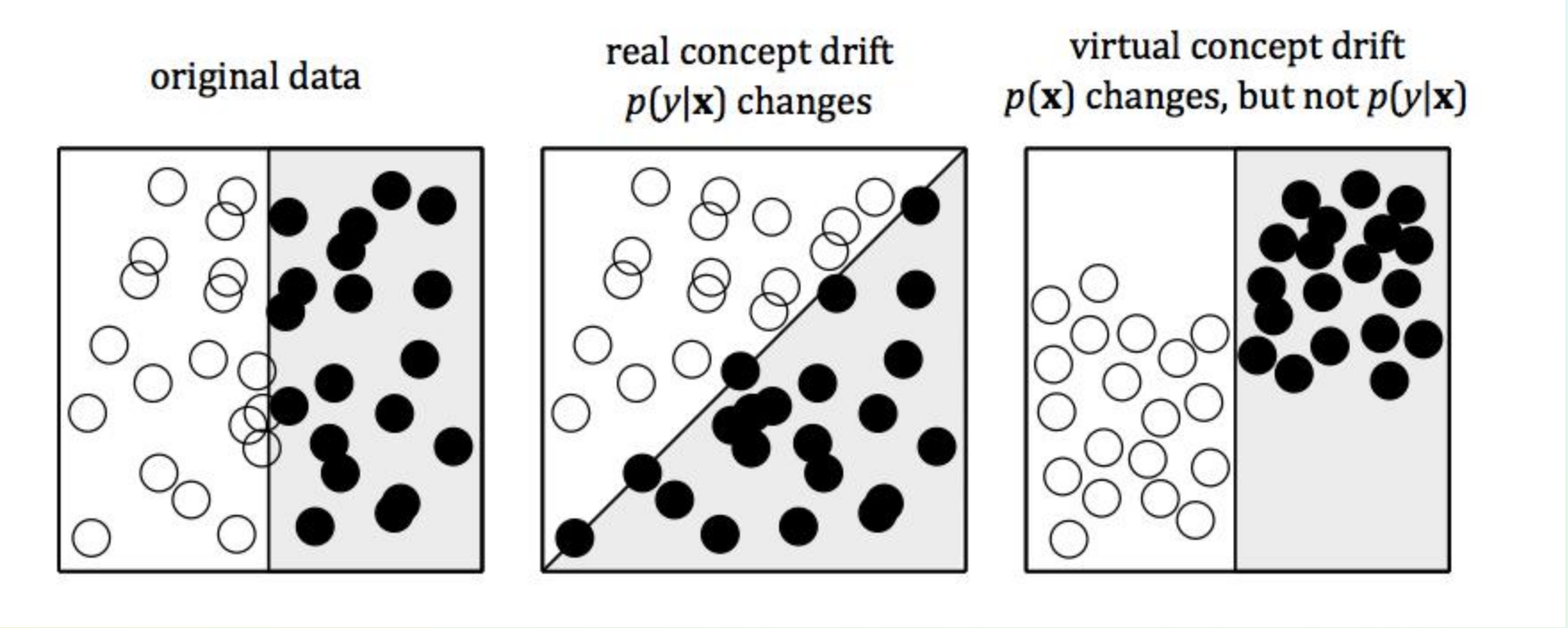
Więcej: Indre Zliobaite, Mykola Pechenizkiy, and Joao Gama: An overview of concept drift applications. Chapter 4 in N.Japkowicz and J.Stefanowski (Eds) Big Data Analysis: New Algorithms for a New Society, Springer (2016).

Zmienność pojęć

Strumień S – sekwencja x_t (t=1,2, …, T)
→ Etykieta y_t nowego przykładu (dostępna) może być użyta do uczenia klasyfikatora C

- Łączny rozkład prawdopodobieństwa $p^t(x,y)$ w chwili t
- Zmienność strumienia → dla czasu t oraz t+Δ, istnieje x taki że $p^t(x,y) \neq p^{t+\Delta}(x,y)$
  - Różne składowe prawdopodobieństwa mogą podlegać zmianom
- Real drift (typowe dla klasyfikacji nadzorowanej)
  - posterior prob. klas $p(y|x)$ zmienia się
- Virtual drift → zmiany w danych, np. $p(x)$, niewpływające na $p(y|x)$, także zmienne prawdopodobieństwa klas $p(y)$

Rzeczywisty vs. wirtualny dryft pojęcia



Za: Dariusz Brzeziński: Block-based and online ensembles for concept drifting data streams. PhD Thesis, Poznań University of Technology, 2015.

Rodzaje zmian pojęć (concept drift)

Strumień S = <S_1,S_2,S_3,…,S_n>, gdzie część S_i generowana przez źródło ze stacjonarnym rozkładem D_i

Dryft → przejście między S_j i S_{j+1}

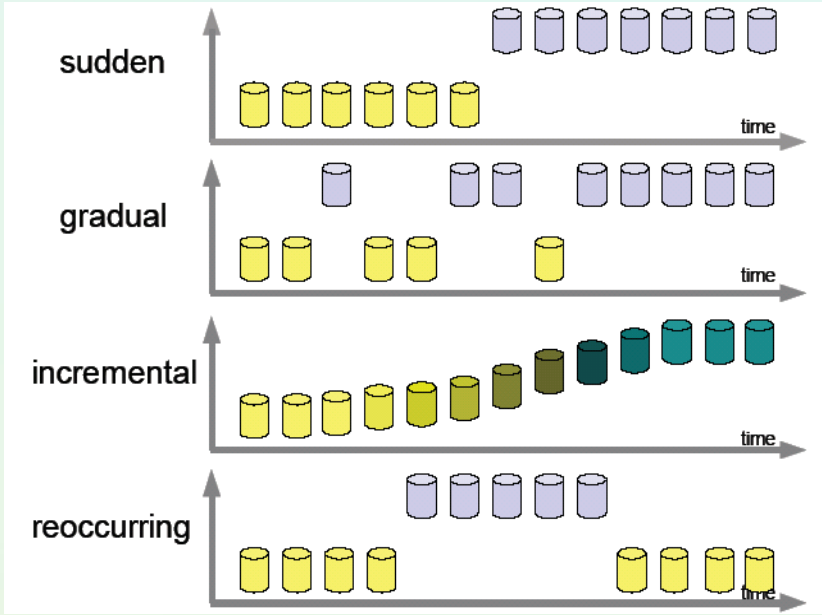
(Widmer → Hidden context of changes)

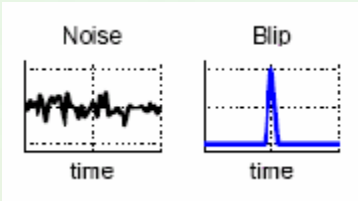
Różne typy zmian

- A **sudden** (abrupt) drift - S_j jest nagle zastąpione przez S_{j+1} ($D_j \neq D_{j+1}$)
- **Gradual** drifts – wolniejsze tempo zmiany
  - W strefie zmian wymieszane prawdopodobieństwa pojawiania się S_j i S_{j+1}
  - Incremental – ciąg drobnych zmian
- **Reoccurring** concepts

Nie powinno się reagować na anomalie (blips)

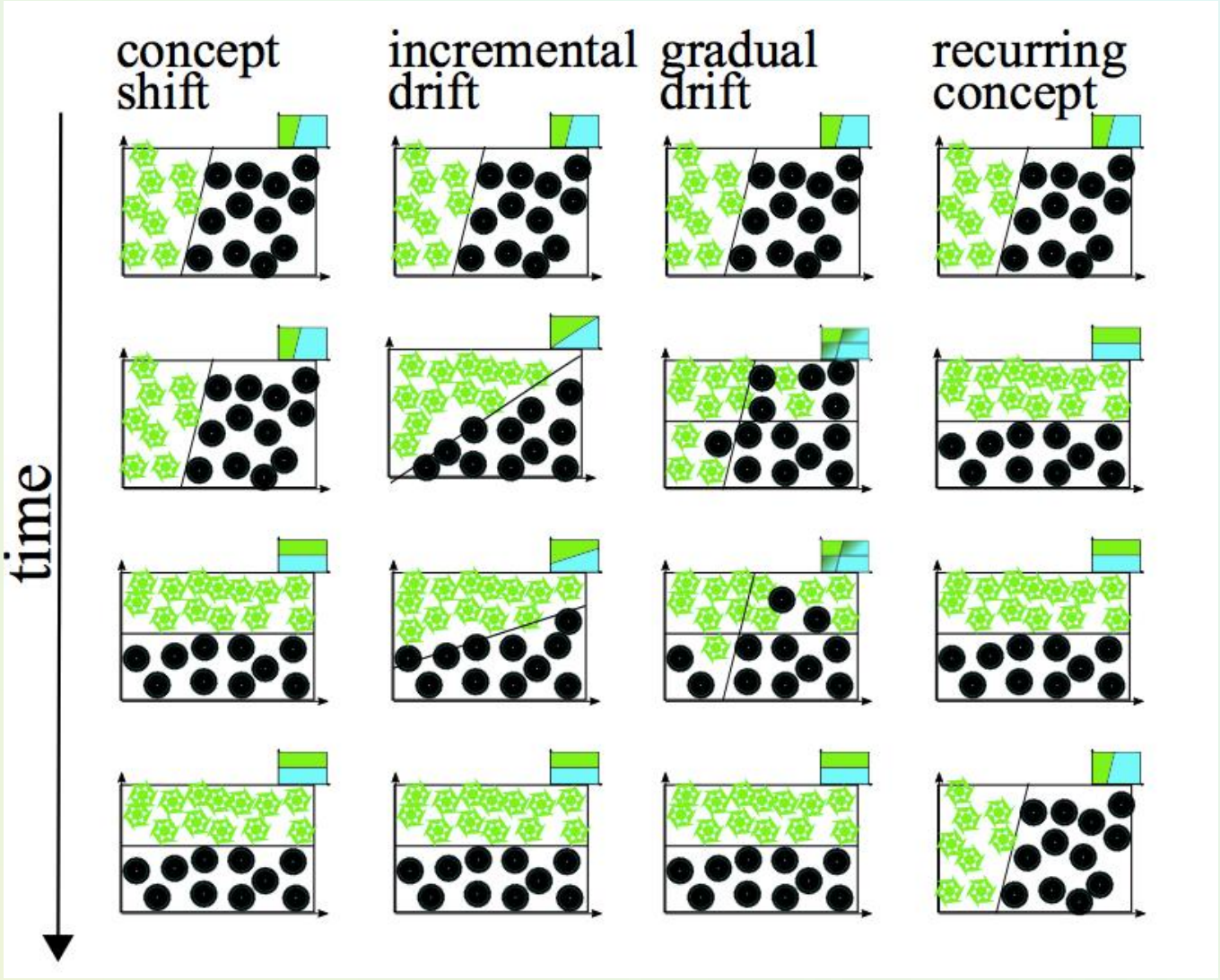
oraz szum informacyjny w sekwencji (noise)





Figures - I.Zliobaite

Ilustracja zmian pojęć



Za: Ammar Shaker: Novel methods for mining and learning from data streams. PhD Thesis, Paderborn University, 2016.

