Controlled Permutations for Testing Adaptive Classifiers

Indrė Žliobaitė
izliobaite@bournemouth.ac.uk
Lecturer in Computational Intelligence
STRC, Bournemouth University
Date (30/03/2011)

Background

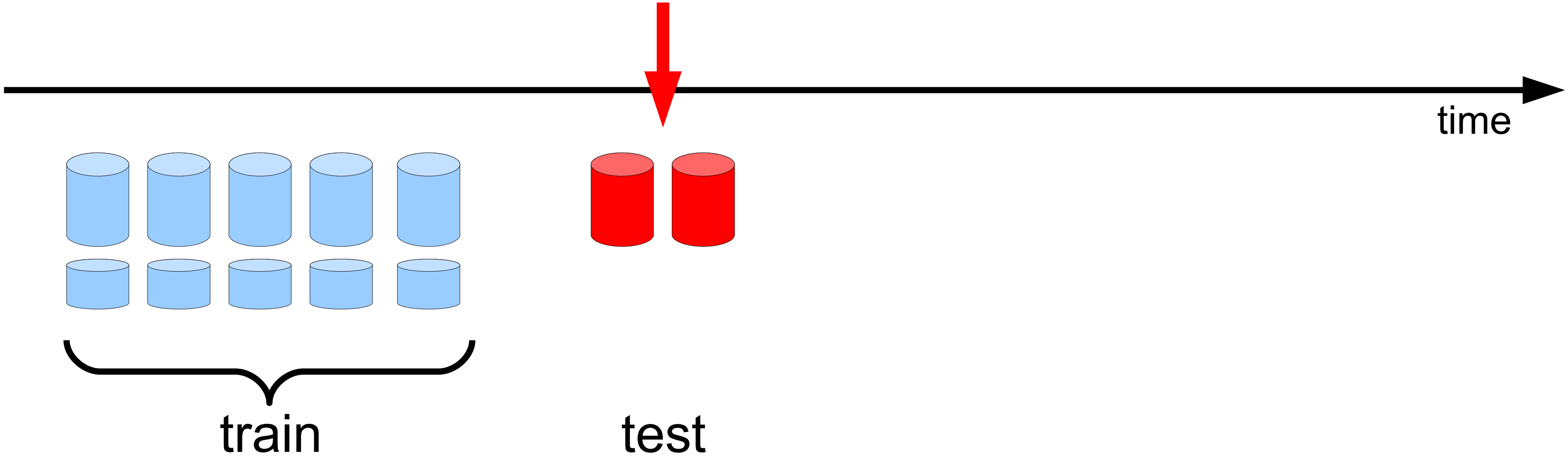
Setting

- Supervised learning
- Learning online (over time)
  - models can be retrained with the incoming data
- Concept drift is expected
  - the data distribution is changing over time
  - models have mechanisms to adapt
- The goal is to optimize the accuracy

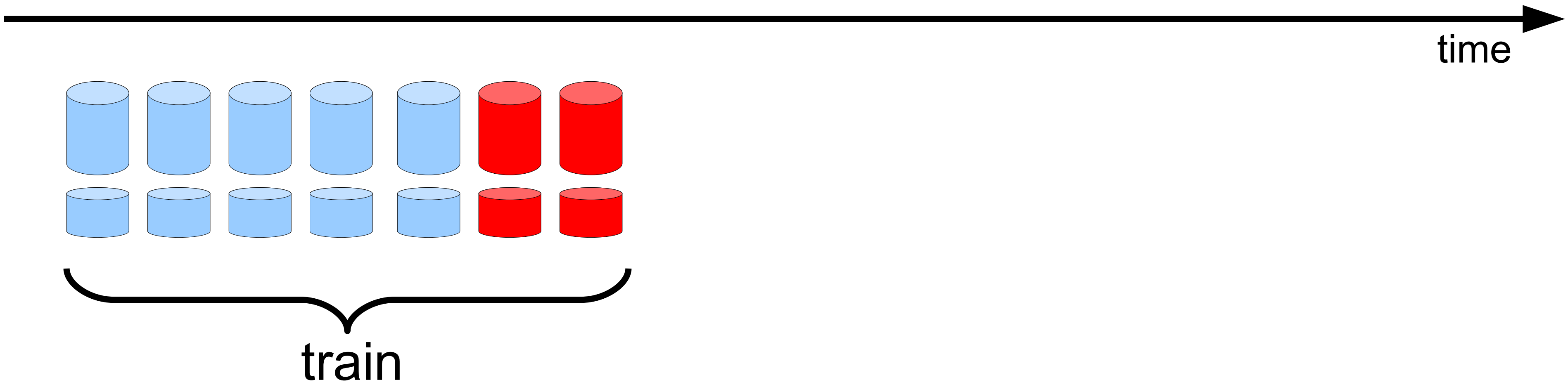
Test-then-train evaluation

- The test-then-train (or prequential) is a standard procedure for testing adaptive classifiers

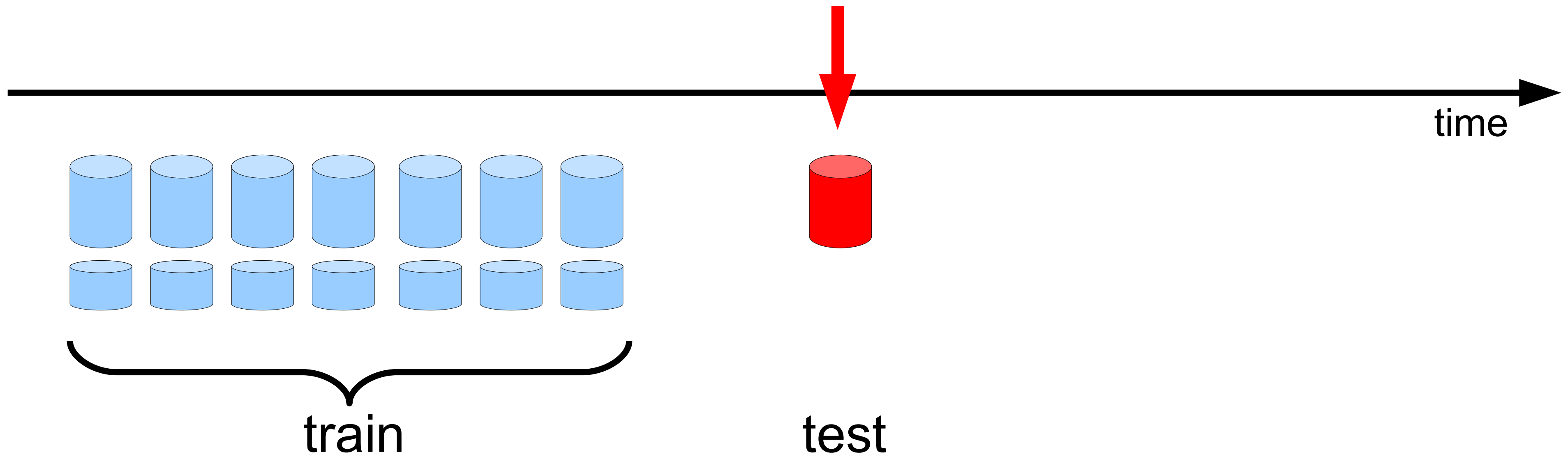
Test-then-train



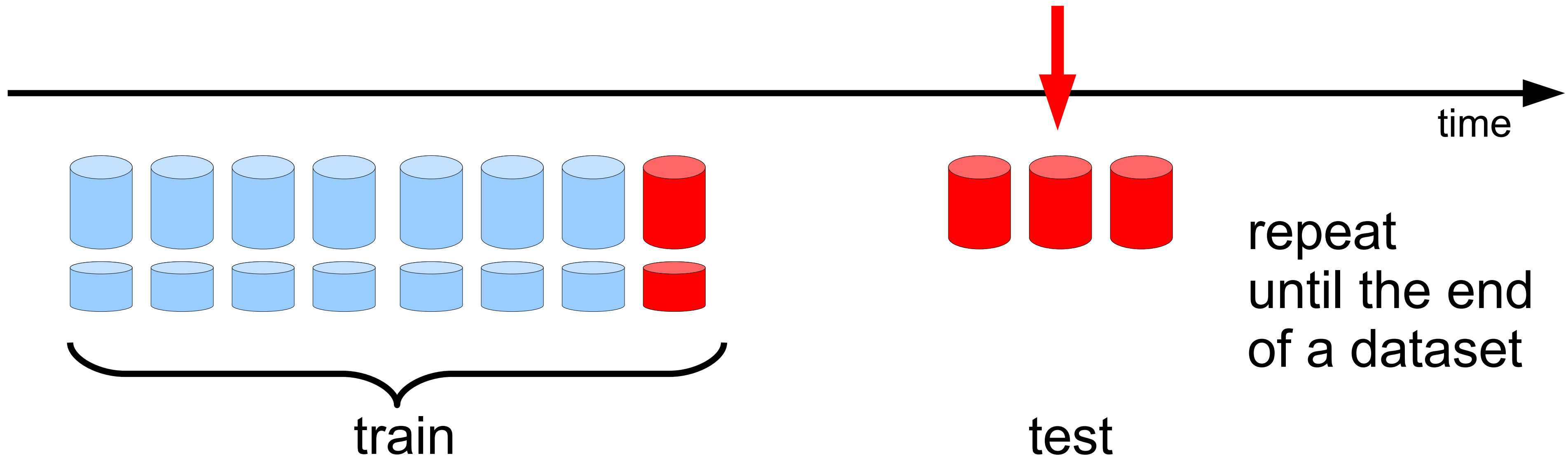
Test-then-train



Test-then-train



Test-then-train



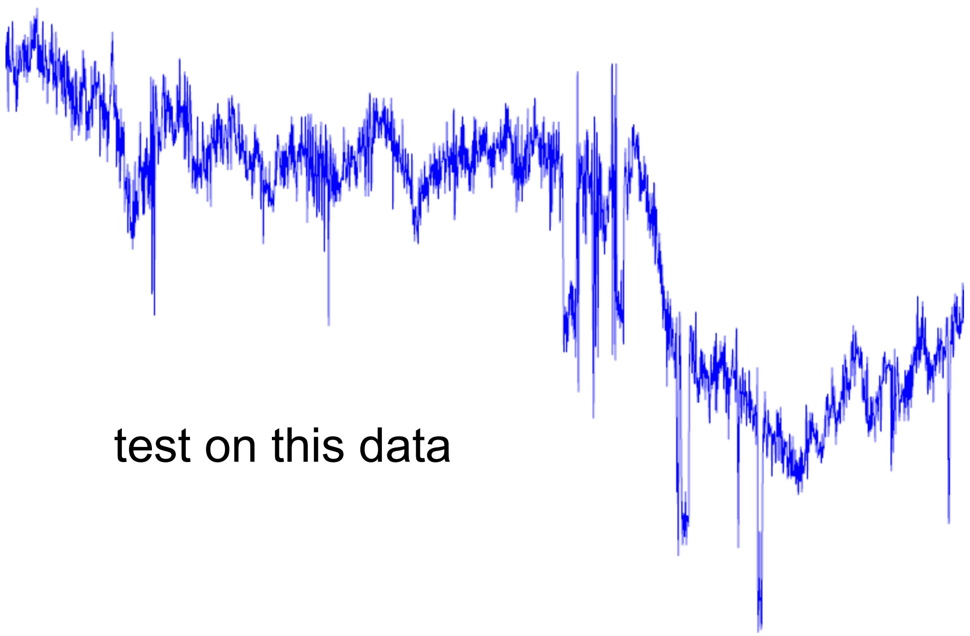
Test-then-train

- The test-then-train (or prequential) is a standard procedure for testing adaptive classifiers
- Passes the dataset in a sequential order once

Test-then-train

- The test-then-train (or prequential) is a standard procedure for testing adaptive classifiers
- Passes the dataset in a sequential order once
- Adaptive classifiers
  - desired: adapt to unexpected changes in general
  - tested: how well they adapt to the fixed configuration of changes

