From Block-based Ensembles to Online Learners in Changing Data streams: If- and How-To

Dariusz Brzezinski and Jerzy Stefanowski

Poznan University of Technology, Poland

From Block Ensembles to Online Learners: If- and How-To

Dariusz Brzezinski and Jerzy Stefanowski

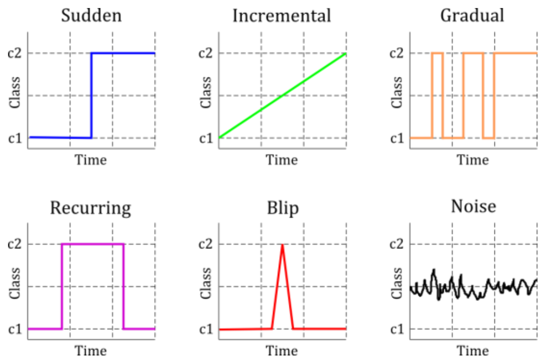
Poznan University of Technology, Poland

Outline

- The problem: from block to online ensembles
- Three strategies
- Experiments
- Conclusions and future work

Data streams with concept drift

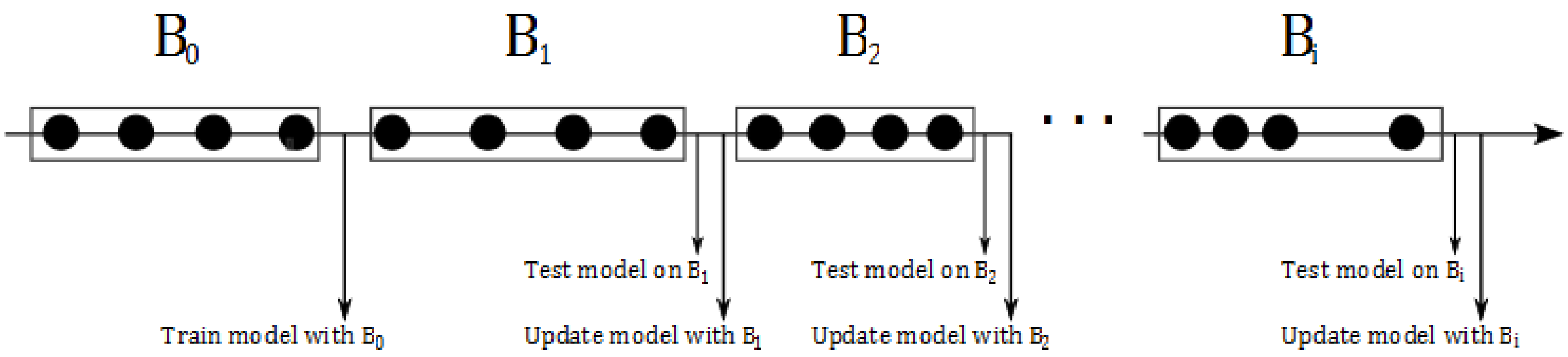
- **Limited time**
  - examples arrive rapidly
  - each example can be processed only once
- **Limited memory**
  - streams are often too large to be processed as a whole
- **Concept drift**
  - data streams can evolve over time
  - many types of concept changes



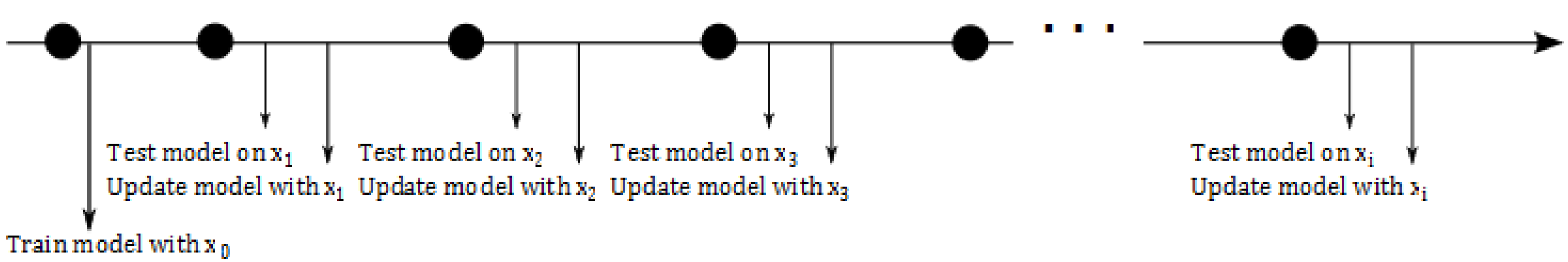
New challenges for data mining algorithms!