Więcej o zmienności pojęć

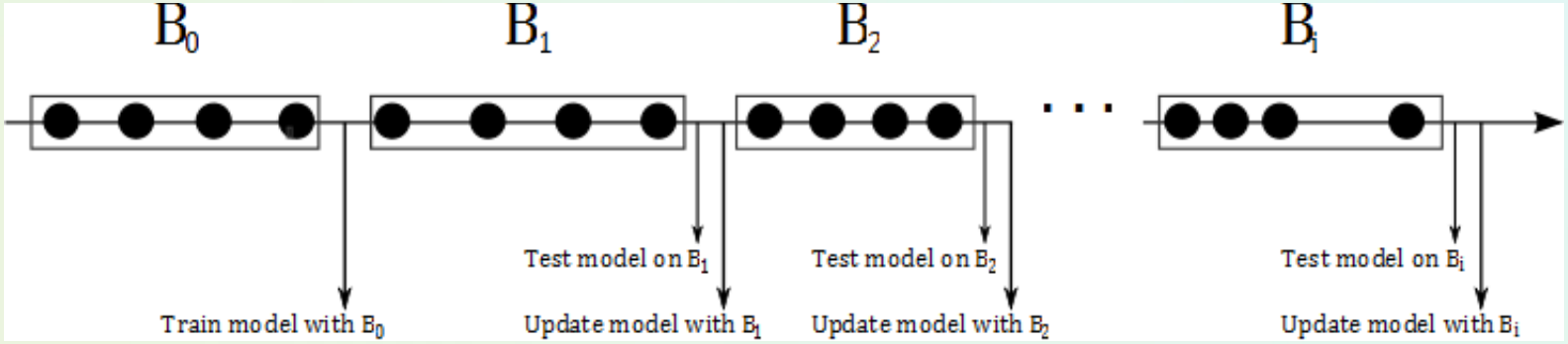
- Real concept drift
  - Pełny lub dotyczący części pojęć - sub-concept drifts
    - Drift severity (scope) – drift limited to a subspace of Dom(x) [Minku et al. 2010]
- Drift reoccurrence
  - Cyclical drifts – powtarzanie się pojęć w pewnej kolejności [Tsymbal 2004]; stałe albo zmienne okresy
  - Non-cyclical drifts
- Covariance drift (a part of virtual drift)

  $p^t(x) \neq p^{t+\Delta}(x)$
- Novel class appearance [Masud et al. 2011]

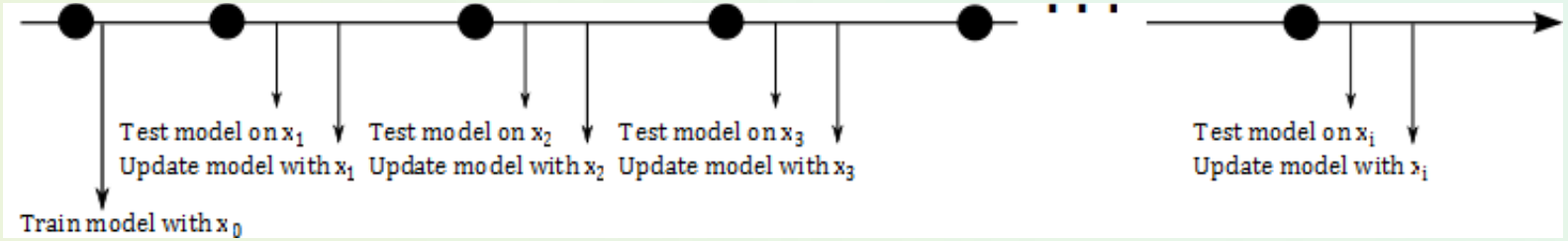
  $p^t(y=C_l)=0$ for *t* and $p^{t+\Delta}(y=C_l)>0$

Różne sposoby przetwarzenia danych

Block processing



Online processing

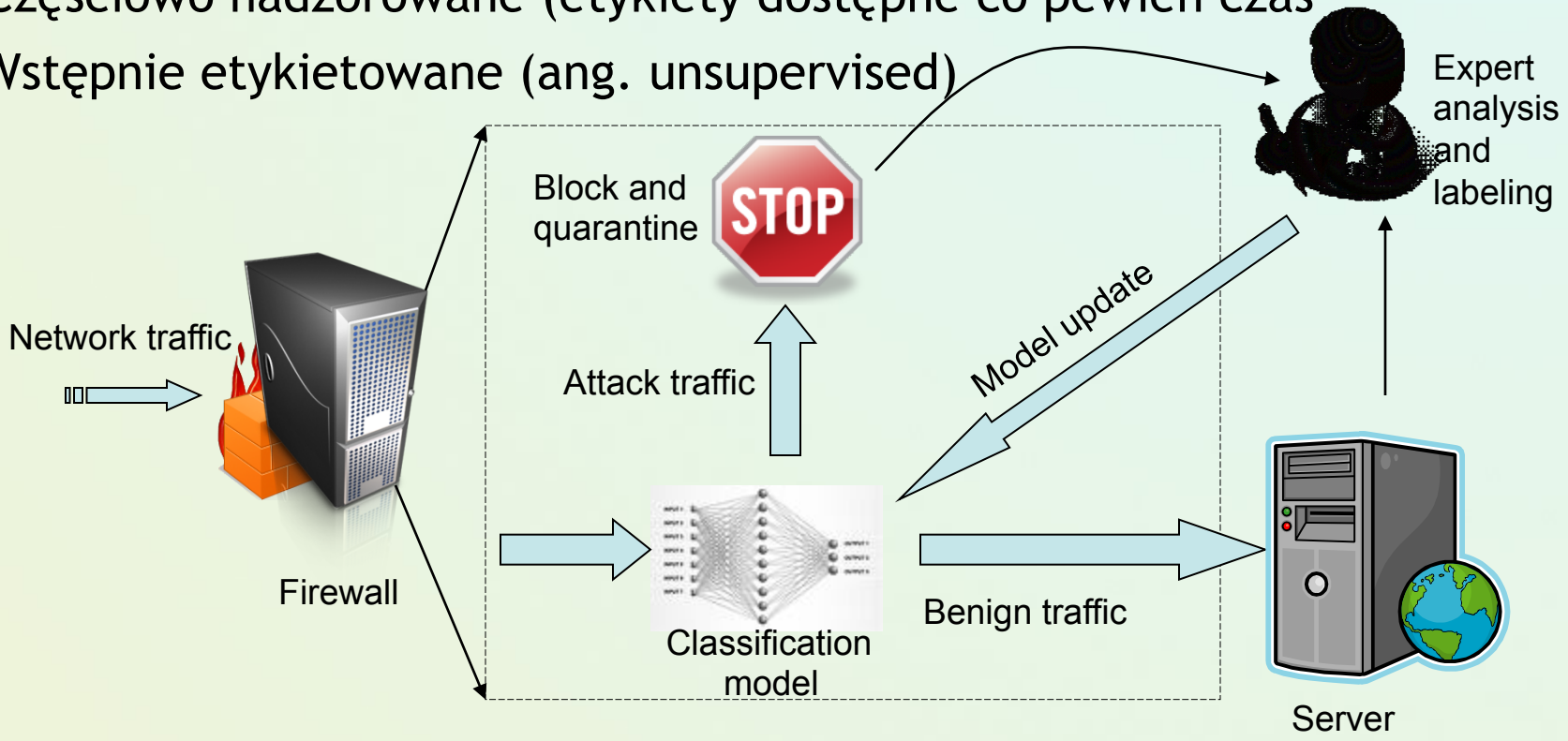


Wpływ na budowę klasyfikatorów i ich ocenę

Dostępność etykiet przykładów

Założenia co do dostępności etykiet

- Pełne i natychmiastowe
- Opóźnienie dostępu do etykiet (ang. delayed labeling)
- Częściowo nadzorowane (etykiety dostępne co pewien czas
- Wstępnie etykietowane (ang. unsupervised)



Ocena klasyfikatorów strumieniowych

- Holdout

  [np., Kirkby 2007]

- Test-then-train

  [np., Kirkby 2007]

- Block-based evaluation

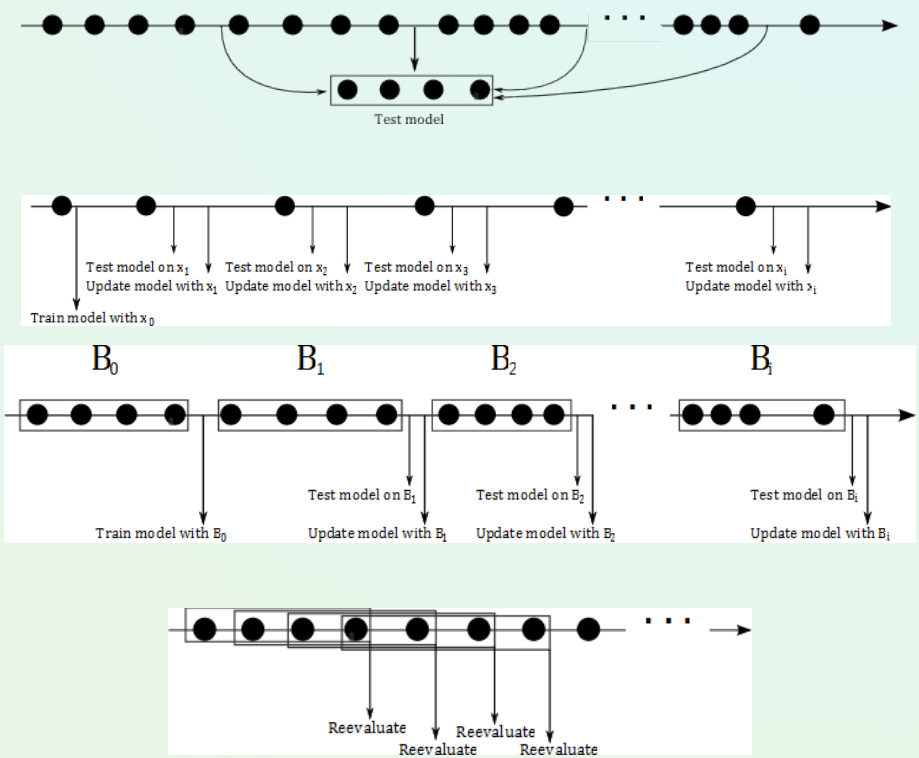
  [np., Brzezinski & Stefanowski 2010]

- Prequential accuracy

  [Gama et al. 2013]

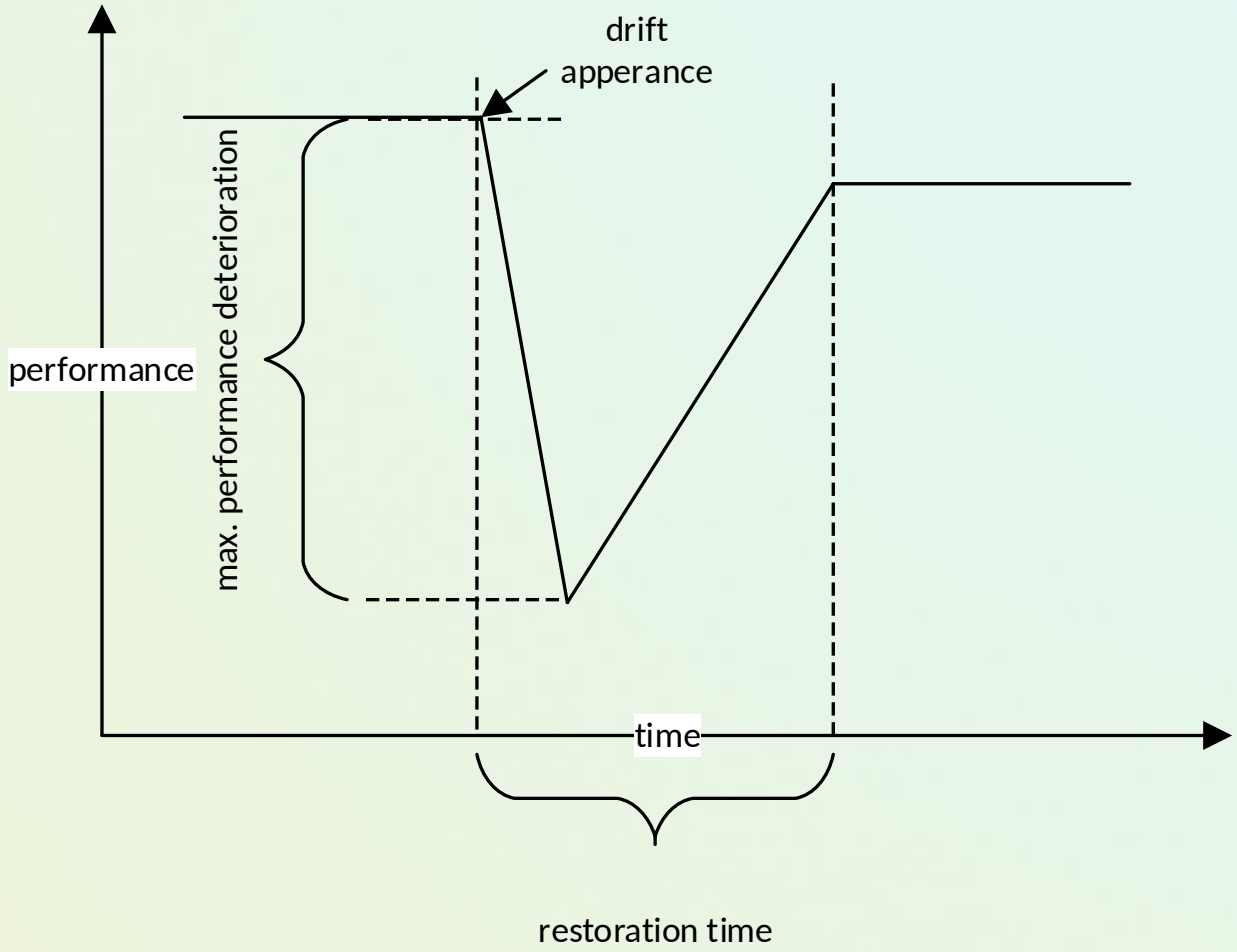
- Inne miary

  [Bifet & Frank 2010, Zliobaite et al. 2014]