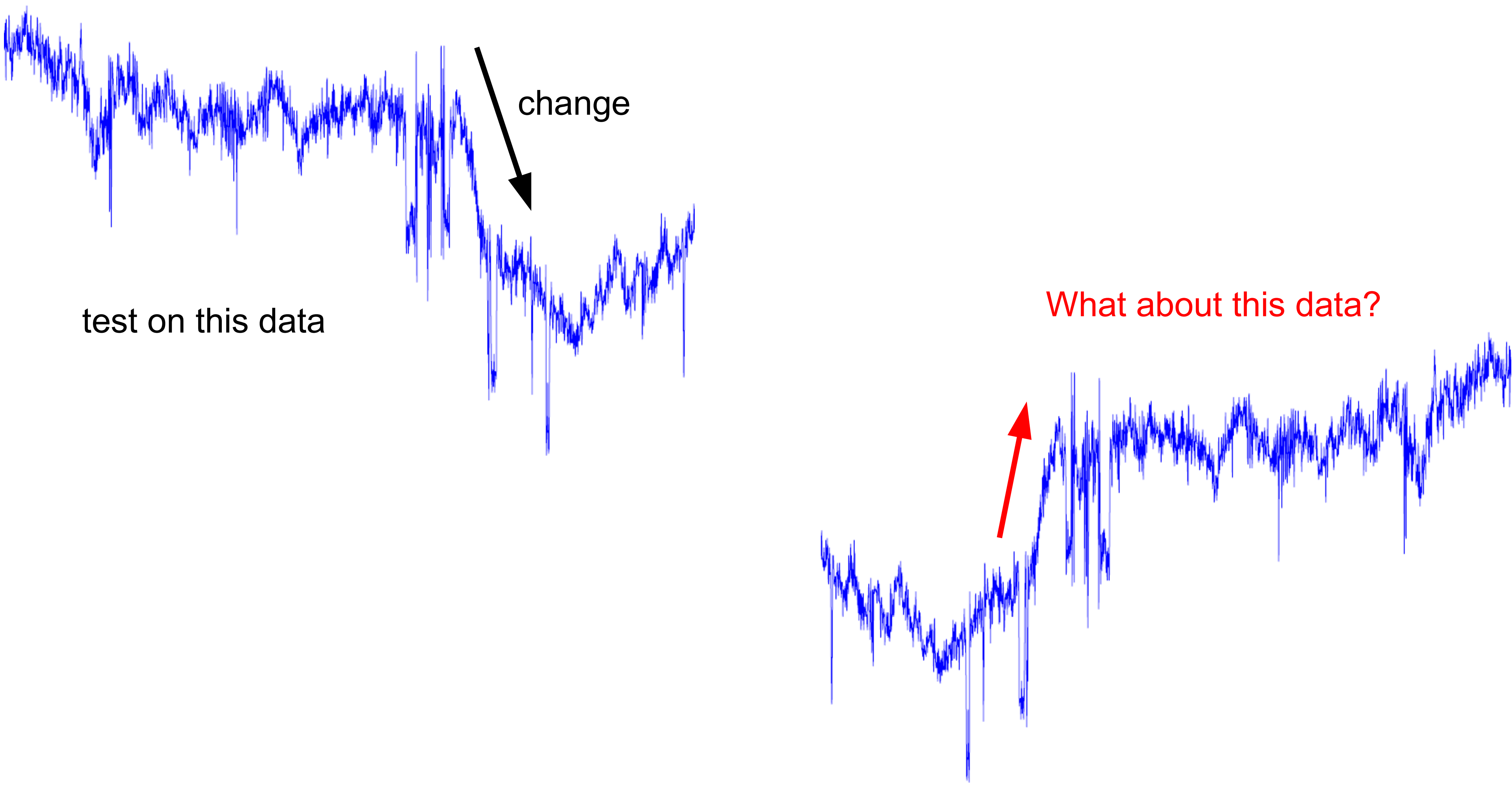
Test-then-train



test on this data

data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Test-then-train



data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Test-then-train

- The test-then-train (or prequential) is a standard procedure for testing adaptive classifiers
- Passes the dataset in a sequential order once
- Adaptive classifiers
  - desired: adapt to unexpected changes in general
  - tested: how well they adapt to the fixed configuration of changes
- The test-then-train risks overfitting

PROBLEM

RISK OF OVERFITTING

Electricity data: an experiment

- Binary classification problem, 44 th. instances
- Original time order ~2 years
- Used as a benchmark for adaptive clasifiers

Original data

Classifier A 78,6%
Classifier B 78,5%

Electricity data: an experiment

Original data

Classifier A **78,6%**
Classifier B 78,5%

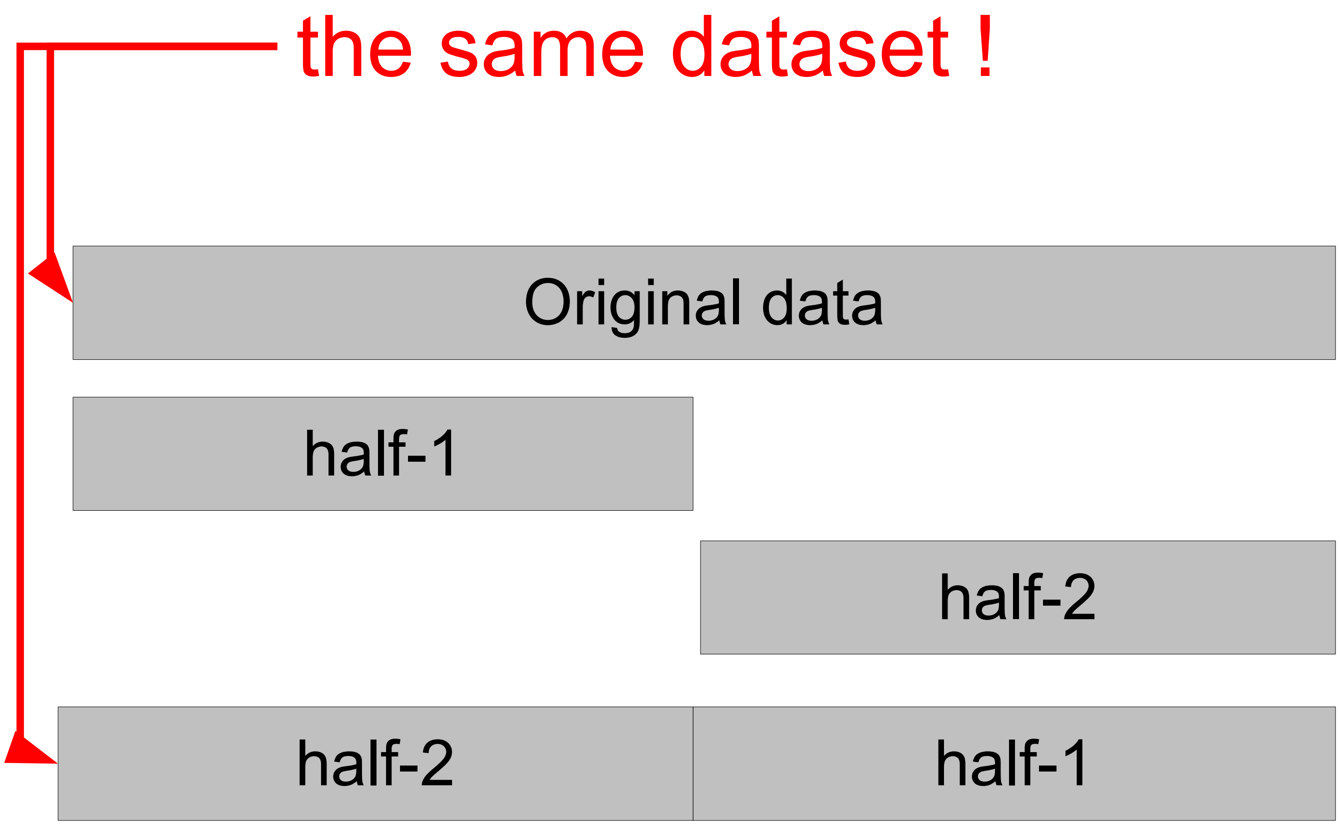
half-1

Classifier A **80,3%**
Classifier B 80,1%

Electricity data: an experiment

Original data	**Classifier A**	**78,6%**
	Classifier B	78,5%
half-1	**Classifier A**	**80,3%**
	Classifier B	80,1%
half-2	Classifier A	76,4%
	Classifier B	**77,2%**

Electricity data: an experiment



Classifier A **78,6%**
Classifier B 78,5%

Classifier A **80,3%**
Classifier B 80,1%

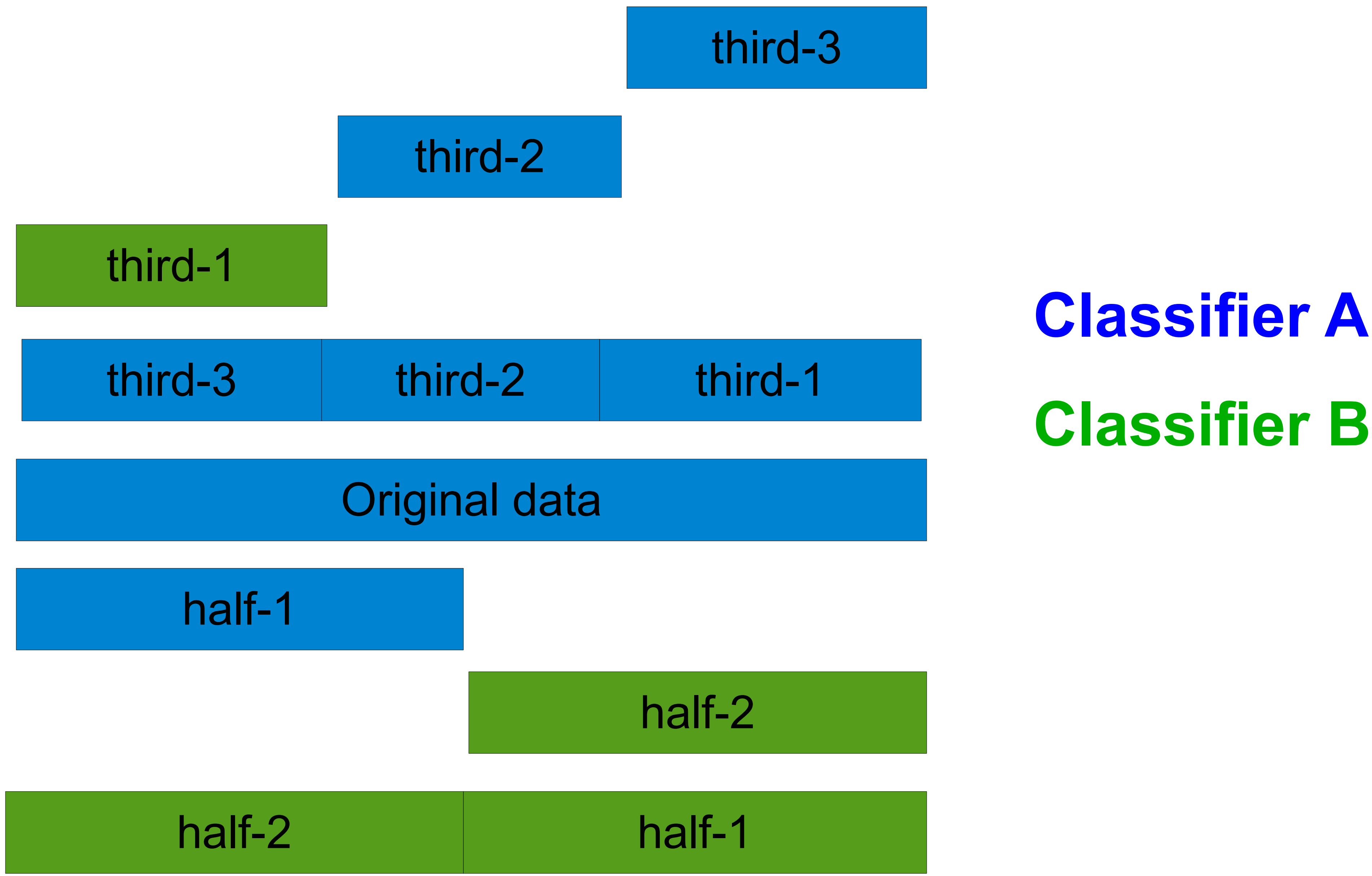
Classifier A 76,4%
Classifier B **77,2%**

