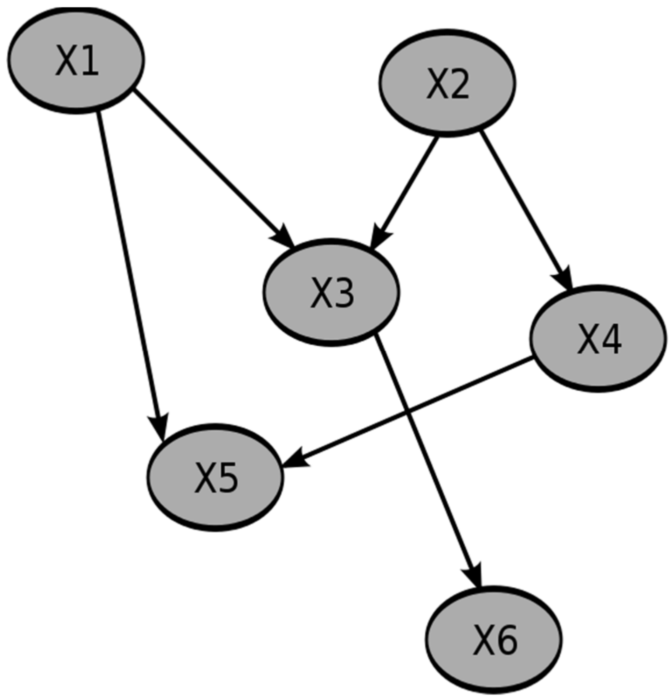
Unsupervised Data Mining: From Batch to Stream Mining Algorithms

Stefan Kramer

Data Mining Group

Johannes Gutenberg-Universität Mainz

Bayesian Networks for **Batch** Discrete Density Estimation

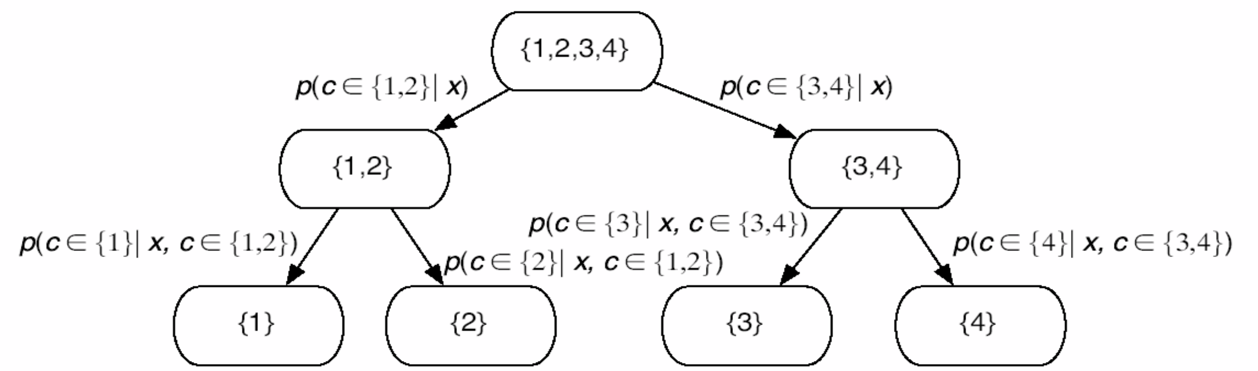


- *Joint probability distribution*
  $P(X_1, X_2, X_3, X_4, X_5, X_6) =$
  $P(X_1)P(X_2)P(X_3|X_1, X_2)$
  $P(X_4|X_2)P(X_5|X_1, X_4)P(X_6|X_3)$
- Queryable
- Structure learning
- *Can this be done online? Arbitrary queries on arbitrarily large data*
- Idea: conditional probabilities or probabilisitic classifiers as basis for an (online probabilistic) inductive database

Prerequisites

Multi-Class Classification: Ensembles of Nested Dichotomies (Frank & Kramer, ICML 2004)

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
0	1	0	0	1	0	1	0	0
1	0	0	1	1	0	1	0	0
0	0	1	1	1	1	0	0	0
0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	1	0
...	...	...	...	...	...	...	...	...



$$P(c=C \mid X_1,\dots X_m)=\prod_{i=1}^{k-1}\left(I(c\in C_{i1})P(c\in C_{i1}\mid X_1,\dots,X_m,c\in C_i)+I(c\in C_{i2})P(c\in C_{i2}\mid X_1,\dots,X_m,c\in C_i)\right)$$

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
0	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1
0	0	1	1	1	1	0	0	1
0	0	1	0	1	1	0	1	0
0	0	0	1	1	1	0	1	0
...	...	...	...	...	...	...	...	...