Na ogół stosuje się proste obliczeniowo miary (błąd, trafność)
Także miary efektywności (pamięć, czas) oraz zdolność reakcji

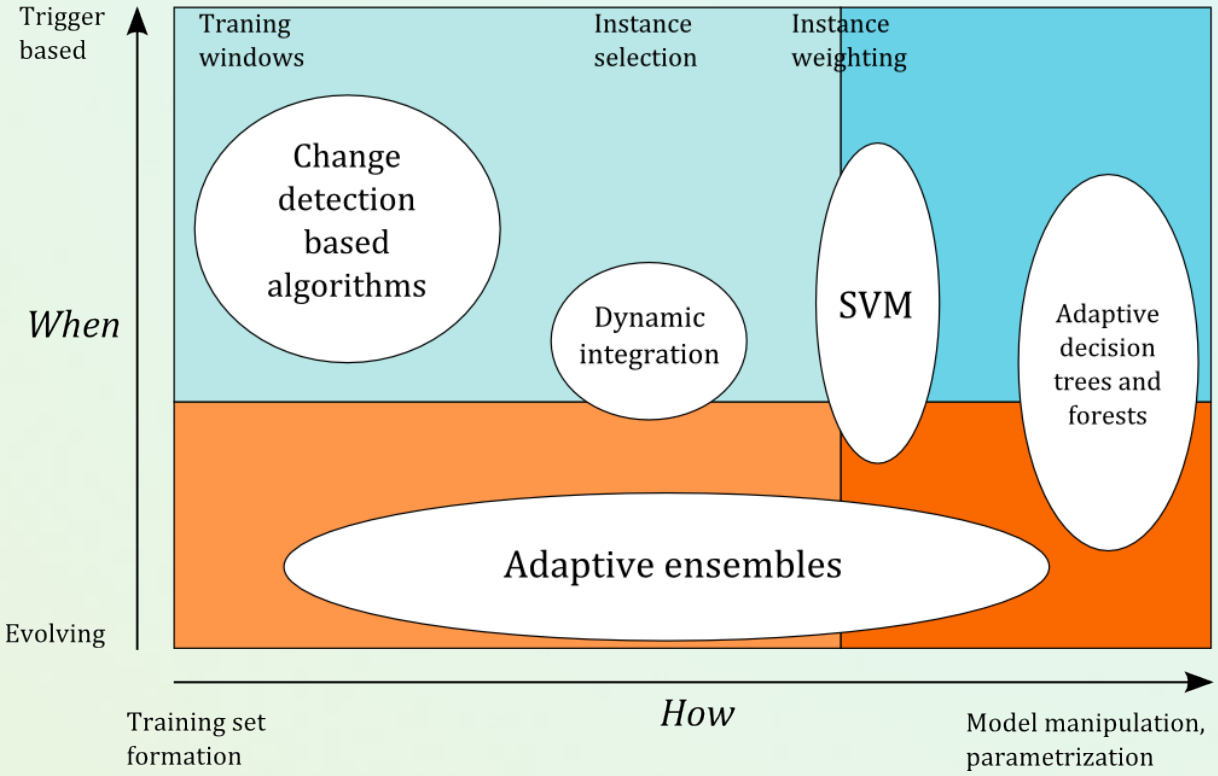
Ocena reakcji wobec zmian w strumienia



Wykrycie dryftu i zdolność adaptacji klasyfikatora

Podział metod [Indre Zliobaite]

- Triggers - detekcja zmiany
- Metody adaptacyjne



Pojedyncze klasyfikatory: VFDT, FISH, FLORA, RILL
Zespoły: SEA, AWE, HOT, Online Bagging, DDD, ADWIN, DWM, OAUE

Drzewa klasyfikacyjne i … (regresji …)

80's

ID3

CART

CHAID

C4.5

…

Oblique trees

Model Trees

Ensembles

90

Mega-induction,
Sampling, splits

Bagging

Boosting

Random Forests

Ivotes

KDD generalizations

end. 90

Data pre-sorting / reduce disc acc.

Approx. representations
Samples, Histograms

Rotation Forests

Gradient Boosted Machines

SLIQ
SPRINT
…

BOAT
CLOUDS
SPIES

Parallel versions

Incremental trees
ID4, ID5R, ITL

Massive incremental tress
Hoeffding trees, **VFDT**

MOA,
SAMOA

Stream ensembles
Online …/ block based

Data streams + drifts
VFDT -> CVFDT; **VFDTc**

Rozwój implementacji w MapReduce, Hadoop / SPARK

Hoeffding Tree

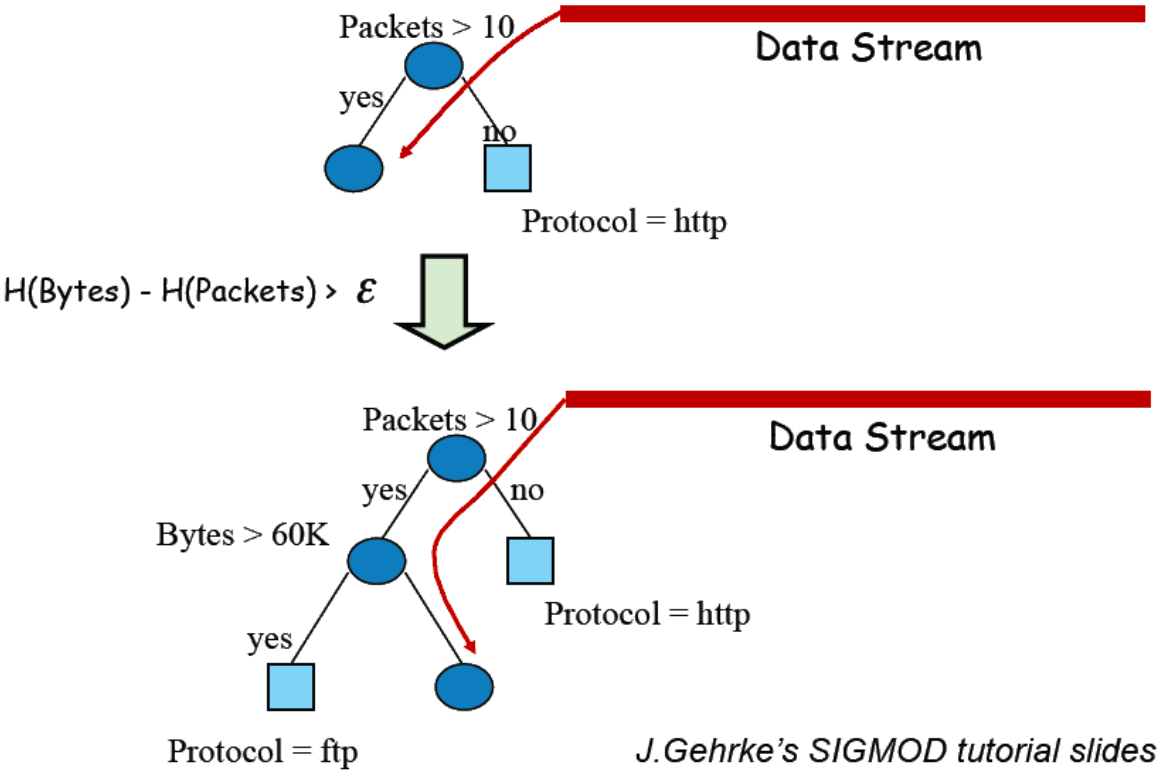
„Mining High-Speed Data Streams”, P. Domingos, G. Hulten; KDD 2000

Główna idea:

Mała próbka danych często może wystarczyć do określenia „najlepszego” atrybutu podziału drzewa decyzyjnego

- Zebranie odpowiednich statystyk z próbki strumienia
- Estymacja wartości funkcji oceny podziału dla każdego atrybutu
- Wykorzystanie granicy Hoeffdinga do zagwarantowania dobrego wyboru atrybutu podziału

Wykorzystanie organiczenia Hoeffiding'a



$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

R – zakres wartości estymowanej funkcji

δ – dopuszczalny błąd estymacji

n – rozmiar próbki

Granica Hoeffdinga jest prawdziwa dla dowolnego rozkładu danych

Very Fast Decision Trees: Main Algorithm

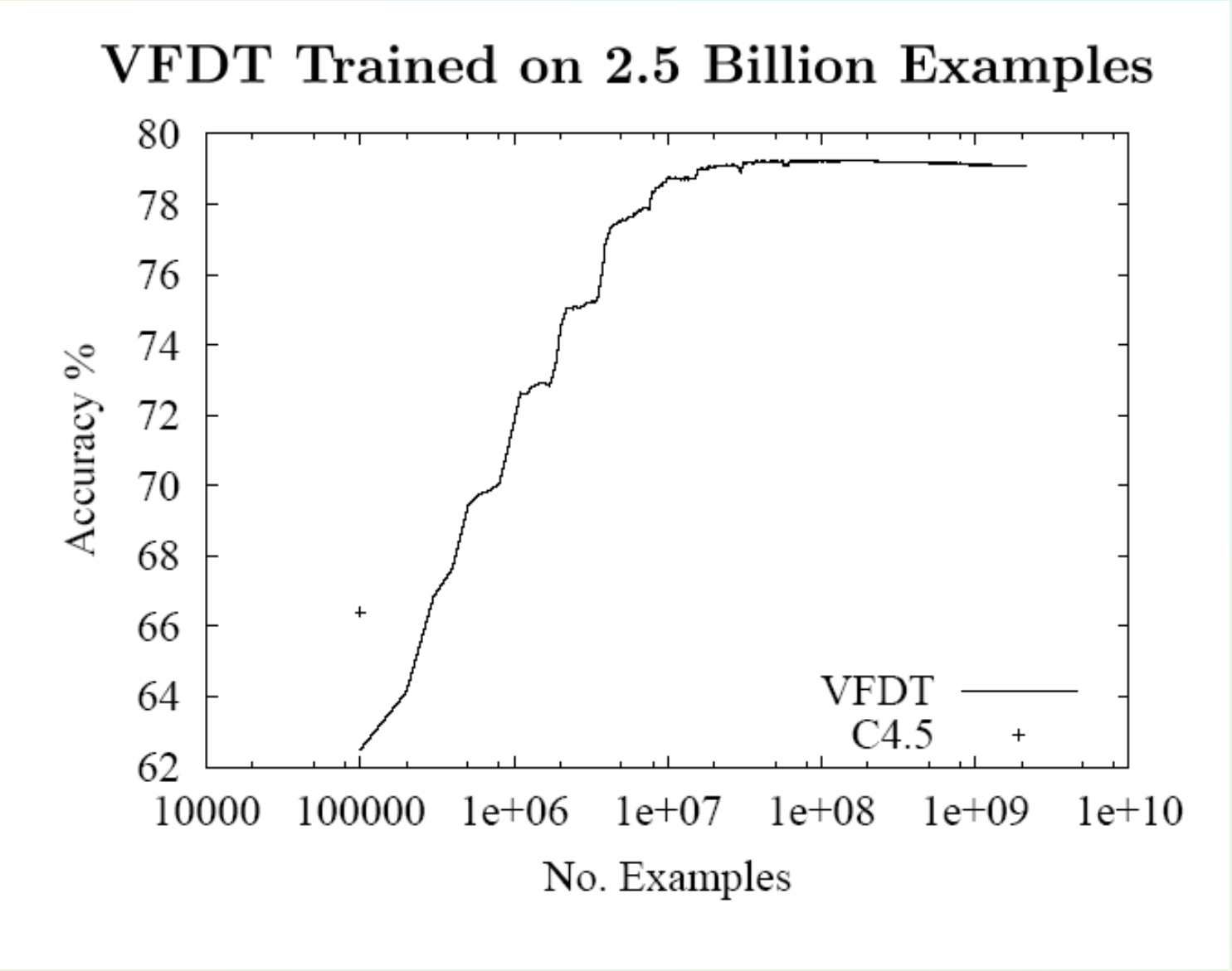
- **Input:** δ desired probability level.
- **Output:** $\mathcal{T}$ A decision Tree
- **Init:** $\mathcal{T} \leftarrow$ Empty Leaf (Root)
- While (TRUE)
  - Read next Example
  - Propagate Example through the Tree from the Root till a leaf
  - Update Sufficient Statistics at leaf
  - If $leaf\,(\#examples) > N_{min}$
    - Evaluate the merit of each attribute
    - Let A_1 the best attribute and A_2 the second best
    - Let $\epsilon = \sqrt{R^2 ln(1/\delta)/(2n)}$
    - If $G(A_1) - G(A_2) > \epsilon$
    - Install a splitting test based on A_1
    - Expand the tree with two descendant leaves