Different processing schemes

Block processing



Online processing

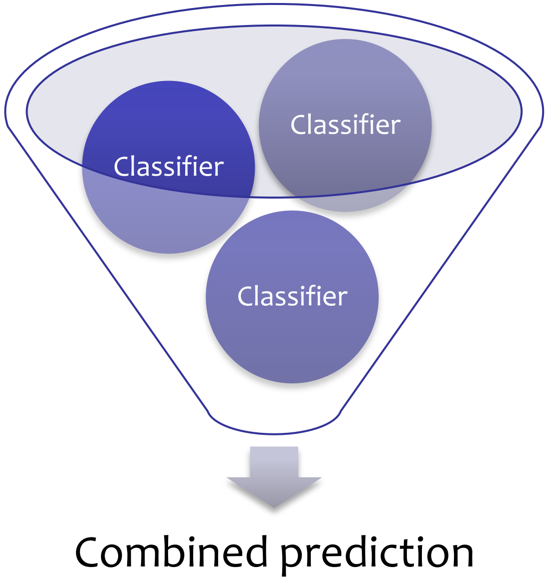


Block to online transformation: Why

- Complementary approaches:
  - Block-based algorithms react well to gradual changes
  - Online algorithms offer quicker reactions to sudden drifts
- Block-based algorithms can be adapted to work in online environments
- Online learners are of more value in most scenarios
- Preliminary results show it’s worth investigating

Block to online transformation: How

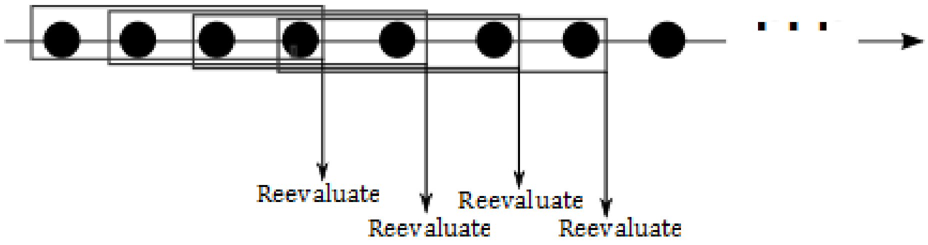
- We focus on certain ensemble methods:
  - Ensembles predict by weighted voting
  - Weights calculated based on classifier performance
  - Ensemble periodically updated with a new *candidate* classifier trained on last *d* examples
- Three **generic** strategies:
  - Windowing technique
  - Additional online ensemble member
  - Drift detector



Strategy 1: Windowing technique

Idea: convert data blocks into sliding windows

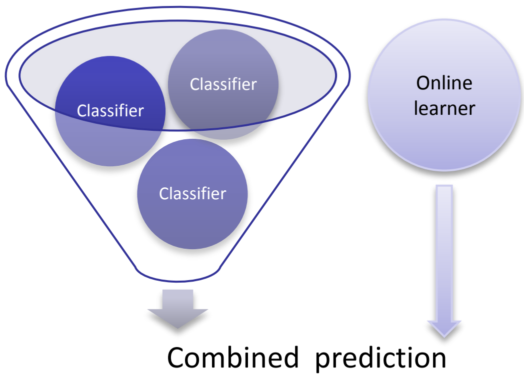
- Component classifiers evaluated and weighted after each example, not every *d* examples
- For efficiency, candidate created every *d* examples
- Online weighting => faster reactions to drift