Classifier A 78,2%
Classifier B **78,4%**

Electricity data: an experiment

third-3	**Classifier A 78,6%**	Classifier B 77,4%
third-2	**Classifier A 78,0%**	Classifier B 76,2%
third-1	Classifier A 81,3%	**Classifier B 82,1%**
third-3 / third-2 / third-1	**Classifier A 78,6%**	Classifier B 78,4%
Original data	**Classifier A 78,6%**	Classifier B 78,5%
half-1	**Classifier A 80,3%**	Classifier B 80,1%
half-2	Classifier A 76,4%	**Classifier B 77,2%**
half-2 / half-1	Classifier A 78,2%	**Classifier B 78,4%**

Electricity data: an experiment



Electricity data: an experiment

third-1

Classifier A 81,3%
Classifier B 82,1%

slightly
different
snapshots

half-1

Classifier A 80,3%
Classifier B 80,1%

Electricity data: an experiment

third-3

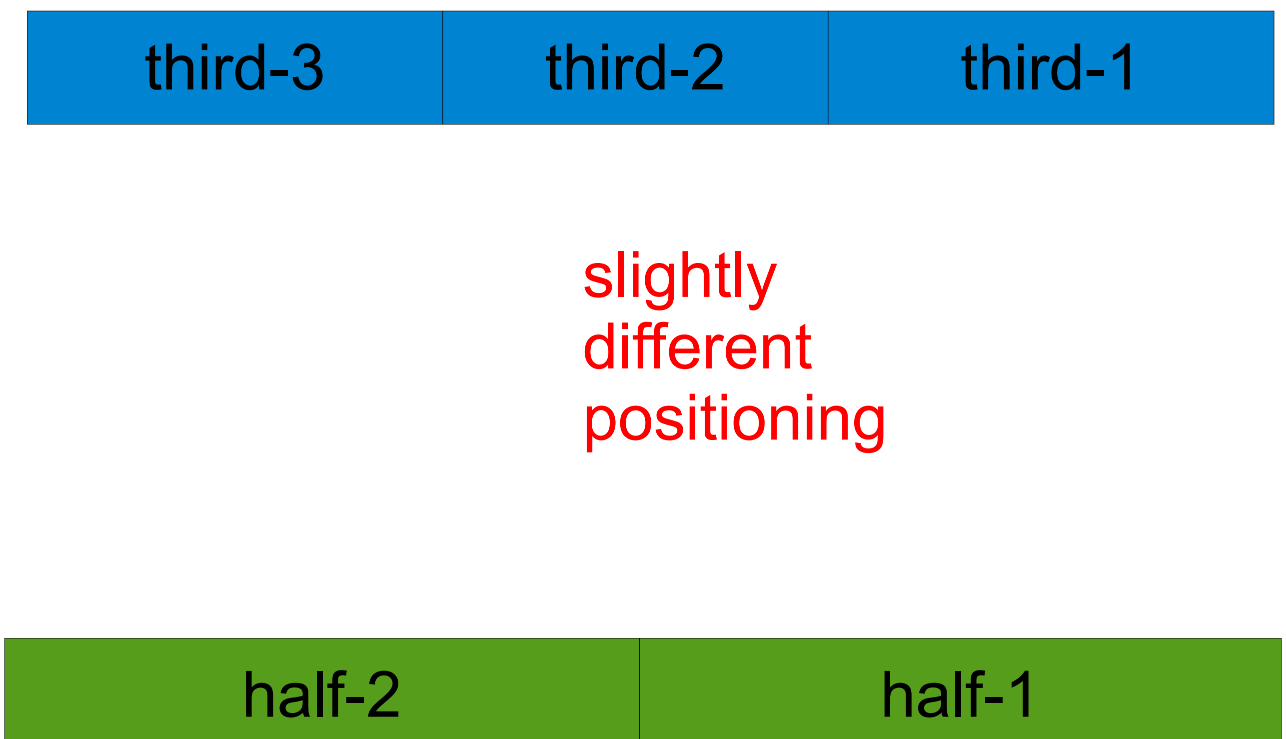
Classifier A 78,6%
Classifier B 77,4%

slightly
different
snapshots

half-2

Classifier A 76,4%
Classifier B 77,2%

Electricity data: an experiment



Classifier A 78,6%
Classifier B 78,4%

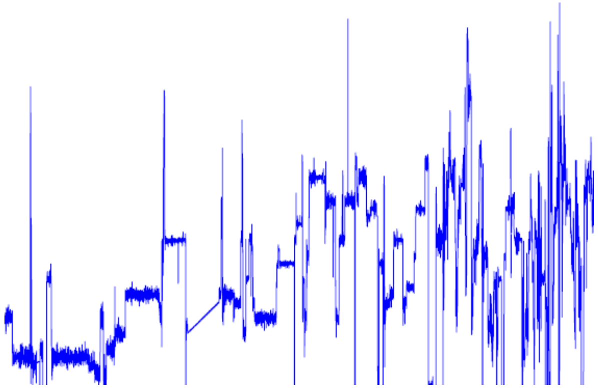
Classifier A 78,2%
Classifier B 78,4%

Why different accuracies?

- Distributions in the two subsamples are different

Why different accuracies?

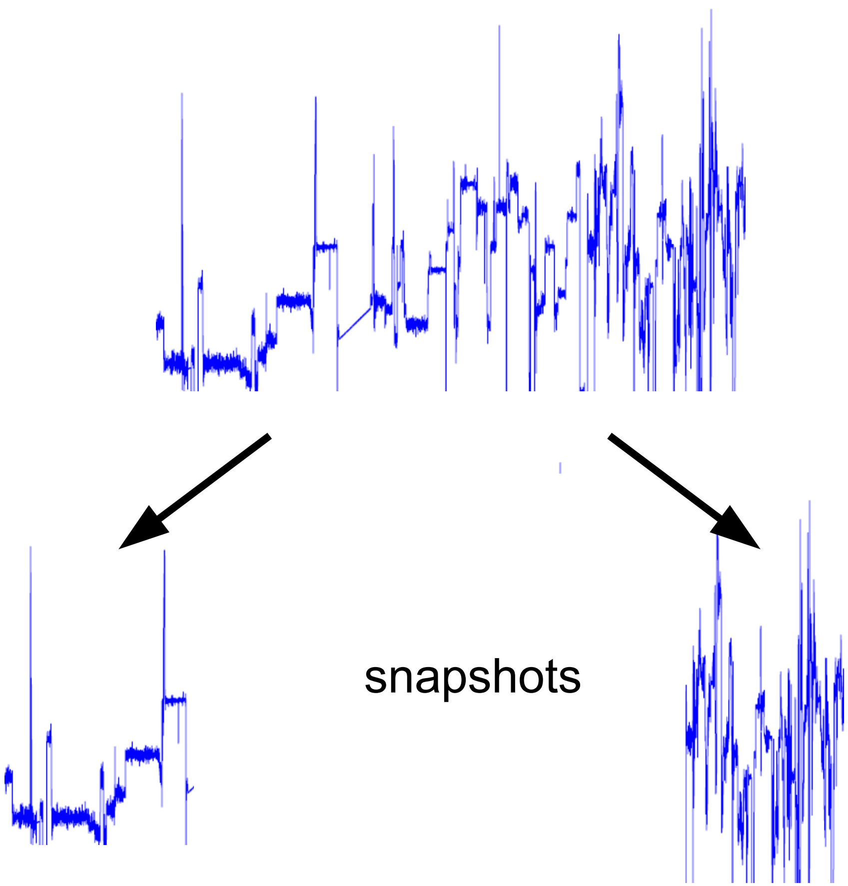
- Distributions in the two subsamples are different



data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Why different accuracies?

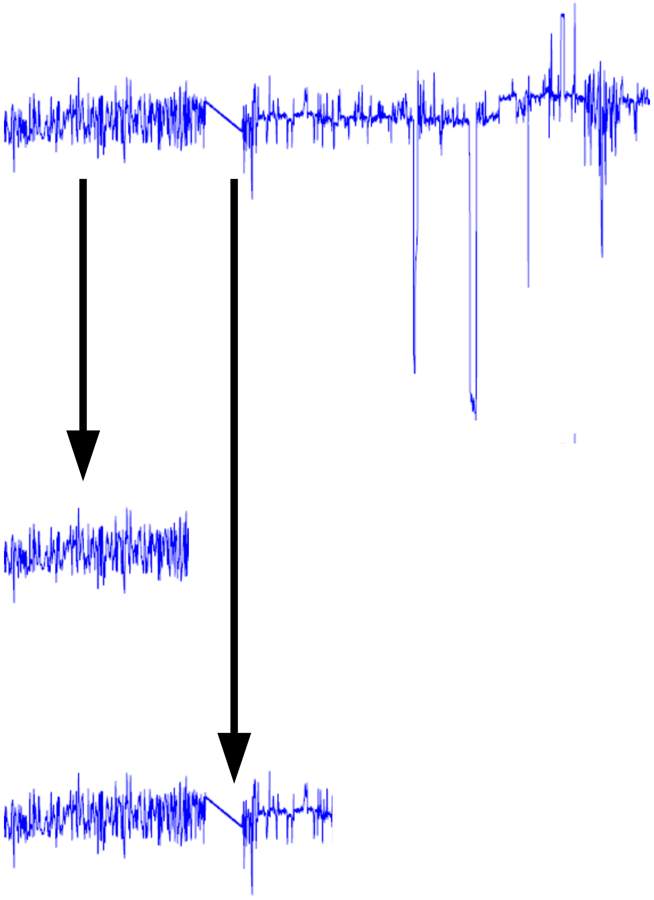
- Distributions in the two subsamples are different



data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Why different accuracies?