Za: wykład J.Gama nt. Mining data streams 2007

VFDT (Very Fast Decision Tree)

- Modyfikacja Hoeffding Tree
  - G - kryterium obliczenie co n_{min}
  - Dodatkowe upraszczanie nieaktywnych poddrzew
  - Słabe atrybuty pomijaj
  - Wstępnie zbuduj mini-drzewo
- Porównanie do Hoeffding Tree: Lepszy czas i pamięć
- Studnia porównawcze do tradycyjnych drzew
  - Przynajmniej porównywalna trafność klasyfikowania
  - Lepsze przetwarzanie (np. 1.61 milionów przykładów)
    - 21 min. VFDT
    - 24 godz. C4.5
- Nie adaptuje się do zmian - concept drift
  - Zapominanie z przesuwanymi oknami CVFDT – J.Gama et al..

Eksperyment Domingos & Hulten



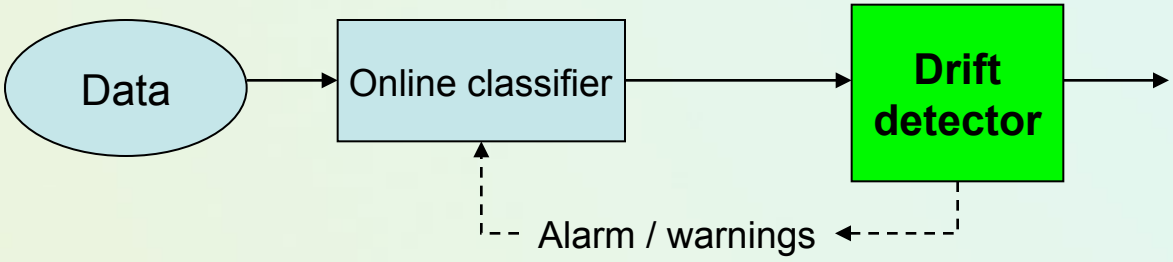
Plan drugiej części wykładu

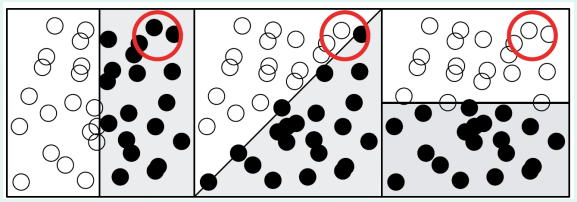
Zespoły klasyfikatorów dla zmiennych strumieni danych

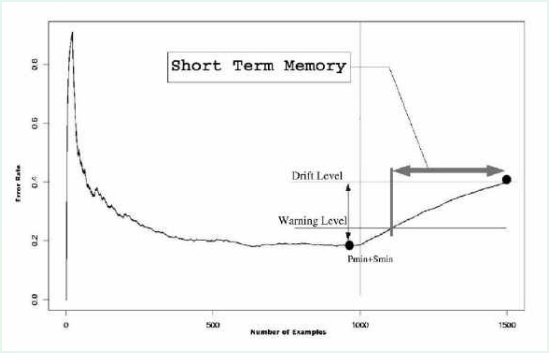
1. Concept drift - detekcja
2. Zespoły klasyfikatorów dla zmiennych strumieni
3. Nasze propozycje (z D.Brzeziński)
   → Accuracy Updated Ensemble (AUE)
   → Online Accuracy Updated Ensemble (OAUE)
4. Niezbalansowanie klas w strumieniu
5. Otwarte problemy i kierunki badań

Przepraszam → kolejne slajdy będą w języku angielskim

Triggers – wykorzystanie detektorów dryftu







Statistical Process Control

- DDM, EWMA,…

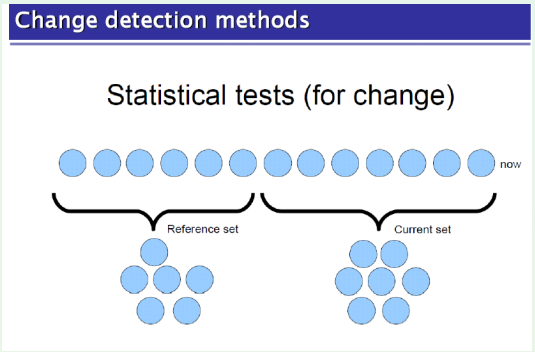
Sequential Analysis

- Cumulative Sum Test, Page-Hinkley test

Monitoring distributions over windows

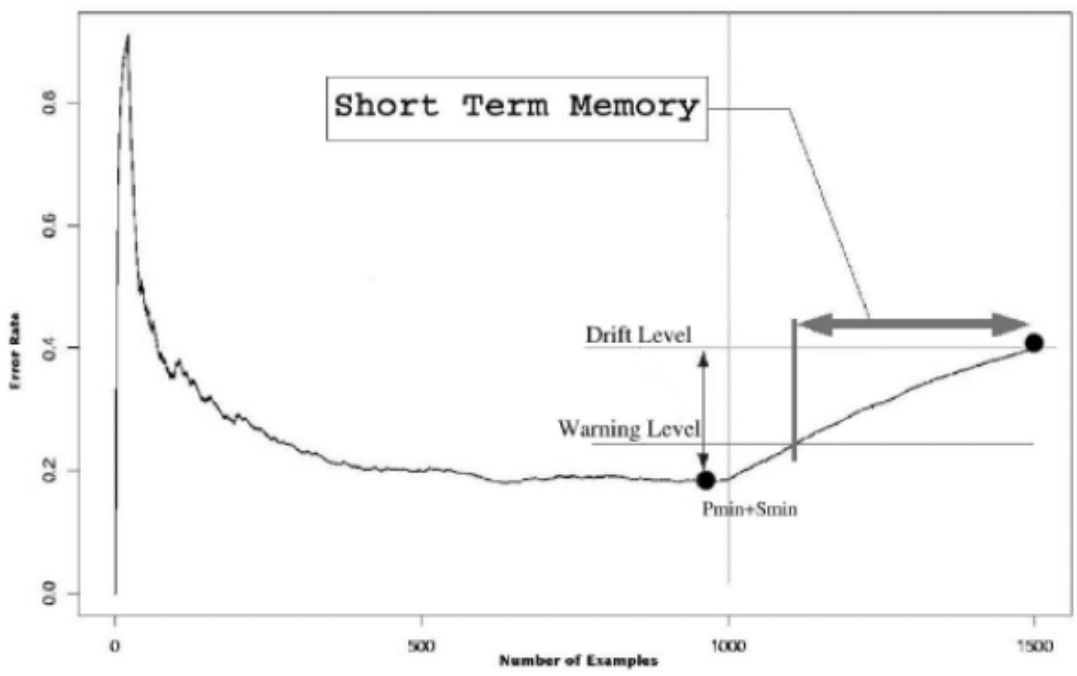
- ADWIN

Context approaches



Przegląd: J.Gama, I.Zliobaite, M.Pechenizkiy, A. Bouchachia: A Survey on Concept Drift Adaptation. ACM Compt. 2013

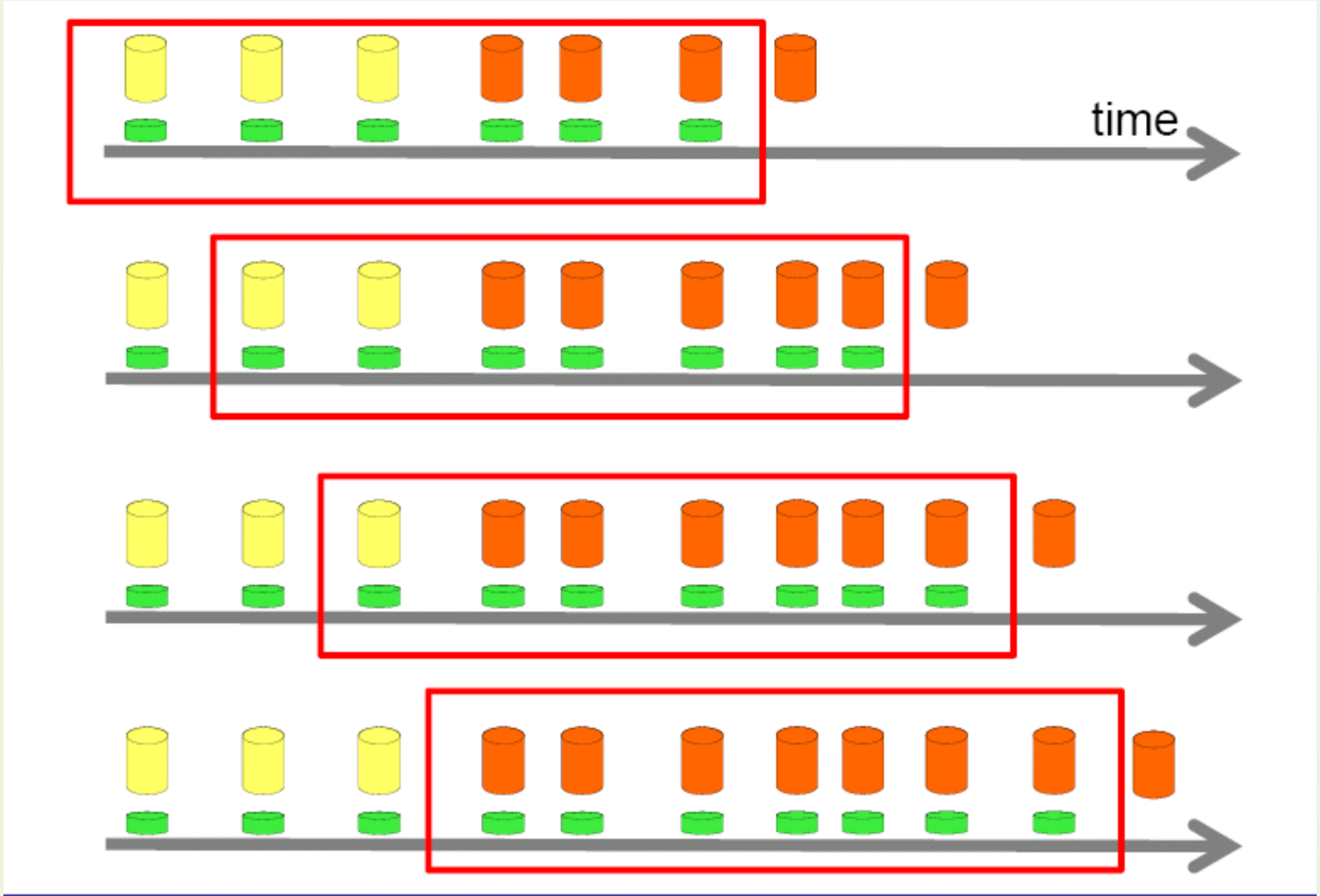
DDM – idea [Gama 2004]



Gama J., Medas P., Castillo G. and Rodrigues P.: Learning with Drift Detection., In SBIA Brazilian Symposium on Artificial Intelligence, LNAI 3171, pp. 286-295, 2004