Strategy 2: Online ensemble member

Idea: introduce an additional online component

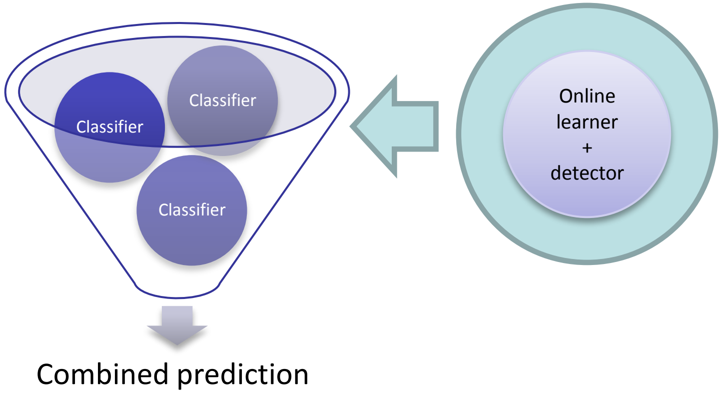
- Online component:
  - has a high weight
  - trained after each example
  - pruned every *d* examples
- Online training => recent data, better prediction



Strategy 3: Drift detector

Idea: react actively to changes in the stream

- Drift detector:
  - incrementally trained
  - forces component retraining when drift is detected
  - reinitialized every *d* example
- Drift detection => fast reactions, quicker retraining



Experimental setup

- 11* algorithms:
  - AWE + 3 modifications
  - AUE + 3 modifications
  - DWM, Online Bagging, ACE
- 8 real datasets
  - 6 artificial and 2 real
  - from 45,000 to 1,000,000 examples
- Different drift scenarios
  - incremental, gradual, sudden, blips, no drift
- Evaluation wrt: time, memory, and accuracy

Results

- The windowing technique improved accuracy of AWE and AUE (2.3%) but at high processing costs (15x)

  => online reweighting is costly but effective
- The online candidate worked for AWE but not AUE

  => the candidate weight should be algorithm-specific
- The drift detector was useful for AUE but not AWE

  => incremental retraining allows chunk size reduction

Results

- The windowing technique improved accuracy of AWE and AUE (2.3%) but at high processing costs (15x)