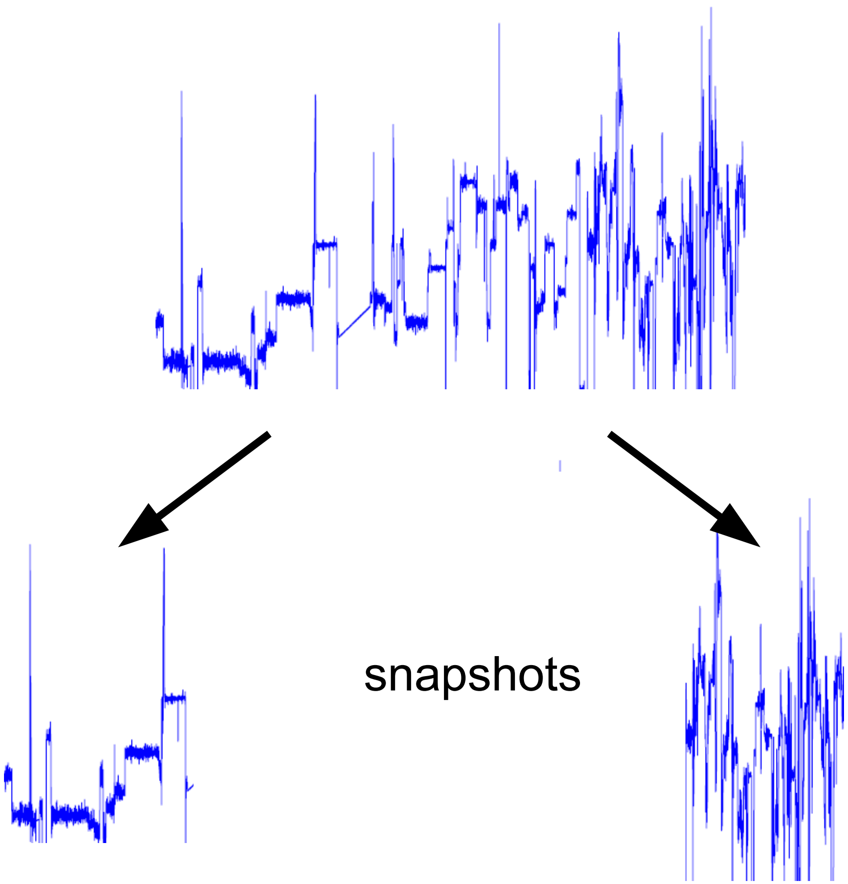
- Distributions in the two subsamples are different

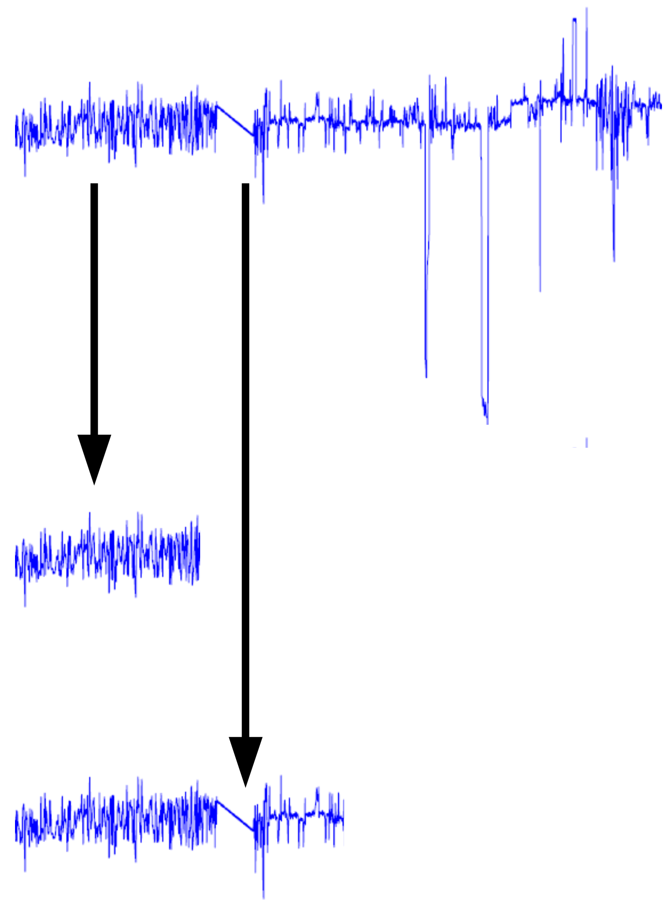


data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Why different accuracies?

- Distributions in the two subsamples are different





data source: P.Kadlec, PhD thesis, Bournemouth University, 2009

Why different accuracies?

- Distributions in the two subsamples are different

- Output of an adaptive classifier depends on the instances seen so far
    - accuracy depends on the order of instances

Why different accuracies?

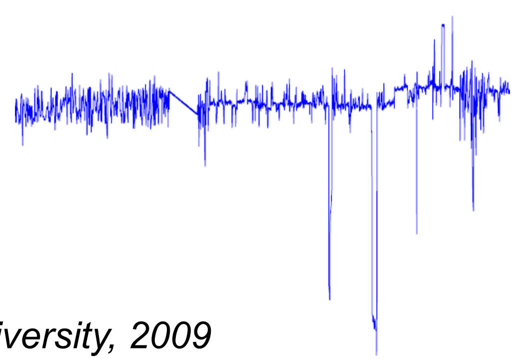
- ~~Distributions in the two subsamples are different~~

  out of the scope

- Output of an adaptive classifier depends on the instances seen so far
  - accuracy depends on the order of instances
  - assume that the distribution in our snapshot represents the distribution of the task well
  - changes in distribution are happening

Why different accuracies?

- Distributions in the two subsamples are different
- Output of an adaptive classifier depends on the instances seen so far
  - accuracy depends on the order of instances
  - assume that the distribution in our snapshot represents the distribution of the task well
  - changes in distribution are happening

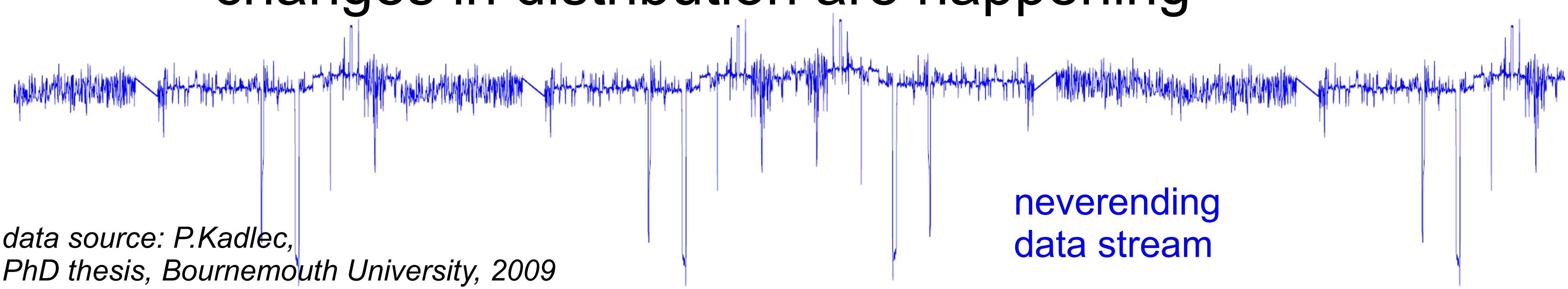


our snapshot

data source: P.Kadlec,
PhD thesis, Bournemouth University, 2009

Why different accuracies?

- Distributions in the two subsamples are different
- Output of an adaptive classifier depends on the instances seen so far
  - accuracy depends on the order of instances
  - assume that the distribution in our snapshot represents the distribution of the task well
  - changes in distribution are happening

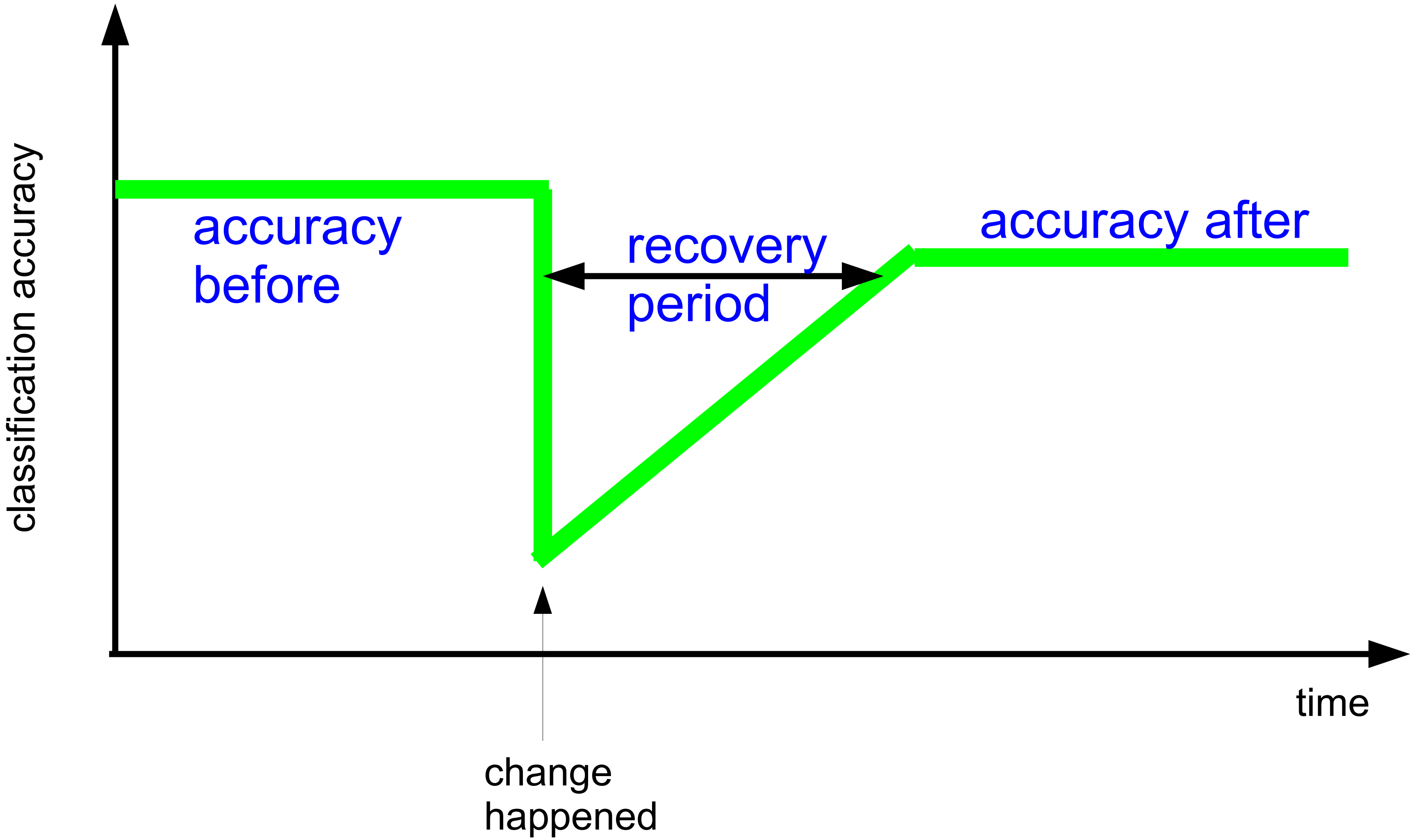


data source: P.Kadlec,
PhD thesis, Bournemouth University, 2009

Why different accuracies?

- Example task
  - Predict a price of a flat x
  - Prices are systematically different at times of ecomonic boom or crisis
  - B: x = 2, C: x = 1
  - Prediction: the average of selected historical prices

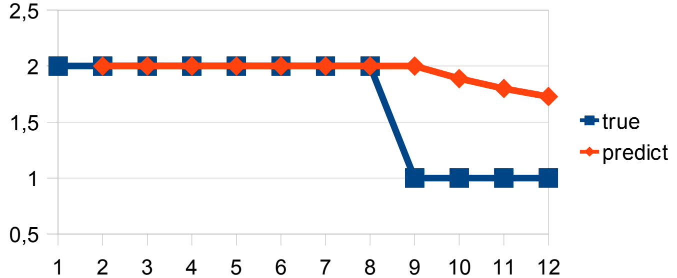
Classifier accuracy under concept drift



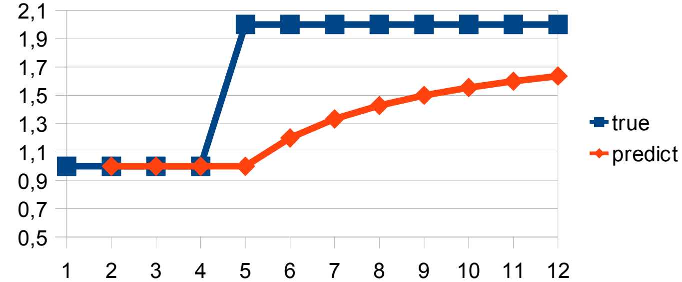
Adaptive learner
Example 1

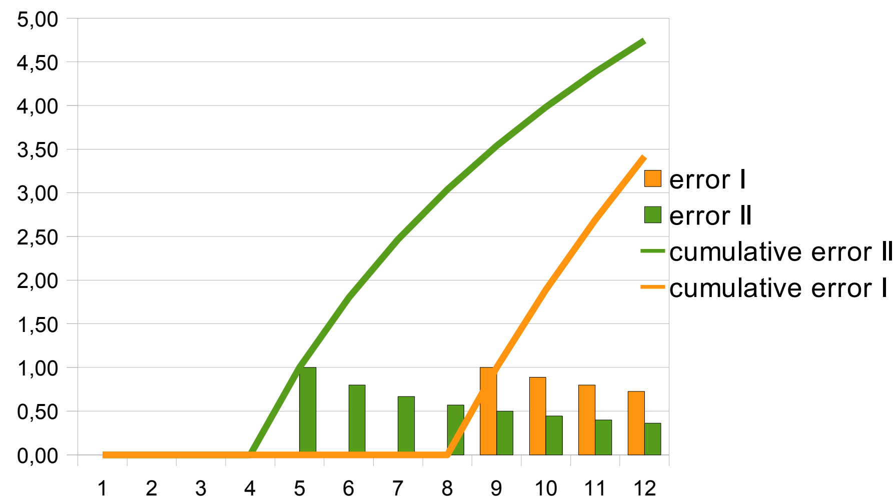
- Naive incremental predictor
  - outputs a mean of all the prices seen so far