Sliding windows (przesuwane okna) – tzw. partial memory (Maloof, R.Michalski)

Retrain examples with selected examples (but computational costs)

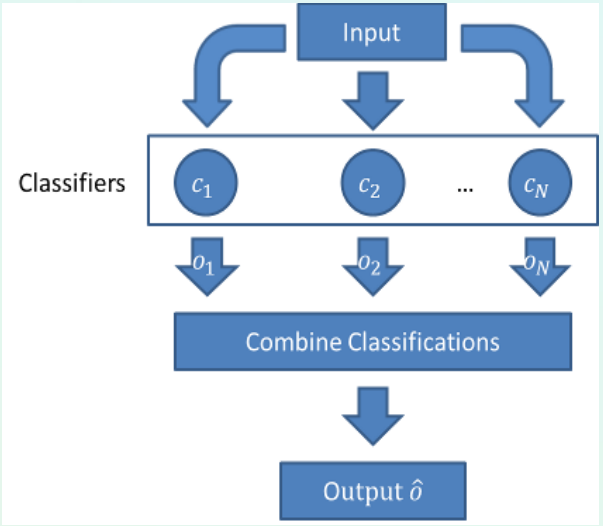


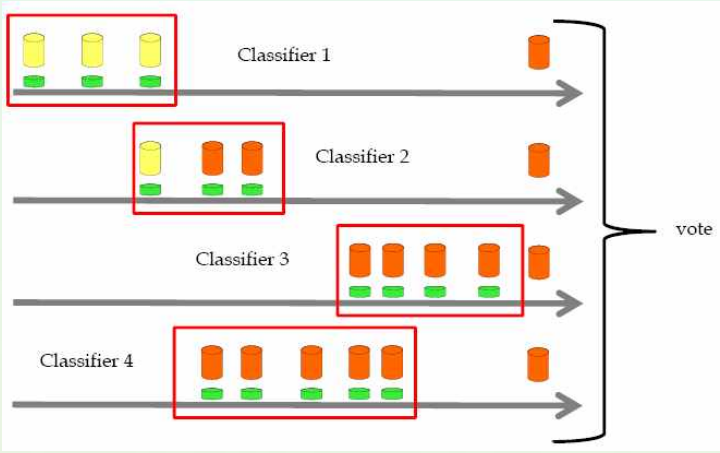
Zastosowanie zespołów klasyfikatorów (ang. ensembles) do zmiennych strumieni

- Many proposals for static data
- Natural for non-stationary frameworks
  - Modular construction
    - Flexibility to incorporate new data
      - adding new components
      - updating existing components
    - Natural forgetting
      → pruning ensembles
    - Continuous adapting aggregation (voting weights) technique
    - Reduce the variance of the error comparing to single classifiers
      → stability
  - Another motivation

Stability-plasticity dilemma

During changes data generated as a mixture of distributions
→ may be modeled as a weighted combination





Multiple classifiers for concept drifting streams

- Different taxonomies → Kuncheva (2004)

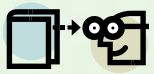
  Dynamic combiners → component classifiers learn in advance, adapting by changing the combination rule [Weighted Majority]

  Updating training data → recent data use to on-line update of component classifiers [Oza]

  Updating ensemble members → update on-line or retrain in a batch mode

  Structural changes of the ensemble → replace "the loser" and add new component [Street; Wang]

- Trigger vs. Adaptive → Active vs. Passive
- This presentation
  - On-line ensembles → learn incrementally after processing single examples
  - Block-based ensembles → learn after processing blocks of data



Prace przeglądowe nt. zespołów klasyfikatorów

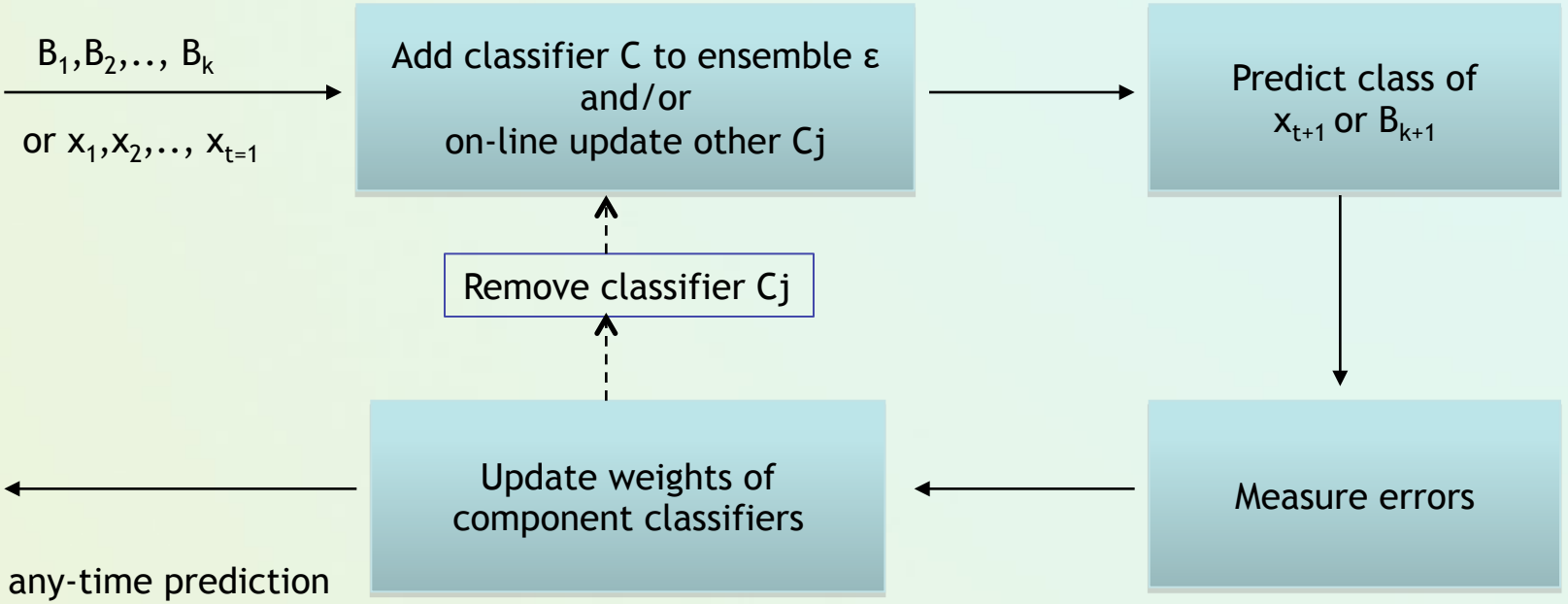
Opublikowane w 2017:

Bartosz Krawczyk, Leandro L. Minku, Joao Gama, Jerzy Stefanowski, Michał Wozniak: Ensemble learning for data stream analysis: a survey, Information Fusion. 37 (2017), 132-156.

Wcześniejsze ogólniejsze prace przeglądowe:

Gregory Ditzler, Manuel Roveri, Cesare Alippi, Robi Polikar. Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine, 10(4):12–25, (2015).

Adaptive Approaches



Continuously adapt the ensemble and its parameters

Learn a number of classifiers on different parts of the data

Weigh classifiers according to recent performance

If classifier performance too weak, it is replaced by new classifier

Taxonomy of adaptive ensembles

Block-based ones:

Streaming Ensemble Algorithm (SEA) – Street & Kim 2001

Accuracy Weighted Ensemble (AWE) Wang et al 2003

- BWE - Deckert 2011, Weighted Aging Ensemble (Wozniak et al 2013)

Learn++.NSE – Polikar et al. 2011

Others , e.g., EAE (Jackowski 2014)

Recurring concepts

- CCP – Katakis et al. 2010
- RCD – Goncalvas et al. 2013
- FAE – Diaz et al 2015

Block processing also in some semi-supervised or novel class detection – Masud et al. 2009; Farid et al 2013

On-line (instance based)

WinNow, Weighted Majority Alg. – Littlestone 1988, L & Warmuth 1994

Dynamic Weighted Majority (DWM) – Kolter & Maloof 2003 → AddExp (2005)

On-line bagging and on-line boosting [Oza]

- BagADWIN,
- Leverage bagging – Bifet et al. 2007
- Using in DDD (Minku, Yao)

Hoeffding Option Trees (HOT)

UFFT (Gama at al. 2005)

ADACC – Jaber 2013

Boosting classifiers for drifting concepts – Scholtz & Klinkenberg 2007 + more

Hybrid approaches

ACE – Nishida 2009, OBWE , AUE → OAUE (Brzeziński)

Majority vote algorithm (Littlestone, Warmuth, 1994)

Train the base classifiers in the ensemble *D*1 ,…, *DL*. **(Pick your experts in horse racing)** - they are not re-train in next step

1. Set all weights to 1, $wj = 1/L$, $i = 1, \ldots, L$. Choose $\beta \in [0,1]$.
2. For a new x (new race), calculate the support for each class as the sum of the weights for all classifiers that voted for that class. (Take all expert predictions for your chosen horse. Sum up the weights for those experts that predicted a win and compare with the sum of weights of those who predicted a loss.). Make a decision for the most supported class.
3. Observe the true label of x (did your favourite win?) and update the weights of the classifiers that were wrong using $wj=\beta wj$.
4. Normalize the weights and continue from 2.

Winnow variant [Littlestone]

- Observe the true label of **x** and update the weights of **all** the classifiers **if the ensemble prediction was wrong**.

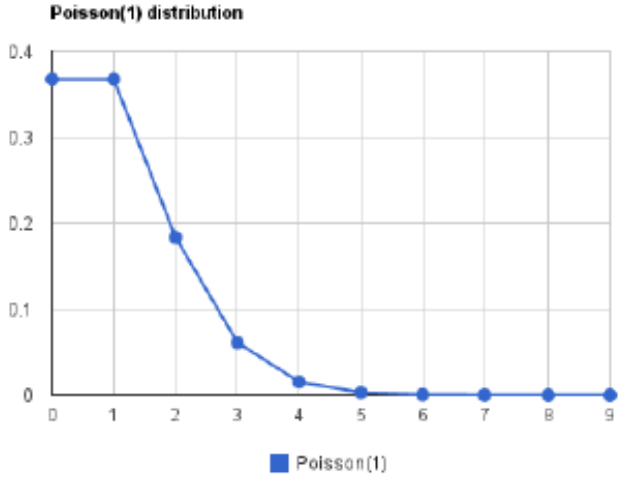
Online Bagging [N.Oza, S.Russel]

What is the rule of example occurrence in a sample?

Example

Dataset of 4 Instances : A, B, C, D

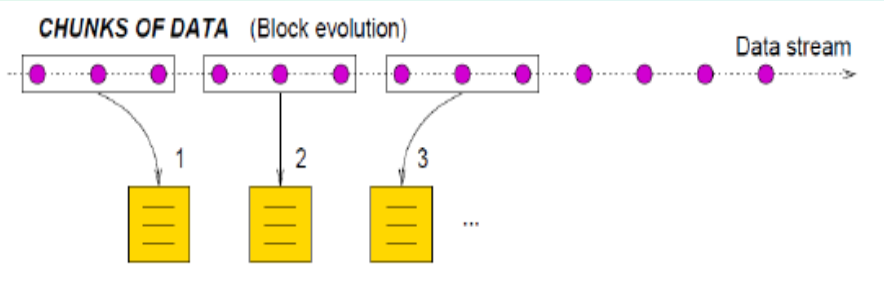
Classifier 1: A, B, B, C: A(1) B(2) C(1) D(0)
Classifier 2: A, B, D, D: A(1) B(1) C(0) D(2)
Classifier 3: A, B, B, C: A(1) B(2) C(1) D(0)
Classifier 4: B, B, B, C: A(0) B(3) C(1) D(0)
Classifier 5: A, C, C, D: A(1) B(0) C(2) D(1)



Each base model's training set contains each of the original training example K times where $P(K = k)$ follows a binomial distribution.

Block-based ensembles

The origins → SEA (Streaming Ensemble Algorithm)



Generic schema

- train *K* classifiers from *K* blocks
- for each subsequent chunk-block B_i

  train a new component classifier

  test other classifiers against the recent block

  assign weight to each classifier

  select top *K* classifiers (remove the weaker classifiers)

Some advantages:

- When examples come in blocks (chunks)
- Use static learning algorithms
- May have smaller computational costs than on-line ensembles

Accuracy Weighted Ensemble - AWE

H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining Concept-Drifting Data Streams using Ensemble Classifiers", KDD'03.