Classifier chains for multi-label classification (Read *et al.*, ECML/PKDD 2009)

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	$\hat{Y}_1$
0	1	0	0	1	1
1	0	0	1	1	0
0	0	1	1	1	1
0	0	1	0	1	1
0	0	0	1	1	1
...	...	...	...	...	...

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	Y_1	$\hat{Y}_2$
0	1	0	0	1	1	1
1	0	0	1	1	0	1
0	0	1	1	1	1	0
0	0	1	0	1	1	0
0	0	0	1	1	1	0
...	...	...	...	...	...	...

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	$\hat{Y}_3$
0	1	0	0	1	1	1	0
1	0	0	1	1	0	1	0
0	0	1	1	1	1	0	0
0	0	1	0	1	1	0	1
0	0	0	1	1	1	0	1
...	...	...	...	...	...	...	...

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	$\hat{Y}_4$
0	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1
0	0	1	1	1	1	0	0	1
0	0	1	0	1	1	0	1	0
0	0	0	1	1	1	0	1	0
...	...	...	...	...	...	...	...	...

Multi-Label Classification: Ensembles of Classifier Chains

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
0	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1
0	0	1	1	1	1	0	0	1
0	0	1	0	1	1	0	1	0
0	0	0	1	1	1	0	1	0
...	...	...	...	...	...	...	...	...

Probabilistic classifier chains:

$$P(Y_1,\ldots,Y_L \mid X_1,\ldots,X_m) = P(Y_1 \mid X_1,\ldots,X_m)\prod_{j=2}^{L} P(Y_j \mid Y_1,\ldots,Y_{j-1}, X_1,\ldots,X_m)$$

Hoeffding Tree Algorithm (Domingos & Hulten, KDD 2000)

Procedure HoeffdingTree(*Stream*, δ)

Let HT = Tree with single leaf (root)

Initialize sufficient statistics at root

For each example (X, Y) in *Stream*

  Sort (X, Y) to leaf using HT

  Update sufficient statistics at leaf

  Compute G for each attribute

  If G(best) – G(2nd best) > $\varepsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n}}$