BBBBBBBBCCCC



CCCCBBBBBBBB

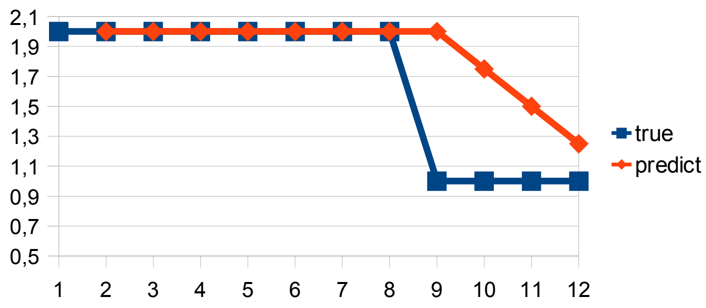




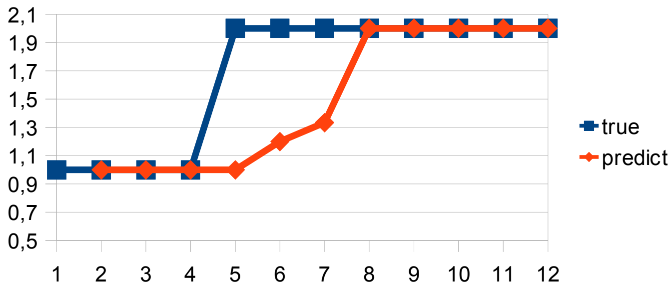
Example 2

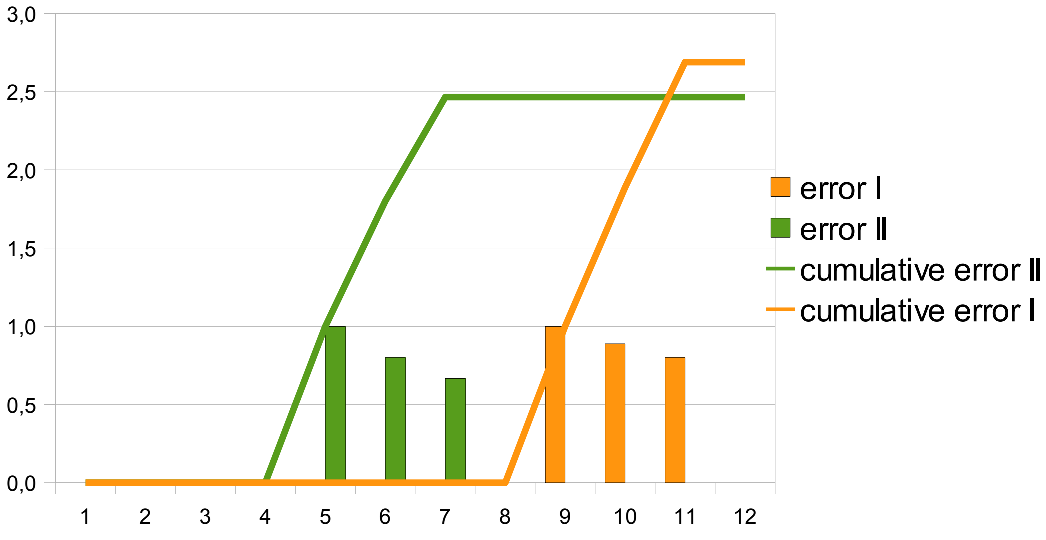
- Detects a change (needs 2 time steps) and discards the old data

BBBBBBBBCCCC



CCCCBBBBBBBB

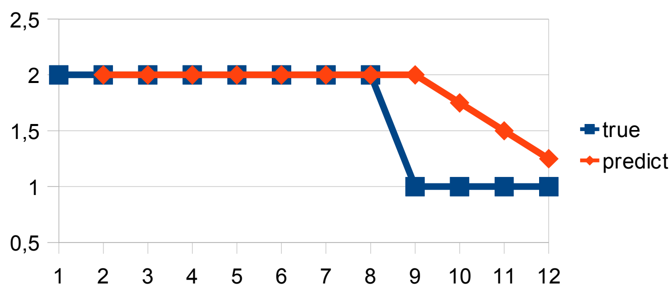




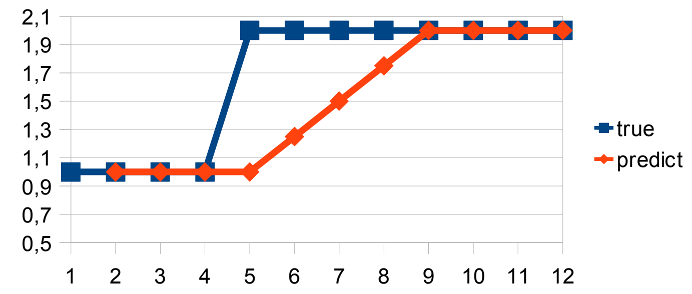
Example 3

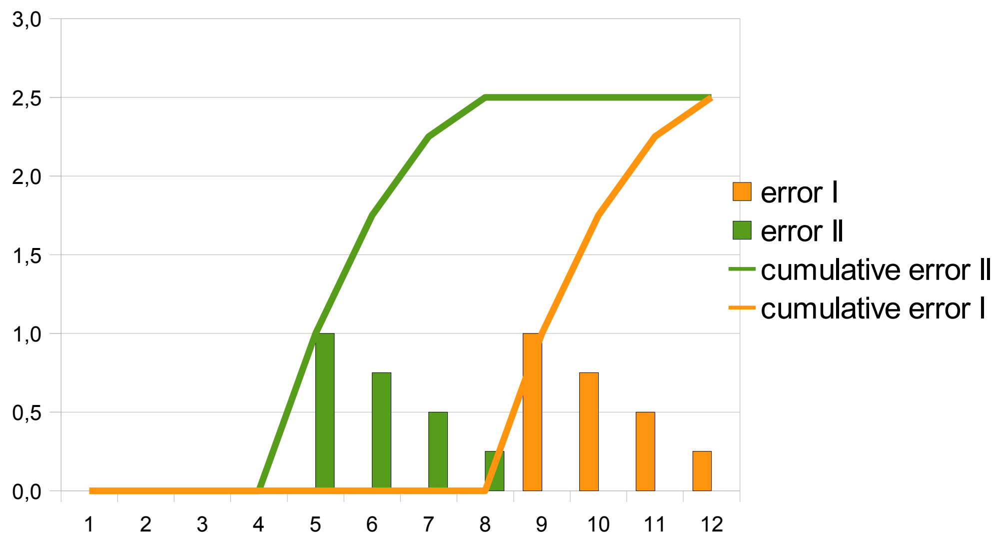
- Constant forgetting rate (4)

BBBBBBBBCCCC



CCCCBBBBBBBB





	BBBBBBBBCCCC	CCCCBBBBBBBB
incremental	3,42	4,75
with detection	2,47	2,69
forgetting	2,50	2,50

- We get different absolute errors
- The data is the same, only the order is different

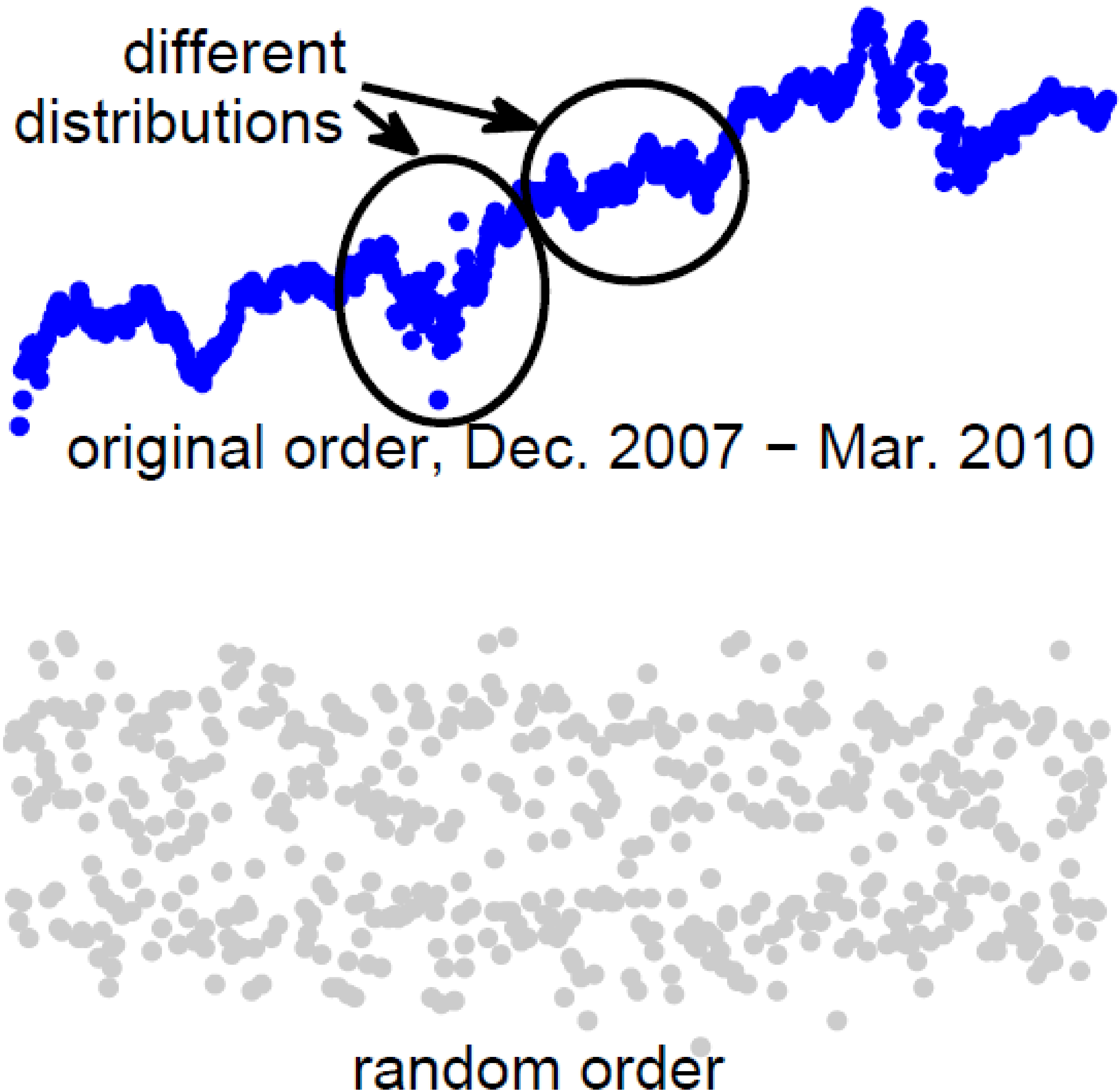
Is it a problem for large datasets?