Idea: **Weight classifiers according to the current data distribution**

- Formal proof that classifiers weighted this way are equally or more accurate than single classifiers built upon all
- Weights approximated by computing the classification error on the most recent data block (chunk)

$$w_{ij} = MSE_r - MSE_{ij}, MSE_{ij} = \frac{1}{|B_i|} \sum_{\{\mathbf{x},y\} \in B_i} (1 - f_y^j(\mathbf{x}))^2, MSE_r = \sum_y p(y)(1 - p(y))^2$$

- The new candidate – 10-fold cross validation on the latest block
- Remove classifiers with weights smaller than MSE_r
  - Maintain a buffer of some pruned classifiers
- Originally used with J48 (classical non-incremental) trees / also other static learners

Limits of block-based algorithms

- Accuracy is highly dependent on data block B_i size
  → needs experimental efforts to tune
- Too slow reactions to sudden concept drifts
  - Small block size may help,
    but not for stability periods and increases computational costs
- Sudden concept drifts can sometimes mute all base classifiers
- Component classifiers often trained only once, never change

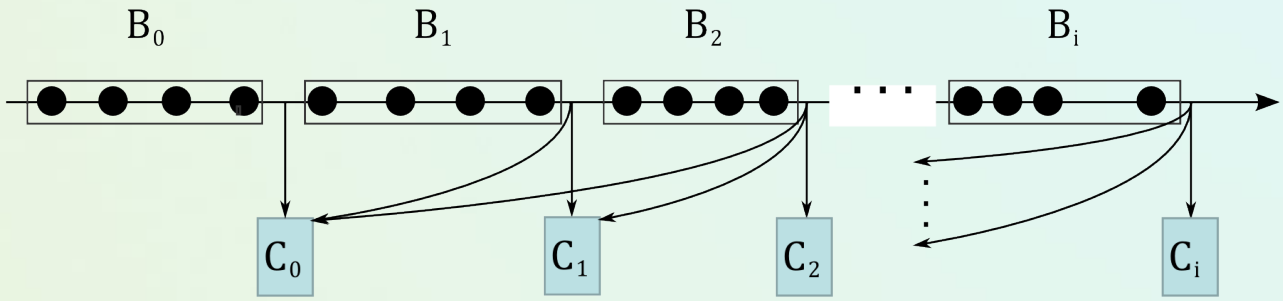
Refer to on-line ensembles:

- They react better to sudden drifts but worse to gradual drifts
- Component classifiers update over time
- However, usually more computationally costly

Our approaches to stream ensembles

- Hypothesis
  - ***Could we combine best properties of both (block, on-line) approaches to sufficiently adapt to several types of concept drifts with satisfactory memory and time?***
- Our proposals
  - Block-based AUE
  - On-line OAUE
- Strong co-operation with Dariusz Brzeziński

Accuracy Updated Ensemble - Motivations



- Keep the block schema of constructing new classifiers, substituting the worst ones, periodical evaluations of components (weighting)
- Incremental updating of component classifiers
  - Improves reaction to various drifts, and reduces the influence of the block size B_i
- Analysis of changes in weighting component classifiers and the role of the new introduced classifier
  - Additional reductions of computational costs

Accuracy Updated Ensemble - AUE

- Incremental component classifiers (Hoeffding Trees,…)
- New weighting of classifiers
  - Non-linear weights
    (better differentiate classifiers
    and resign from extra pruning with MSE_r)

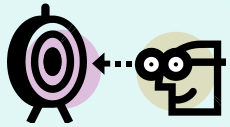
$$w_{ij} = \frac{1}{MSE_{ij} + MSE_r + \varepsilon}$$

- Newest classifier treated as the best one (MSE_{ij}=0) – gets a highest weight and always substitute the worst classifier
- Reducing computational costs
  - Resign from an extra buffer of classifiers
  - Manage memory limits → prune trees
  - No extra cross-validation for a newest classifier

D. Brzezinski, J. Stefanowski: **Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm**. IEEE Transactions on Neural Networks and Learning Systems, 25 (1), 81-94 (2014).

D.Brzeziński, JStefanowski: Accuracy updated ensemble for data streams with concept drift, Proceeding of HAIS 2011.

Experimental evaluation



15 Datasets

11 synthetic and 4 real ones

45 000 - 10 000 000 instances

Different drift scenarios

- incremental, gradual, sudden, recurring, mixed, blips, no drift
- Fast vs. slow rate

MOA generators →

Hyperplane, RBF, SEA, Tree, LED

AUE → Implementation in MOA

Base classifiers → Hoeffding Tree with NB leaf predictions (n_{min}=100, δ=0.01, φ=0.05)

Evaluate:

→ time, memory, predictive accuracy

Table 3.1: Characteristic of datasets

Dataset	Instances	Attributes	Classes	Noise	Drifts	Drift type
Hyp_S	1M	10	2	5%	1	incremental
Hyp_F	1M	10	2	5%	1	incremental
RBF_B	1M	20	4	0%	2	blips
RBF_{GR}	1M	20	4	0%	4	gradual
RBF_{ND}	1M	20	2	0%	0	none
SEA_S	1M	3	4	10%	3	sudden
SEA_F	2M	3	4	10%	9	sudden
$Tree_S$	1M	10	4	0%	4	sudden recurring
$Tree_F$	100k	10	6	0%	15	sudden recurring
LED_M	1M	24	10	10%	3	mixed
LED_{ND}	10M	24	10	20%	0	none
Elec	45k	7	2	-	-	unknown
CovType	581k	53	7	-	-	unknown
Poker	1M	10	10	-	-	unknown
Airlines	539k	7	2	-	-	unknown

Component analysis of AUC:

AUC with *K* = 10 classifiers; Tested *K* = 5,.., 40

Block size d = 500 instances; (experim. up to 4000)

Some other experimental studies → (k=7, d=300)

Best accuracy - update all components for drift datasets

Details in : D.Brzezinski, Block-based and on-line ensembles for concept-drifting data streams, Ph.D. Thesis, 2015 + IEEE TNNLS 2014

Comparative Study

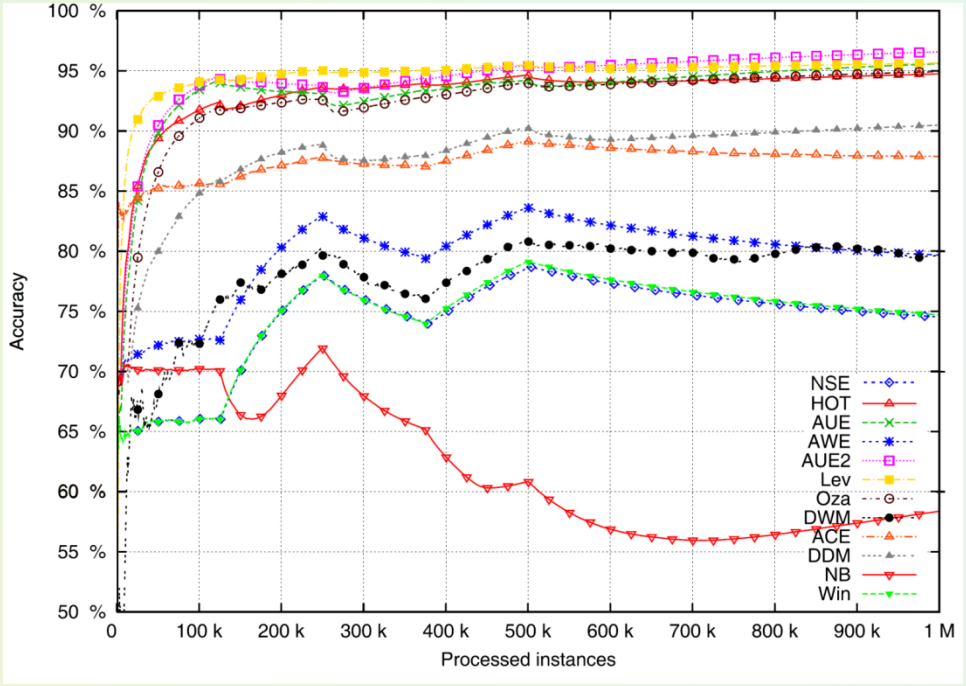
AUE compared against 11 other algorithms:

- Accuracy Weighted Ensemble (AWE)
- Hoeffding Option Trees (HOT)
- Adaptive Classifiers Ensemble (ACE)
- AUE (+ its previous 2011 version)
- Online Bagging, Leveraging Bagging
- Dynamic Weighted Majority (DWM)
- Learn^{++}.NSE
- Single HT with a window (Win)
- Naive Bayes (NB)

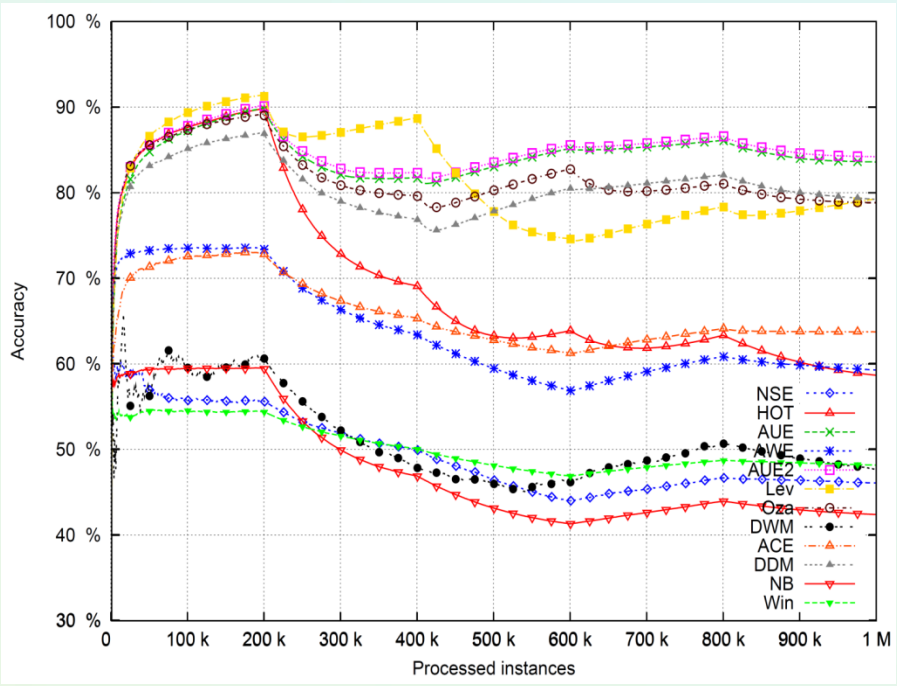
Mostly MOA implementations (D. Brzezinski),
ACE and Learn^{++}.NSE adapted from other versions

- The same 15 datasets
  - Evaluate: time, memory, predictive accuracy

Reacting to different changes



Classification accuracy on RBF_{GR} (slow, gradual changes)



Classification accuracy for $Tree_S$ dataset (fast, sudden)

Comparative study – classification accuracy

Ranks in Friedman test

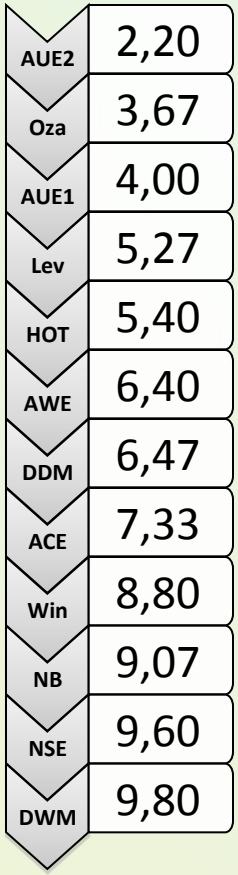
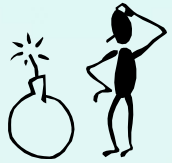