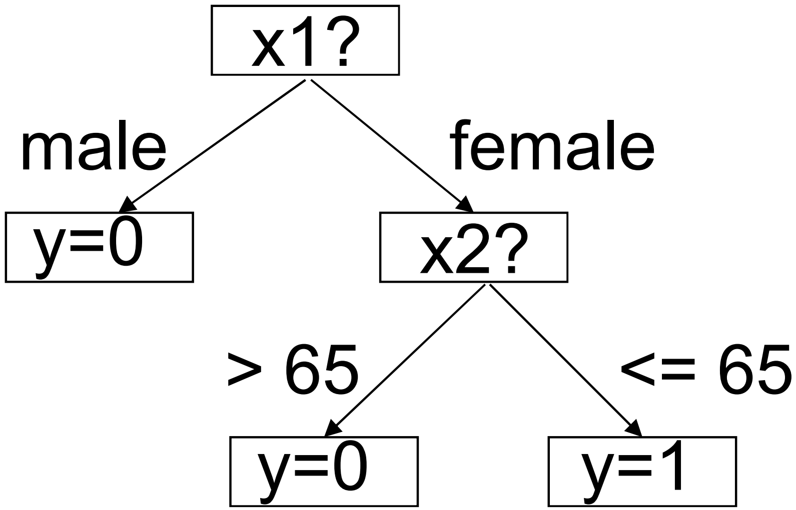
  then

    Split leaf on best attribute

    For each branch

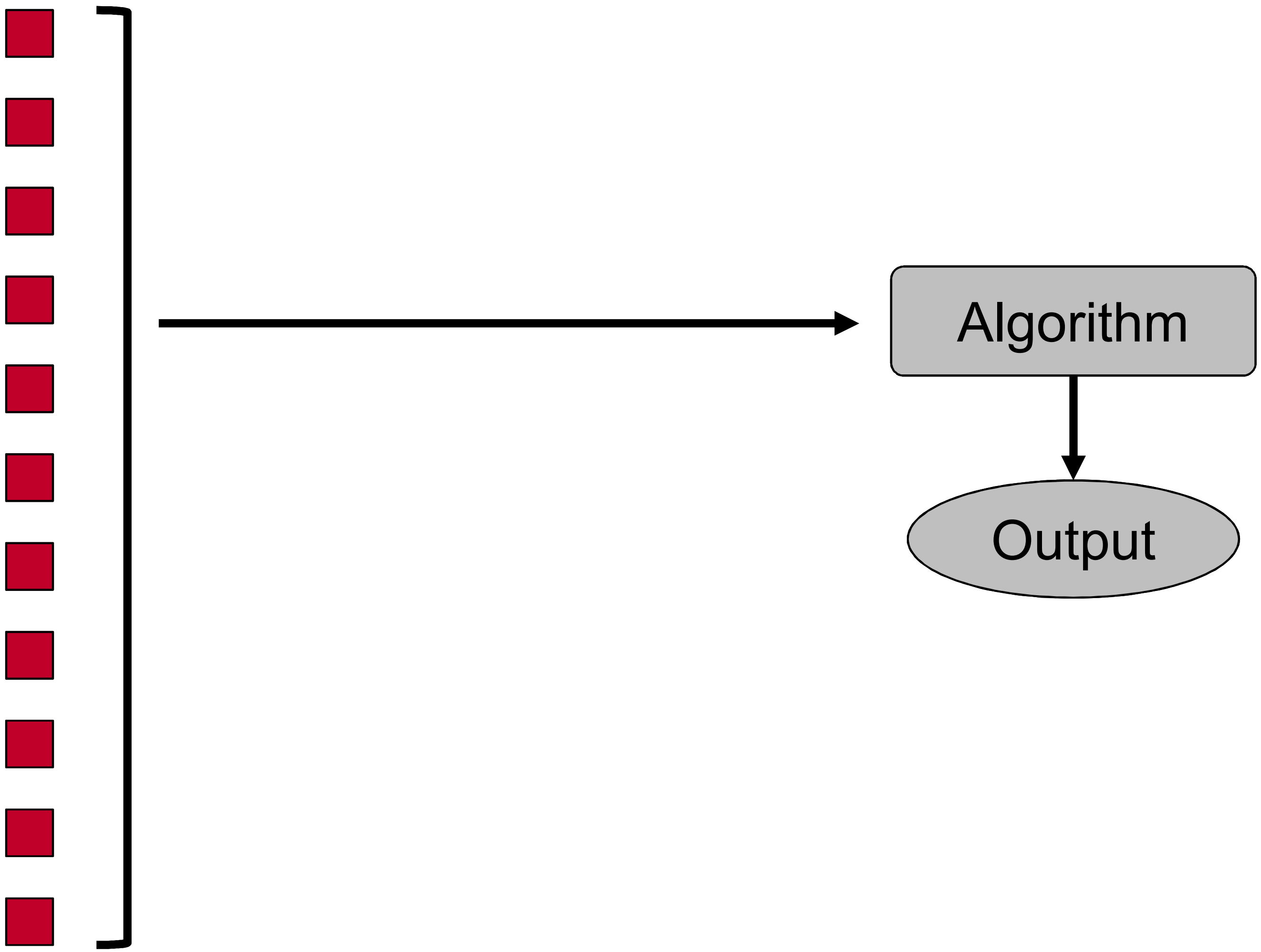
      Start new leaf, init sufficient statistics