- For the problem to go away
  - the sequence needs to be very long
  - with only a few distributions
  - and many changes

CONTROLLED PERMUTATIONS

What to do?

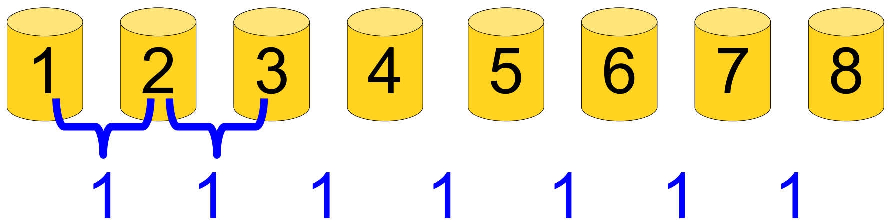
- Do multiple tests to reduce the risk of overfitting
- Random permutations are not suitable
  - they make the data distribution uniform over time


different
distributions
original order, Dec. 2007 − Mar. 2010
random order

What to do?

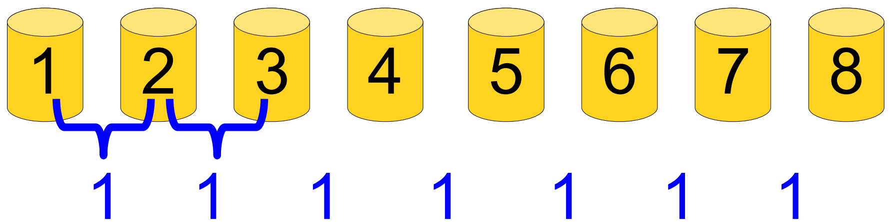
- Do multiple tests to reduce the risk of overfitting
- Random permutations are not suitable
  - they make the data distribution uniform over time
- Solution: controlled permutations
  - preserve local distributions of data
  - the instances that were near
    need to remain close after a permutation

Average Neighbor Distance

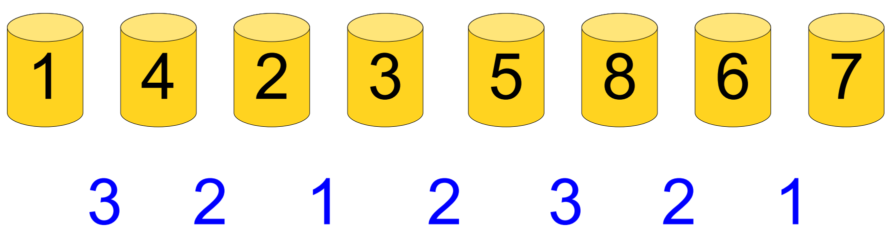


AND = 1

Average Neighbor Distance

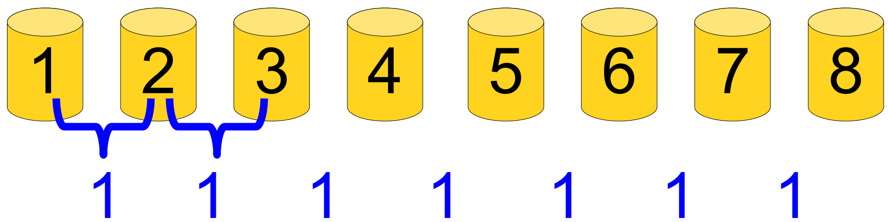


AND = 1

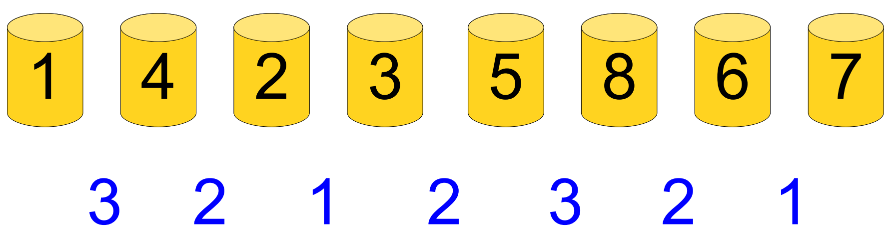


AND = 14/7 = 2

Average Neighbor Distance



AND = 1



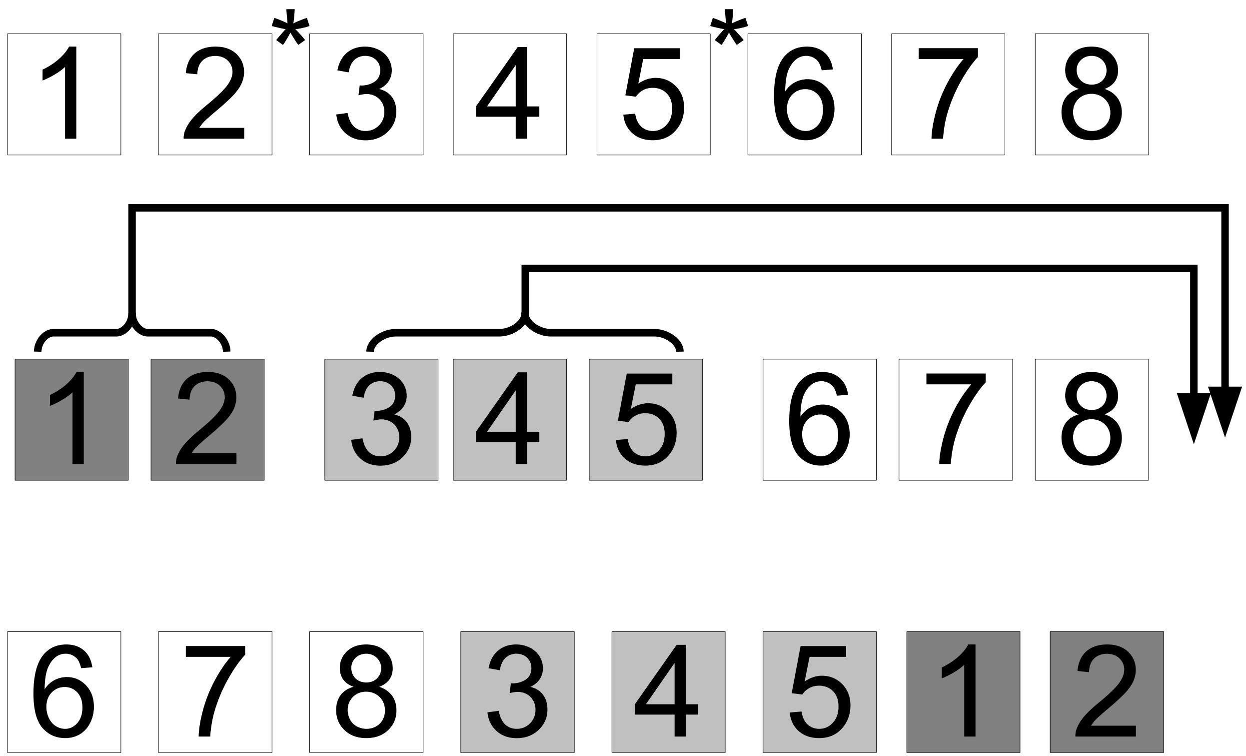
AND = 14/7 = 2

Local distributions are preserved if the expected AND of a permutation is less than random