Table 1: Average classification accuracy

Data	ACE	AWE	AUE1	AUE2	HOT	DDM	Win	Lev	NB	Oza	DWM	NSE
Hyp_S	80,65	**90,43**	88,59	88,43	83,23	87,92	87,56	85,36	81,00	89,89	71,20	86,83
Hyp_F	84,56	89,21	88,58	**89,46**	83,32	86,86	86,92	87,21	78,05	89,32	76,69	85,39
RBF_B	87,34	78,82	94,07	94,77	93,79	88,30	73,07	**95,28**	66,97	93,08	78,11	73,02
RBF_GR	87,54	79,74	93,37	94,43	93,24	87,99	74,67	**94,74**	62,01	92,56	77,80	74,49
RBF_ND	84,74	72,63	92,42	**93,33**	91,20	87,62	71,12	92,24	72,00	91,37	76,06	71,07
SEA_S	86,39	87,73	89,00	**89,19**	87,07	88,37	86,85	87,09	86,18	88,80	78,30	86,23
SEA_F	86,22	86,40	88,36	**88,72**	86,25	87,80	85,55	86,68	84,98	88,37	79,33	85,07
Tree_S	65,77	63,74	84,35	**84,94**	69,68	80,58	50,15	81,69	47,88	81,67	51,19	49,37
Tree_F	45,97	45,35	**52,87**	45,32	40,34	42,74	41,54	33,42	35,02	43,40	29,30	33,90
LED_M	64,70	67,11	67,29	67,58	66,92	67,17	65,52	66,74	67,15	**67,62**	44,43	62,86
LED_ND	46,33	51,27	50,68	51,26	51,17	51,05	47,07	50,64	**51,27**	51,23	26,86	47,16
Elec	75,83	69,33	70,86	77,32	**78,21**	64,45	70,35	76,08	73,08	77,34	72,43	73,34
Cov	67,05	79,34	81,24	85,20	**86,48**	58,11	77,19	81,04	66,02	80,40	80,84	77,16
Poker	67,38	59,99	60,57	66,10	74,77	60,23	58,26	**82,62**	58,09	61,13	74,49	59,56
Airlines	66,75	63,31	63,92	**67,37**	66,18	65,79	64,93	63,10	66,84	66,39	61,00	63,83

Block to online transformation: Why



- AUE – block-based with incremental elements:
  - Improved reaction to various types of drifts
  - 'best' average accuracy + faster and less memory consuming than the most competitive ensembles
- On-line learners are of more value in some scenarios
  + in some environments class labels available after each example (but on-line ensembles may be more costly)
- Block-based inspirations:
  - Component evaluation and their weighting,
  - Ensemble periodically updated with a new *candidate* classifier trained on last *d* examples
- Could these inspirations + AUE be adapted to work in online environments?

  + with acceptable computational costs!

Block to online transformation: How

- We modify a generic block based training schema:
  - Weight classifiers, remove the worst
  - Keep periodically adding a new *candidate* classifier trained on last *d* examples
- Three **transformation** strategies:
  - Windowing technique
  - Additional online (instance) ensemble member
  - Drift detector with on-line classifier

More → D. Brzezinski, J. Stefanowski, 2012. From Block-based Ensembles to Online Learners in Changing Data Streams: If- and How-To. Proc. ECML PKDD 2012 Workshop on Instant Interactive Data Mining.

D. Brzezinski, J. Stefanowski, 2014. Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, 265, 50-67.

Online Accuracy Updated Ensemble

- Advantages of periodical adaptation mechanisms and on-line classifiers with weighting after each example

  → minimize computational costs of the transformation strategies

Basic characteristics of a new online examples

- On-line learning classifiers
- Periodical adding a new classifiers with highest weight
- No blocks → sliding window
- New weighting function → on-line more efficient per example

Moreover,

- Perform comparable to state-of-the-art algorithms, wrt. accuracy, memory and time

More → D. Brzezinski, J. Stefanowski, 2014. Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, 265, 50-67.

Non-linear weight calculated after each example

$$MSE_i^t = \begin{cases} MSE_i^{t-1} + \frac{e_i^t}{d} - \frac{e_i^{t-d}}{d}, & t - \tau_i > d \\ \frac{t - \tau_i - 1}{t - \tau_i} \cdot MSE_i^{t-1} + \frac{e_i^t}{t - \tau_i}, & 1 \leq t - \tau_i \leq d \\ 0, & t - \tau_i = 0 \end{cases}$$

$$e_i^t = (1 - f_{iy}^t(\mathbf{x}^t))^2$$

$$MSE_r^t = \begin{cases} MSE_r^{t-1} - r^{t-1}(y^t) - r^{t-1}(y^{t-d}) + r^t(y^t) + r^t(y^{t-d}), & t > d \\ \sum_y r^t(y), & t = d \end{cases}$$

$$r^t(y) = p^t(y)(1 - p^t(y))^2$$

$$w_i^t = \frac{1}{MSE_r^t + MSE_i^t + \epsilon}$$

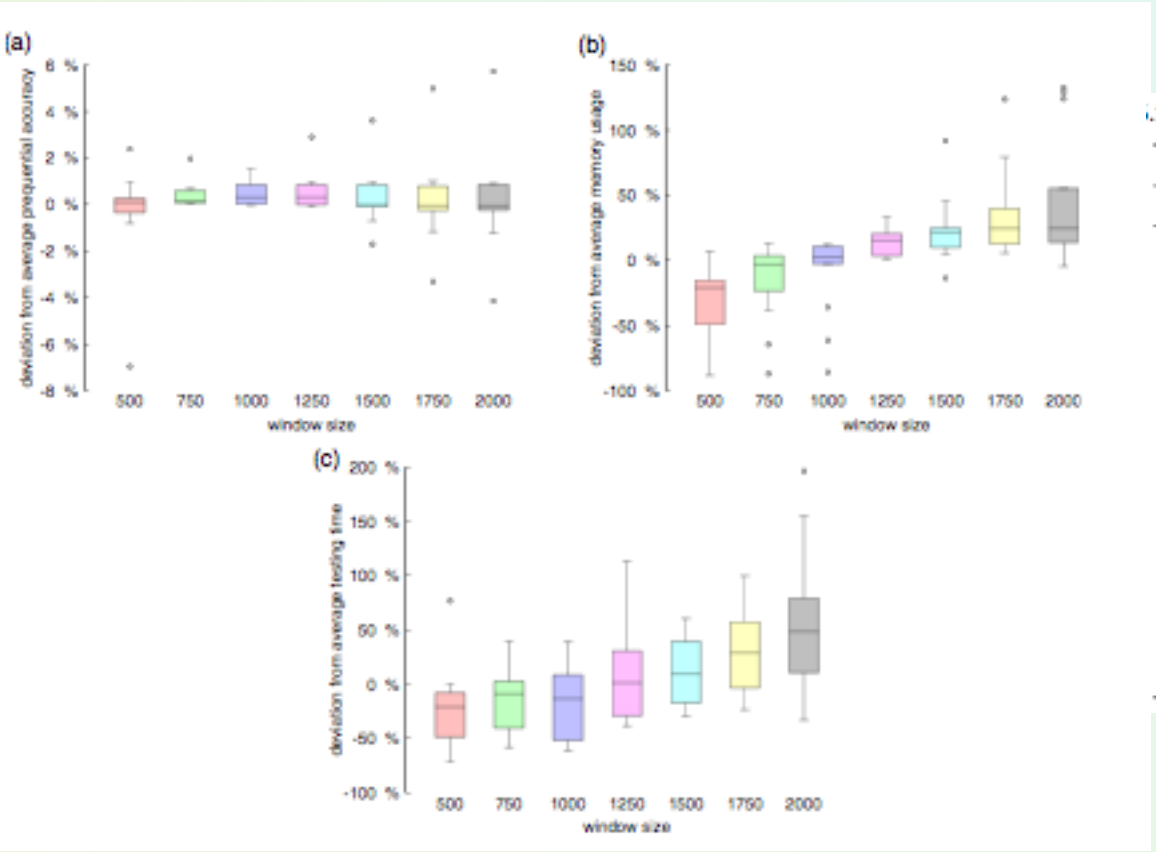
Experimental analysis

- 5 on-line algorithms: ACE, DWM, Lev, Bag, OAUE
- 15 datasets → the same as before
- Different types of drifts
- Evaluation wrt: time, memory, and accuracy

- Studying the impact of OAUE elements
  - Different size of sliding windows
    - Window size *d* → no impact on accuracy, but time and memory are proportional to it
  - Non-linear vs. linear weighting functions

    → Linear one better on fastest drifting streams

Sliding window *d* in OAUE



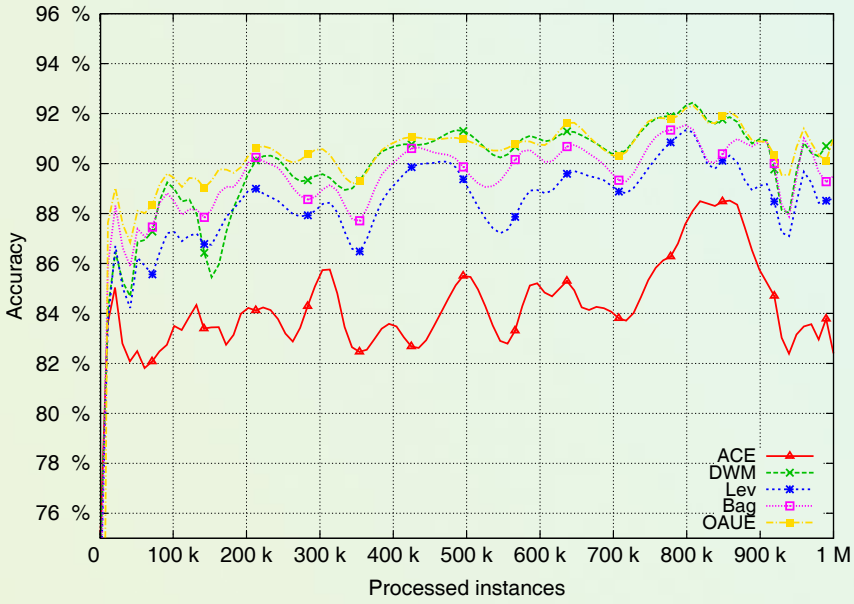
.2: Average prequential accuracy [%] of OAUE for different window

	Window size						
	500	750	1000	1250	1500	1750	2000
Airlines	67.50	66.93	67.03	67.12	66.72	66.33	66.23
CovType	90.07	90.85	90.91	91.08	91.43	91.51	91.58
Hyper_F	90.55	90.43	90.42	90.26	90.30	90.24	90.19
Hyper_S	89.05	89.04	88.94	89.00	88.98	88.92	88.97
LED_M	53.40	53.40	53.38	53.24	52.65	52.40	52.38
LED_{ND}	51.54	51.48	51.40	51.39	51.35	51.27	51.28
PAKDD	80.24	80.23	80.23	80.20	80.20	80.20	80.17
Poker	81.54	87.92	88.87	90.18	90.81	92.01	92.65
Power	15.73	15.58	15.54	15.34	15.27	15.23	14.87
RBF_B	96.78	97.59	97.83	97.84	97.96	98.00	97.90
RBF_{GR}	96.69	97.27	97.38	97.46	97.56	97.53	97.43
SEA_G	88.95	88.85	88.81	88.79	88.70	88.67	88.62
SEA_S	89.41	89.32	89.31	89.28	89.23	89.22	89.15
Tree_{SR}	46.23	46.05	45.86	45.21	44.39	43.66	43.28
Wave	84.34	85.25	85.47	85.58	85.53	85.50	85.49
Wave_M	83.86	84.75	84.85	84.87	84.86	84.73	84.66

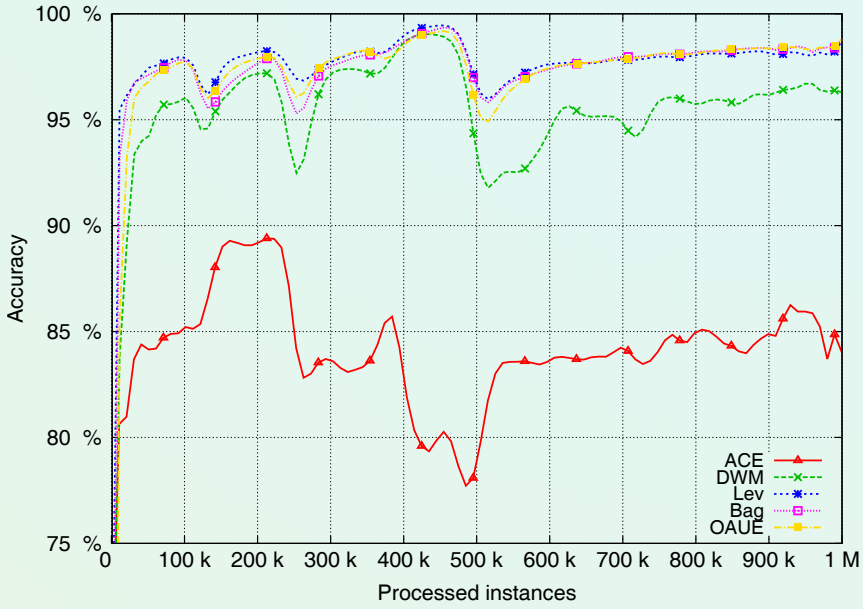
Window size *d* → no impact on accuracy, but time and memory are proportional

More → D. Brzezinski, J. Stefanowski, 2014. Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, 265, 50-67.