Return HT

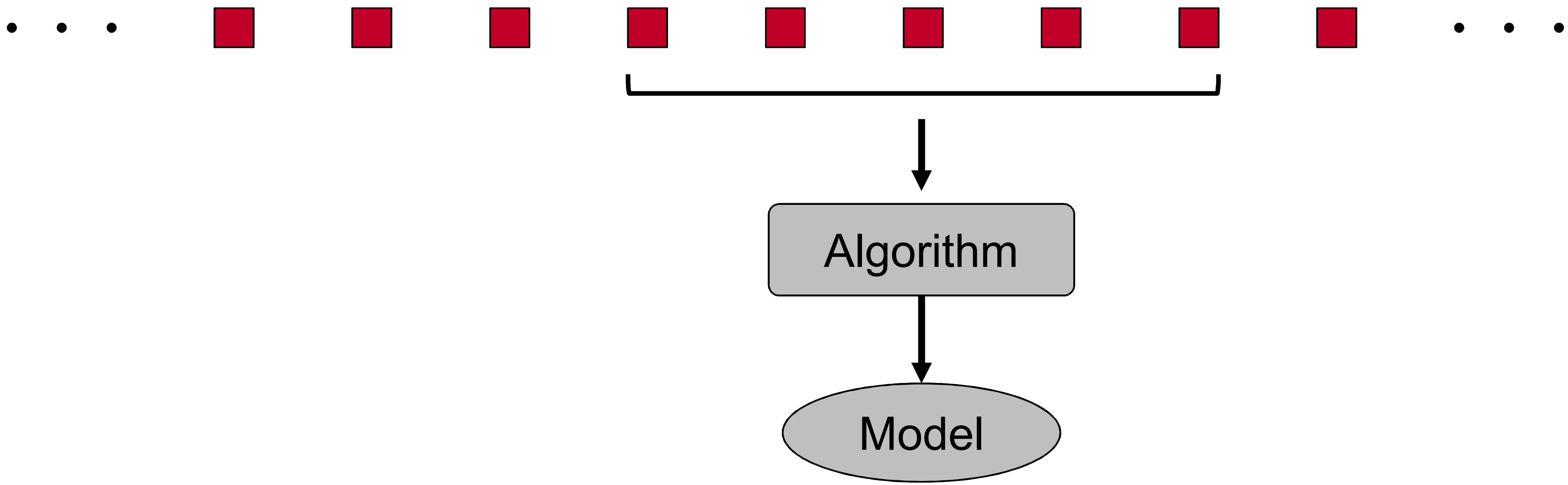


Estimation of Discrete Densities Online (EDDO)

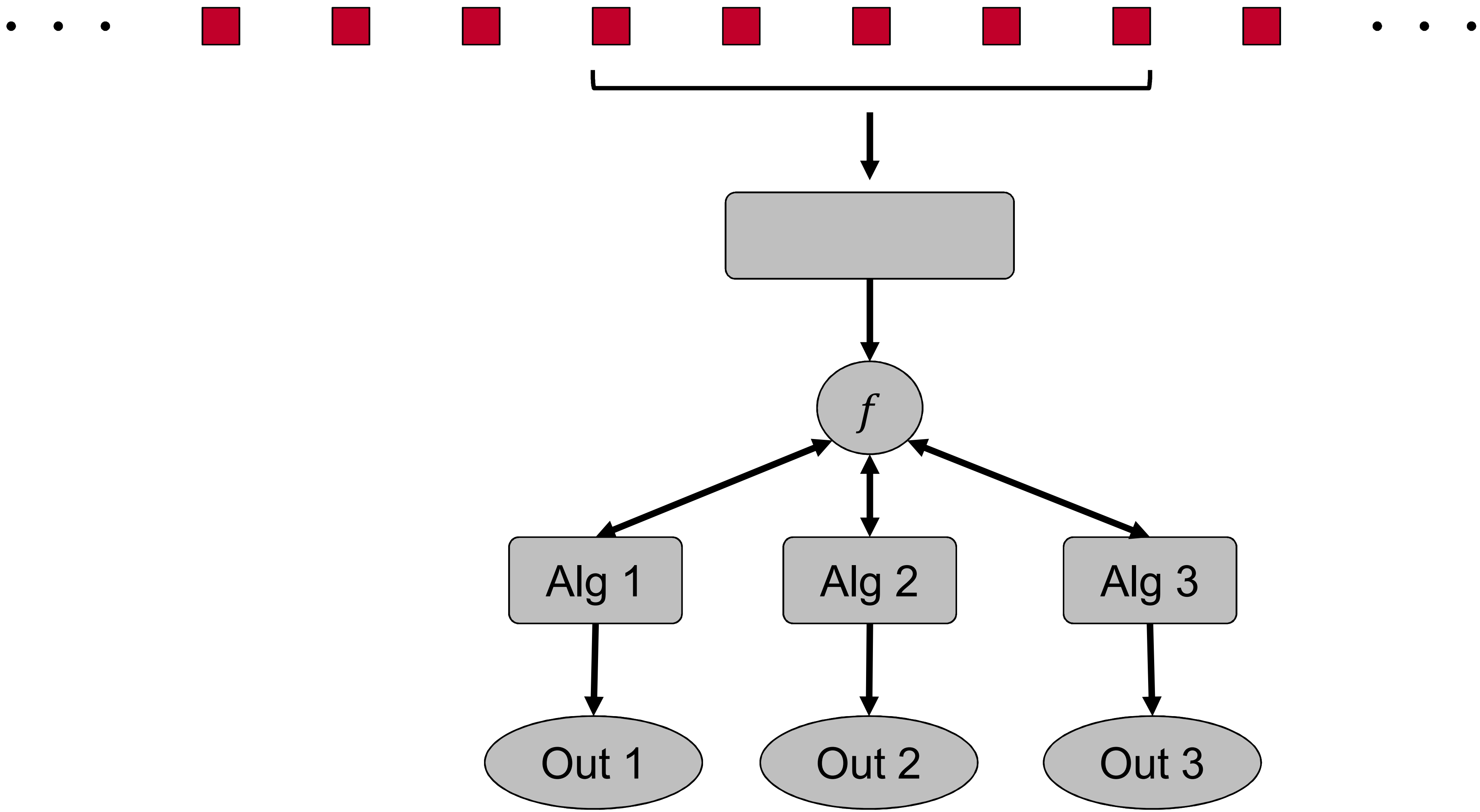
Not Batch Learning, ...