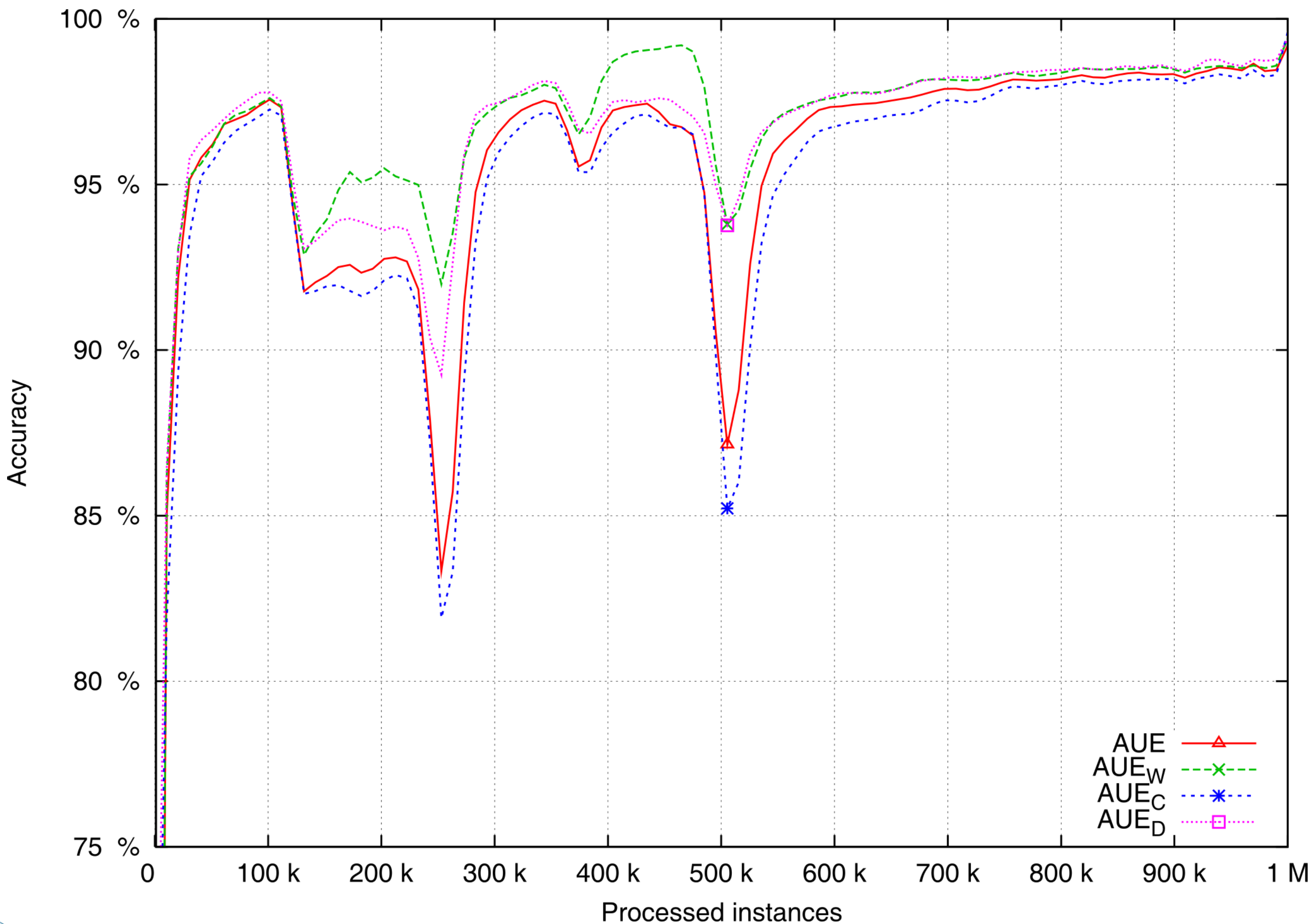
  => online reweighting is costly but effective
- The online candidate worked for AWE but not AUE

  => the candidate weight should be algorithm-specific
- The drift detector was useful for AUE but not AWE