AND (random) = (n + 1) /3 ~ n

Theoretical result

Time permutation



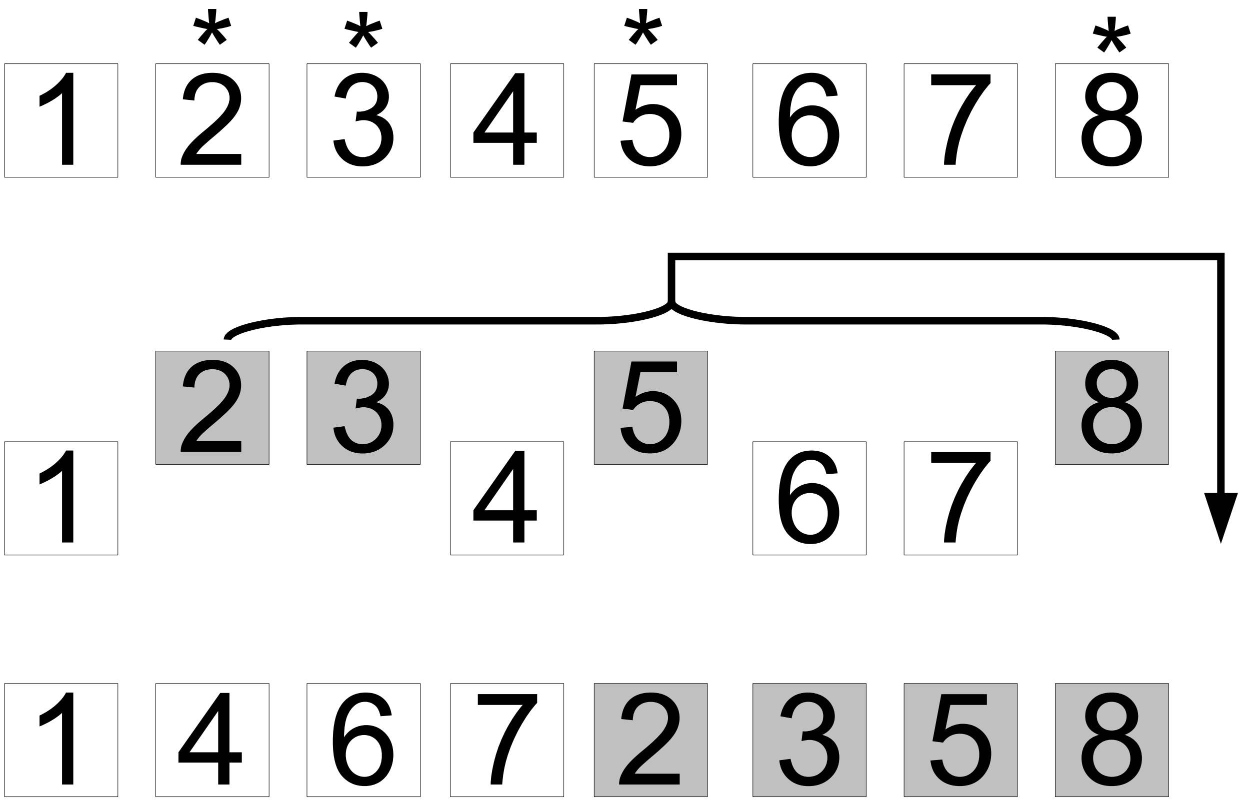
Split with probability p

Reverse the order of blocks

$\text{AND} = 1 + 2np\, f(n,p) < 3$

Theoretical result

Speed permutation



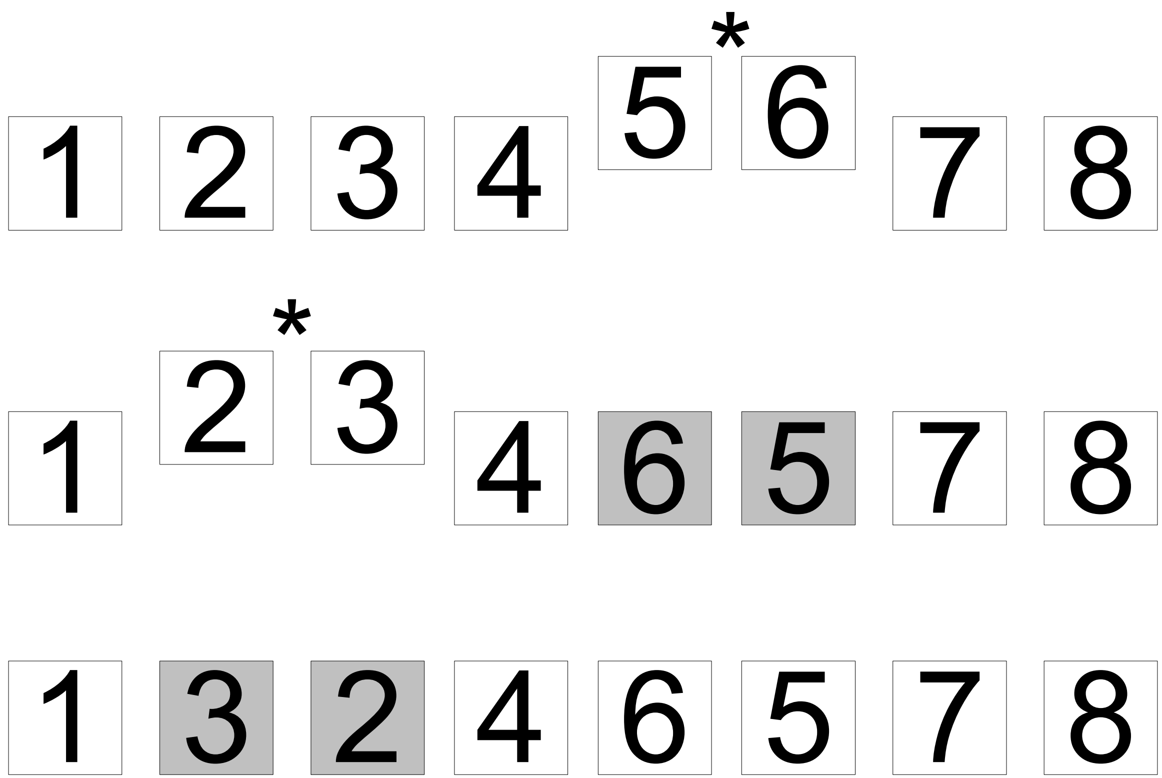
Lift with probability p

Shift lifted instances to the end

$$AND = 3(n + 1)/(n - 1) - f(n,p) \sim 3$$

Theoretical result

Shape permutation



Step 1: lift with the probability 1/n

Step 2: the same

Do $k<2n$ steps

Theoretical result

$$\text{AND} = 1 + 2k/(n - 1) - f(n,k) < \color{red}{3}$$

Demo

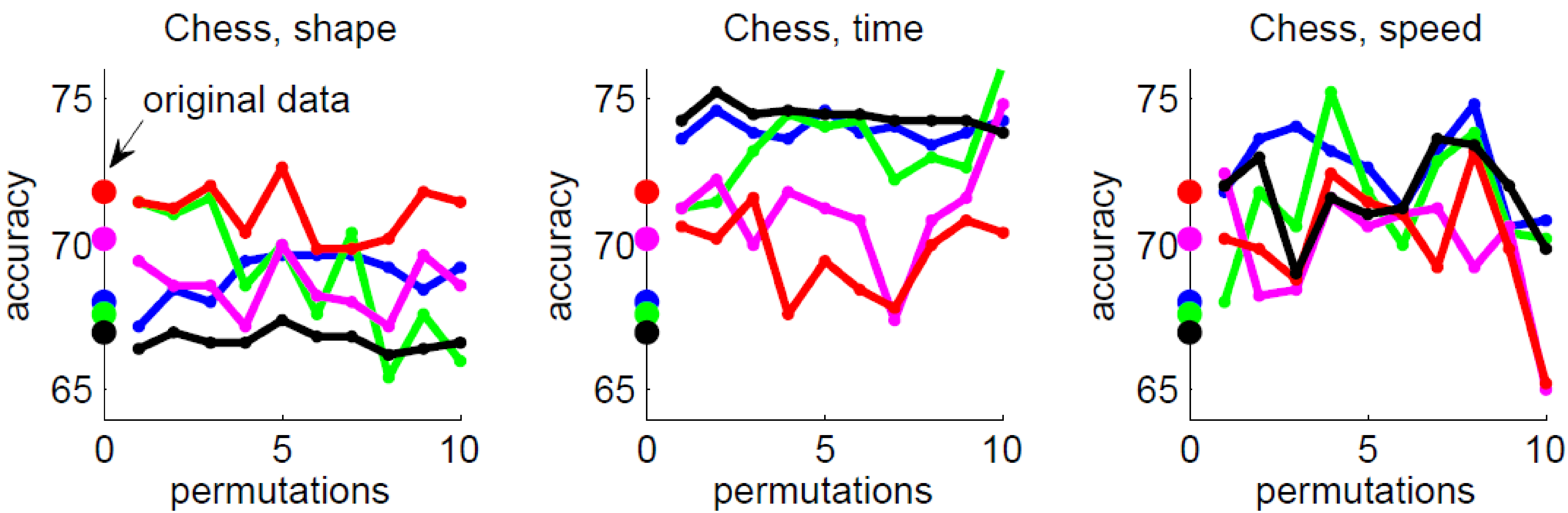
HOW TO USE
THE PERMUTATIONS

Experiments

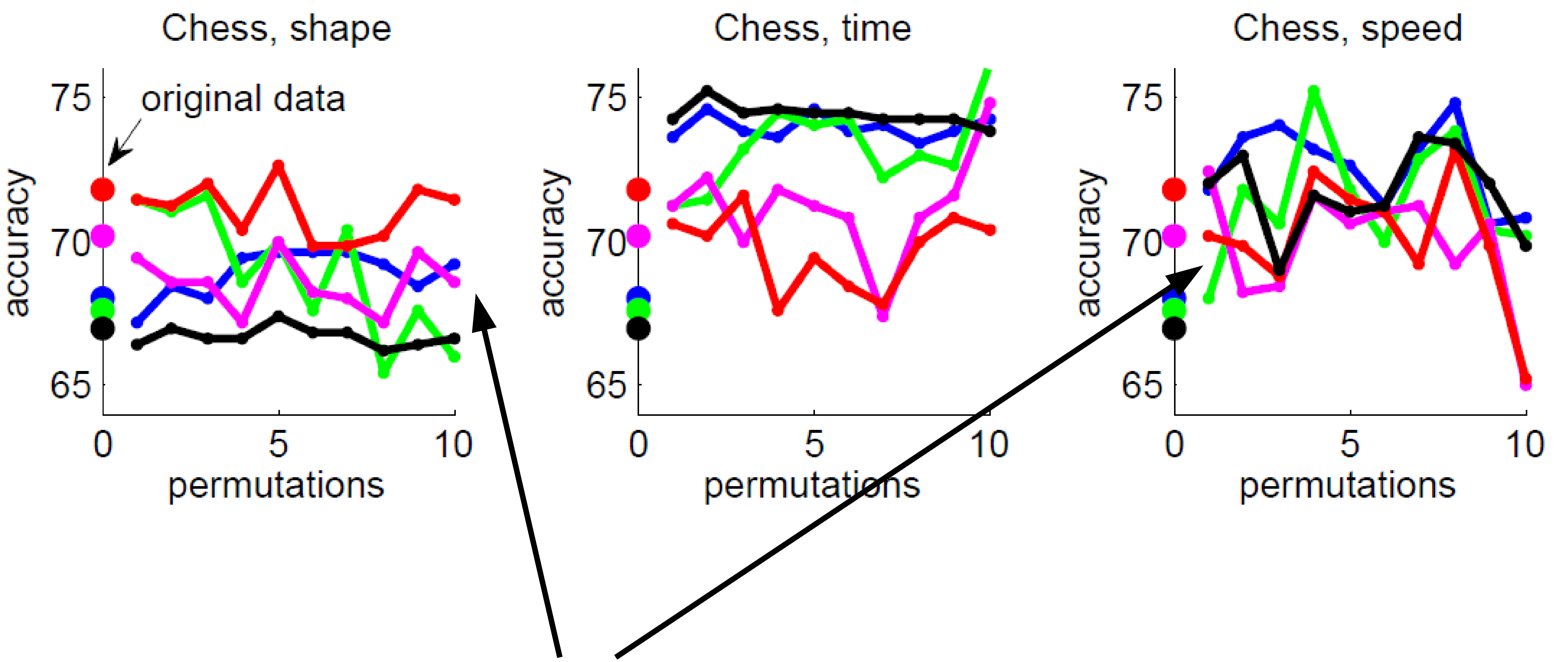
- Permuted mutiple test sets can be used
  - to assess volatility of classifier performance

- Three real datasets
  - small (chess), medium (Luxe) and large (electricity)
  - have original order in time, 2-3 years
- Five adaptive classifiers (from MOA)
  - OzaBag (1,10), DDM, EDDM, HoeffOptionTree