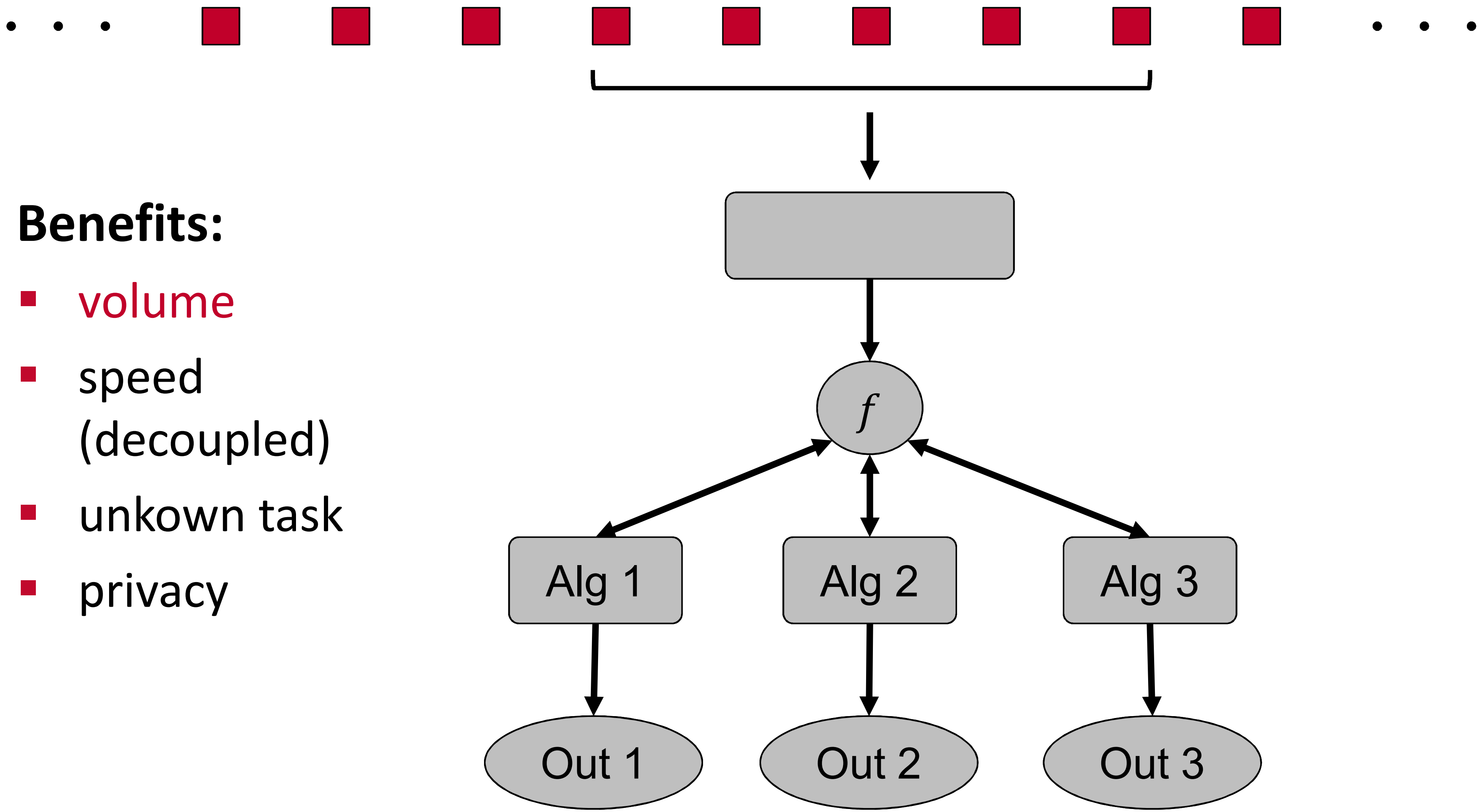
… But Stream Mining



Condensed Representation