Reacting to different changes

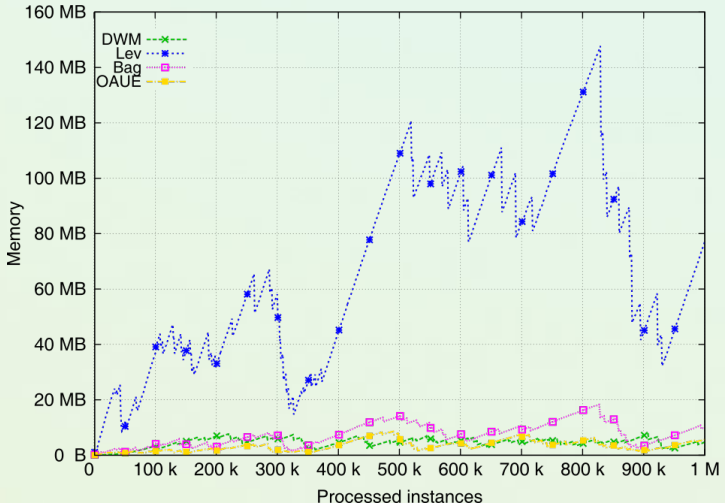


Prequential classification accuracy on Hyper$_F$ (fast, sudden changes)



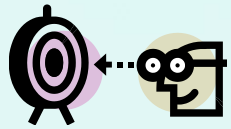
Accuracy on RBF$_{GR}$ (slow, gradual changes)

Memory Hyper$_F$

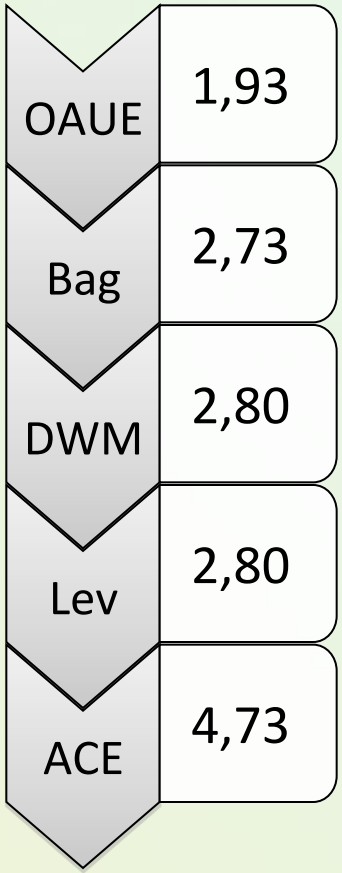


More → D. Brzezinski, J. Stefanowski, Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, 265, (2014) 50-67.

Comparison of on-line classifiers



Ranks in Friedman test



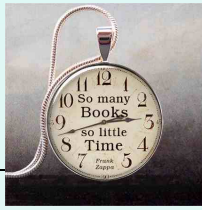
Average prequential classification accuracy

Data	ACE	DWM	Lev	Bag	OAUE
Airlines	64,86	64,98	62,84	64,24	**67,02**
CovType	69,47	89,87	**92,11**	88,84	90,98
Hyp_F	84,34	89,94	88,49	89,54	**90,43**
Hyp_S	79,62	88,48	85,43	88,35	**88,95**
LED_M	46,45	53,34	51,31	53,33	**53,40**
LED_ND	39,80	51,48	49,98	**51,50**	51,48
PAKKDD	-	**80,24**	79,85	80,22	80,23
Poker	79,79	91,29	**97,67**	76,92	88,89
RBF_B	84,78	96,00	**98,22**	97,87	97,87
RBF_GR	84,16	95,49	**97,79**	97,54	97,42
SEA_G	85,97	88,39	**89,00**	88,36	88,83
SEA_S	85,98	89,15	89,26	88,94	**89,33**
Tree_SR	43,39	42,48	47,88	**48,77**	46,04
Wave	-	84,02	83,99	**85,51**	85,50
Wave_M	-	83,76	83,46	**84,95**	84,90

	ACE	DWM	Lev	Bag	OAUE
Memory	--	1.81	3.56	2.6	2.0
Time	2.5	1.81	4.81	3.2	2.6

Ranks in Friedman test

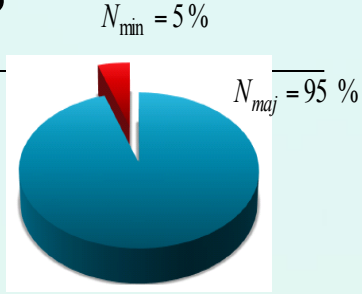
Conclusions → towards on-line



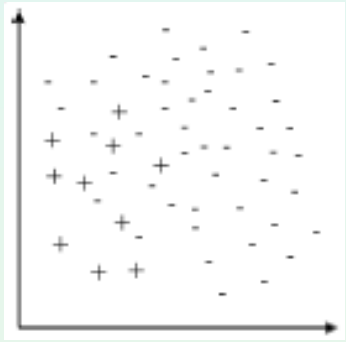
- In environments, where labels are available after each instance
  → the AUC new elements may be insufficient, so …
- Online generalization → OAUE
  - Trains (update) and weight component classifiers with each incoming example
  - Efficient (time & memory) formula for estimating errors
  - Overcome limits of too simple transformation strategies
- Experiments:
  - Parameters (*d*) of AUE → not so influential
  - Comparative study → OAUE provides best averaged classification accuracy with quite good time & memory costs

More: D. Brzezinski, J. Stefanowski: . Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, Volume 265 (2014).

Classifiers for imbalanced data streams



- The minority class underrepresented + data distribution difficulties
- Class imbalance → still a challenge for static machine learning
- Learning from imbalanced, evolving data streams → even more difficult
  - Class imbalances and various drifts
  - Changes of class distributions



- Still less attention in DS, limited research …
  - Uncorrelated bagging [J.Gao et al 2008-2014]
  - The selective recursive approach (SERA) [Chen, He 2009-2014]
  - Extensions of Learn++.CDS → combination concept drift with re-sampling [Polikar, Ditzler 2013]
  - New framework for online learning ensembles from imbalanced streams [Wang, Minku, Yao]
  - …

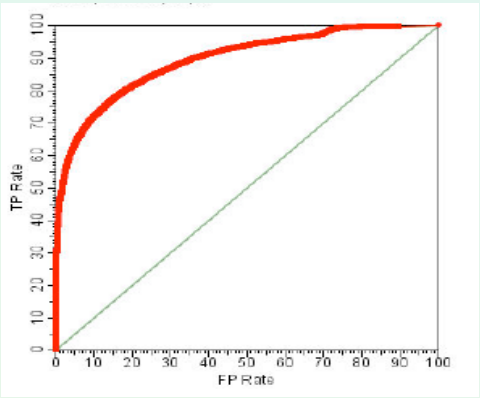
Prequential AUC and its properties

AUC (ROC) → popular for static data / difficulties for streams

New proposal → Prequential AUC

- ❑ Acceptable time and memory (comparing to simpler measures)
- ❑ Good visualization of changes in time
- ❑ Suitable for imbalanced evolving data streams
  - Better indicate concept drifts in (highly) imbalanced data
  - Show changes of the imbalance ratio

Original	Predicted	
	+	-
+	*TP*	*FN*
-	*FP*	*TN*



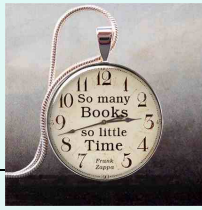
Others

- ❑ Statistically consistent with standard AUC in case stationary data streams
- ❑ Can be used with drift detectors

Research of D. Brzezinski → deeper analysis of PreqAUC properties

More → D. Brzezinski, J. Stefanowski, Prequential AUC for Classifier Evaluation and Drift Detection in Evolving Data Streams. New Frontiers in Mining Complex Patterns ECMLPKDD, 2015.

Open issues



- Specific focused or general purpose techniques for handling drifts
  - Better understanding what forms of drift are handled by each detector or adaptation technique
- Provide insights into changes
  - Interpretability, local vs. global change
- Including additional knowledge in drift adaptations
  - Context or temporal relationships
- Novel class detection or more structural changes
- Detectors for imbalanced changes (also other data changes than the global imbalance ratio)
- Evaluation issues
  - New measures, adaptability, multiple-criteria point of view
  - New testing procedures (controlled permutations, ..)
  - Unavailability of suitable public benchmark data sets

Open issues (2)

Classification challenges

- Availability of ground truth in on-line
  - Delayed labels, obtained on demands, uncertain
- Semi-supervised, unsupervised approaches
- Complex and heterogeneous data representations
- Structured output or specific classification problems

Big Data special requirements

- Need more efficient time- and storage algorithms
- New platforms, e.g. SAMOA project

Knowledge Challenges

- Discover novel knowledge about how a domain evolves
- Understand how things change
- Monitor existing knowledge

Some References

- **J.Stefanowski, 2015, Adaptive Ensembles for Evolving Data Streams--Combining Block-Based and Online Solutions**. Proc. ECML PKDD 2015 Workshop Nfmcp, Spinger LNAI, 3-16.
- D. Brzezinski, J. Stefanowski, 2014. **Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm**. IEEE Transactions on Neural Networks and Learning Systems, Volume 25 (1), 81-94.
- D. Brzezinski, J. Stefanowski, 2014. **Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams**. Information Sciences, Volume 265, 50-67.
- D. Brzezinski, J. Stefanowski, 2013. **Classifiers for Concept-drifting Data Streams: Evaluating Things That Really Matter**. Proc. ECML PKDD 2013 Workshop Real-World Challenges for Data Stream Mining.
- D. Brzezinski, J. Stefanowski, 2012. **From Block-based Ensembles to Online Learners in Changing Data Streams: If- and How-To**. Proc. ECML PKDD 2012 Workshop on Instant Interactive Data Mining.
- D. Brzezinski, J. Stefanowski, 2011. **Accuracy Updated Ensemble for Data Streams with Concept Drift**. Proc. 6th International Conf. HAIS 2011, Part II, Volume 6679 of LNCS, Springer, 155-163.
- D. Brzezinski, J. Stefanowski, 2015. **Prequential AUC for Classifier Evaluation and Drift Detection in Evolving Data Streams**. New Frontiers in Mining Complex Patterns, LNCS Volume 8983, 87-101.
- M. Deckert, J. Stefanowski: **Comparing Block Ensembles for Data Streams with Concept Drift**. In: New Trends in Databases and Information Systems, Springer Comput. Intelligence, vol. 185, 2012, 69-78
- M. Deckert, J. Stefanowski: **RILL: Algorithm for learning rules from streaming data with concept drift**. Proc. ISMIS 2014, vol. 8502 of LNAI, 20-29.
- M.Kmieciak, J.Stefanowski: **Handling Sudden Concept Drift in Enron Message Data Streams**. Control and Cybernetics, vol. 40 no. 3, 2011, 667-695.

Check them at: www.cs.put.poznan.pl/jstefanowski or www.cs.put.poznan.pl/dbrzezinski

Dziękuję za uwagę

Pytania lub komentarze?

Kontakt:
Jerzy.Stefanowski@cs.put.poznan.pl
www.cs.put.poznan.pl/jstefanowski