Volatility Chess

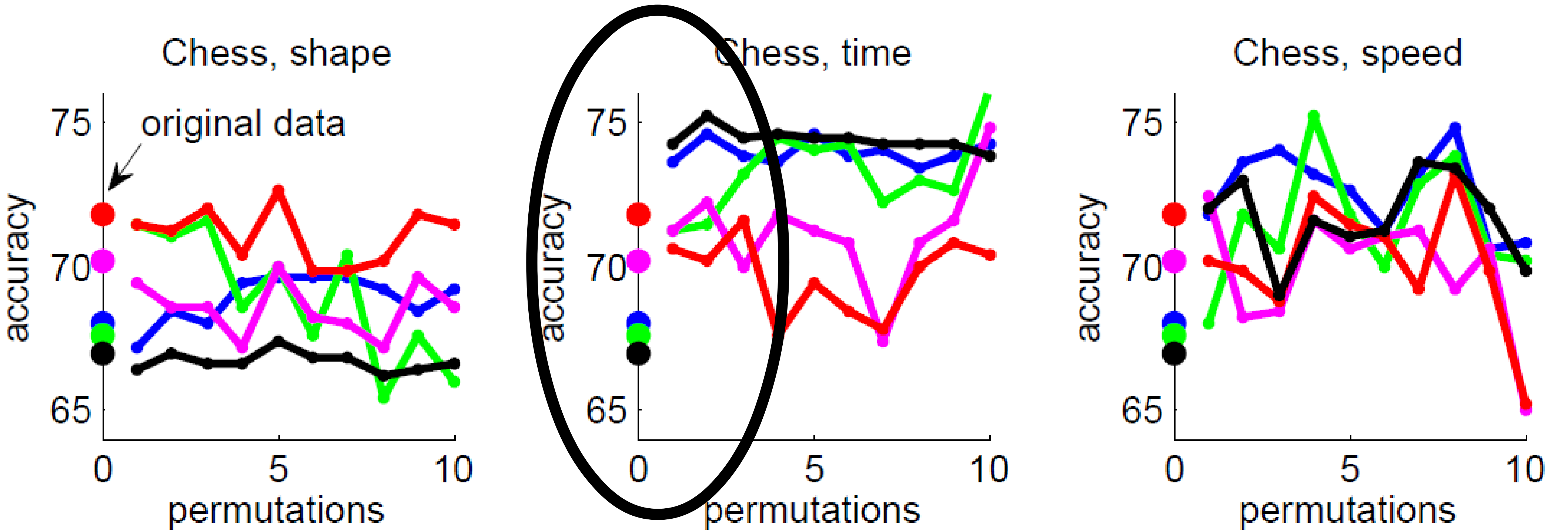


Volatility Chess



Oza1 and Oza10 show similar accuracies,
but Oza10 is much more volatile

Volatility Chess



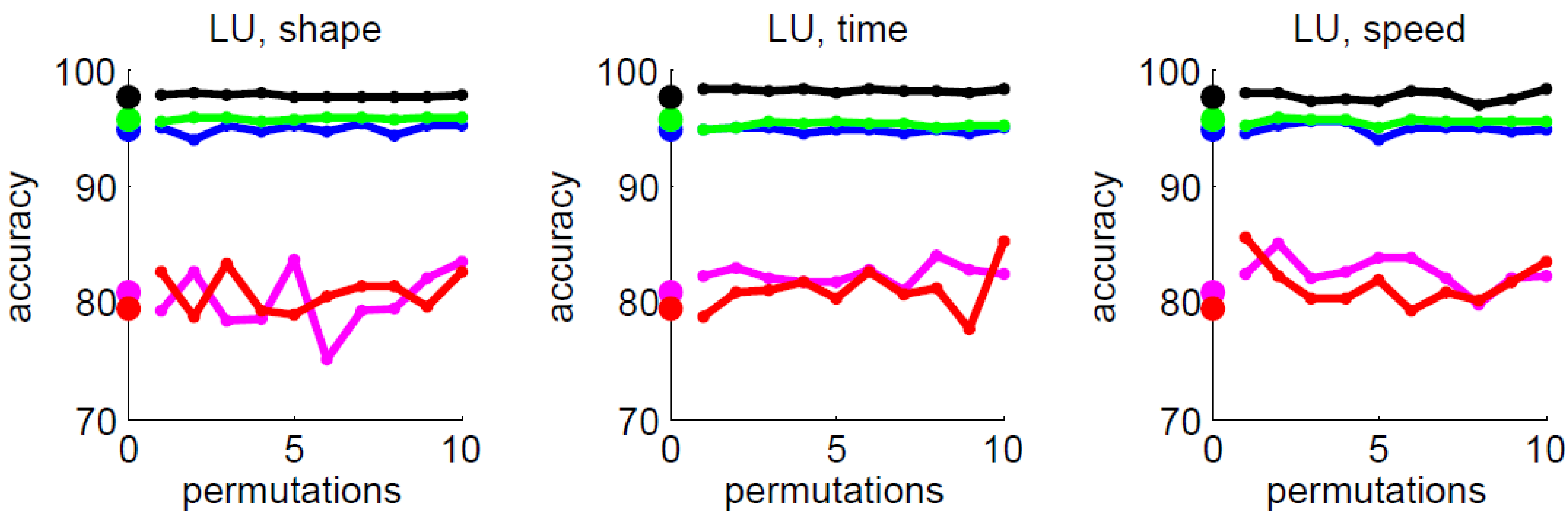
Accuracies go up when a sudden change is introduced, Hoeff and Oza do not handle gradual drift that well

Accuracy ranking Chess

- Original: EDDM>DDM>Oza1>Oza10>Hoeff
- Time: Hoeff>Oza1>Oza10>DDM>EDDM
- Speed: Oza1>Hoeff>Oza10>EDDM>DDM
- Shape: EDDM>Oza10>Oza1>DDM>Hoeff

Results over 10 permutations

Volatility Luxe

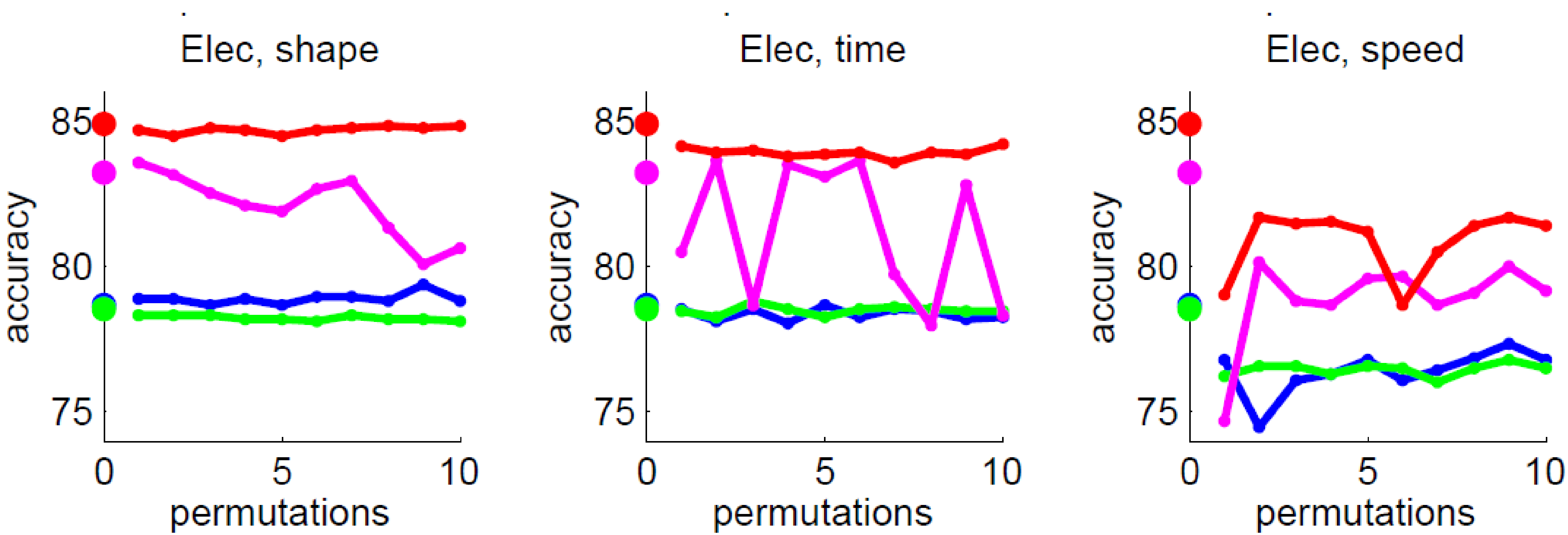


Easy dataset,
DDM and EDDM use Naive Bayes base classifier,
it is less accurate

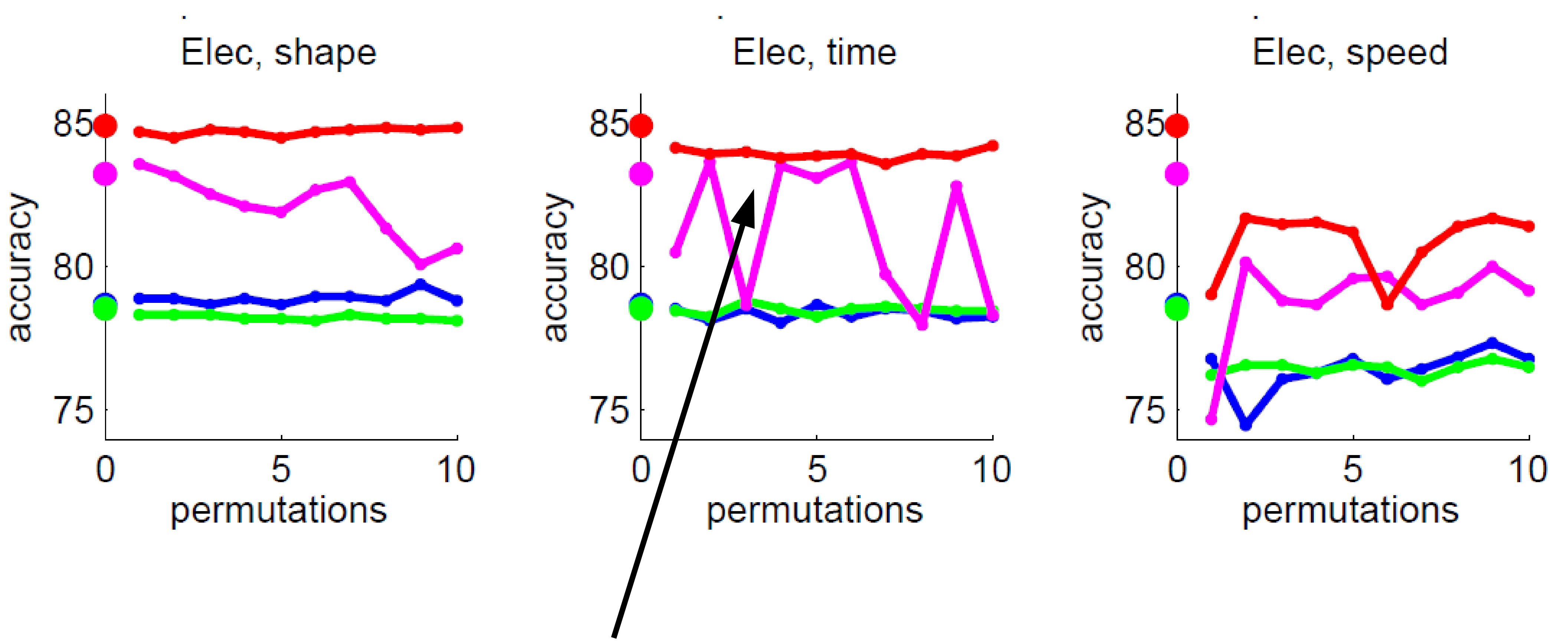
Accuracy ranking Luxe

- Original: Hoeff>Oza10>Oza1>DDM>EDDM
- Time: Hoeff>Oza10>Oza1>DDM>EDDM
- Speed: Hoeff>Oza10>Oza1>DDM>EDDM
- Shape: Hoeff>Oza10>Oza1>EDDM>DDM

Volatility Electricity

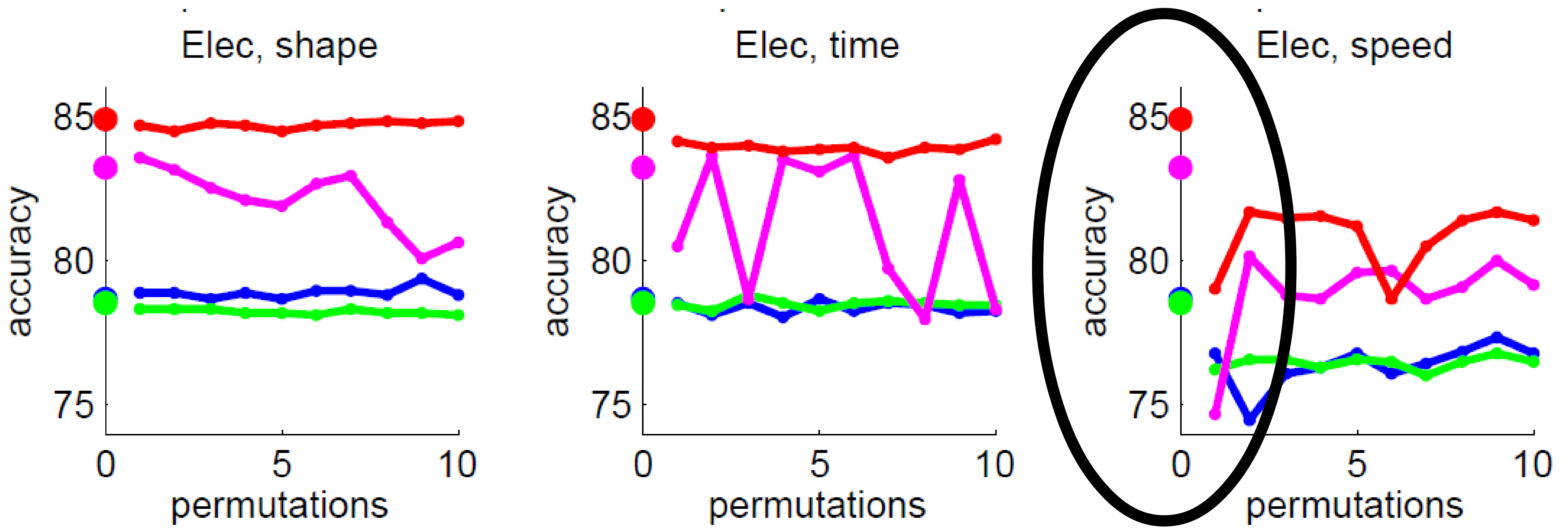


Volatility Electricity



DDM is volatile,
EDDM is not

Volatility Electricity



If we speed up the data, accuracies go down

Accuracy ranking Electricity

- Original: EDDM>DDM>Oza1>Oza10
- Time: EDDM>DDM>Oza10>Oza1
- Speed: EDDM>DDM>Oza1=Oza10
- Shape: EDDM>DDM>Oza1>Oza10

FINAL

Conclusion

- The test-then-train evaluation risks overfitting
- Multiple tests
  - can reduce the risk of overfitting,
  - as they allow to analyze volatility,
  - and assess robustness of the learning models
- A random permutation would destroy changes
- The controlled permutations allow to generate multiple test sets that preserve local distributions

The end, thank you!

http://zliobaite.googlepages.com/permutations