  => incremental retraining allows chunk size reduction

Periodical training and incremental reweighting improved accuracy

Accuracy on the RBF dataset



Results

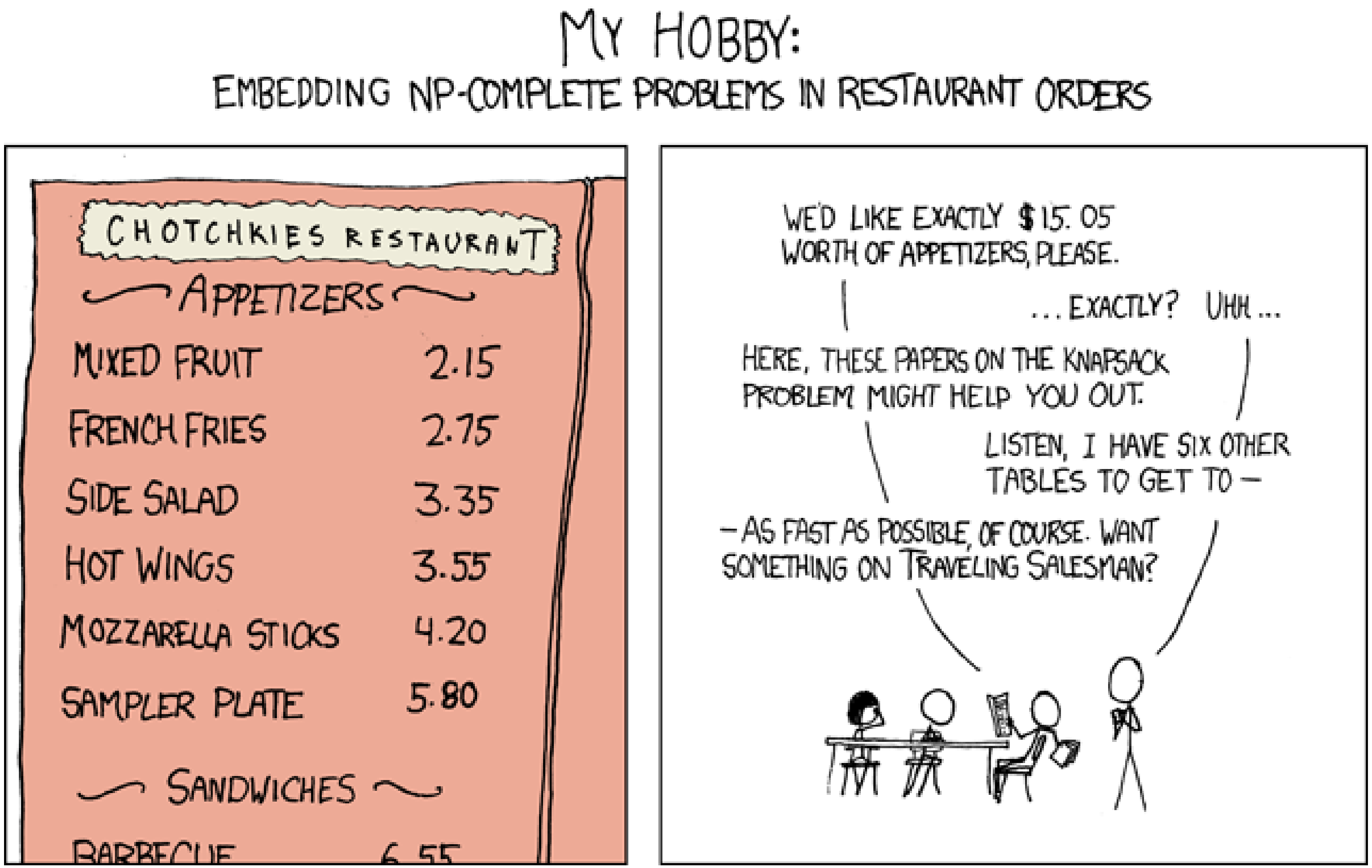
- Proposed modifications were more accurate than DWM and ACE, and comparable to Online Bagging
- The proposed modifications were less memory consuming than Online Bagging
- Fully incremental versions were additionally tested
  - each component updated after *each* example
  - accuracy further improved
  - practically no additional costs

Conclusions

From Block Ensembles to Online Learners

- **If:**
  - It is profitable to retain periodical evaluation and accuracy based weighting in online environments
- **How:**
  - Results obtained by the 3 proposed strategies suggest that components should be incrementally evaluated, reweighted, and trained
  - Algorithm-tailored strategies could be an interesting topic for further reserach

Thank You!



MY HOBBY:
EMBEDDING NP-COMPLETE PROBLEMS IN RESTAURANT ORDERS
CHOTCHKIES RESTAURANT
APPETIZERS
MIXED FRUIT 2.15
FRENCH FRIES 2.75
SIDE SALAD 3.35
HOT WINGS 3.55
MOZZARELLA STICKS 4.20
SAMPLER PLATE 5.80
SANDWICHES
WE'D LIKE EXACTLY $15.05 WORTH OF APPETIZERS, PLEASE.
...EXACTLY? UHH...
HERE, THESE PAPERS ON THE KNAPSACK PROBLEM MIGHT HELP YOU OUT.
LISTEN, I HAVE SIX OTHER TABLES TO GET TO –
– AS FAST AS POSSIBLE, OF COURSE. WANT SOMETHING ON TRAVELING SALESMAN?