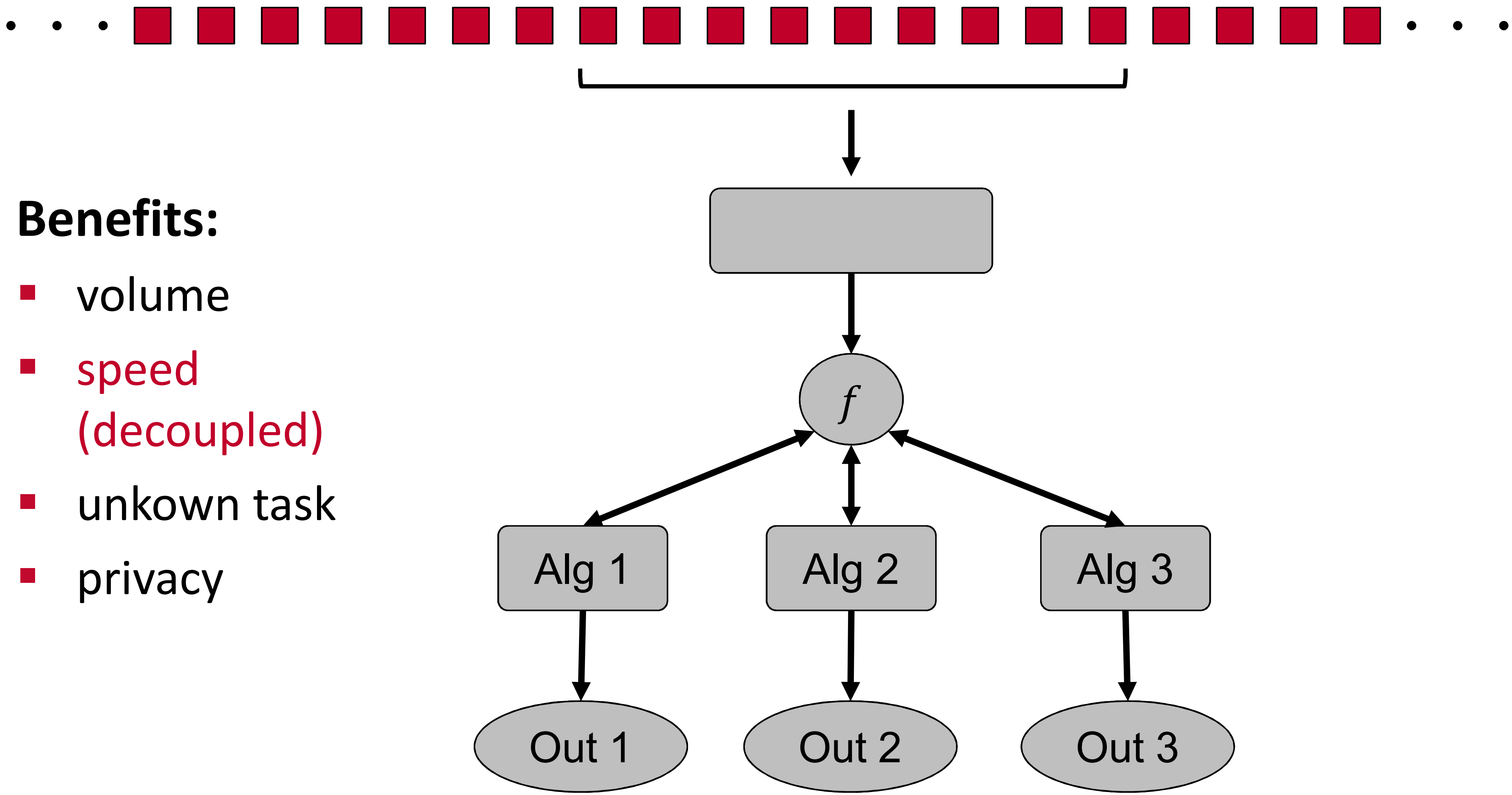
Benefits



Benefits:

- volume
- speed (decoupled)
- unkown task
- privacy

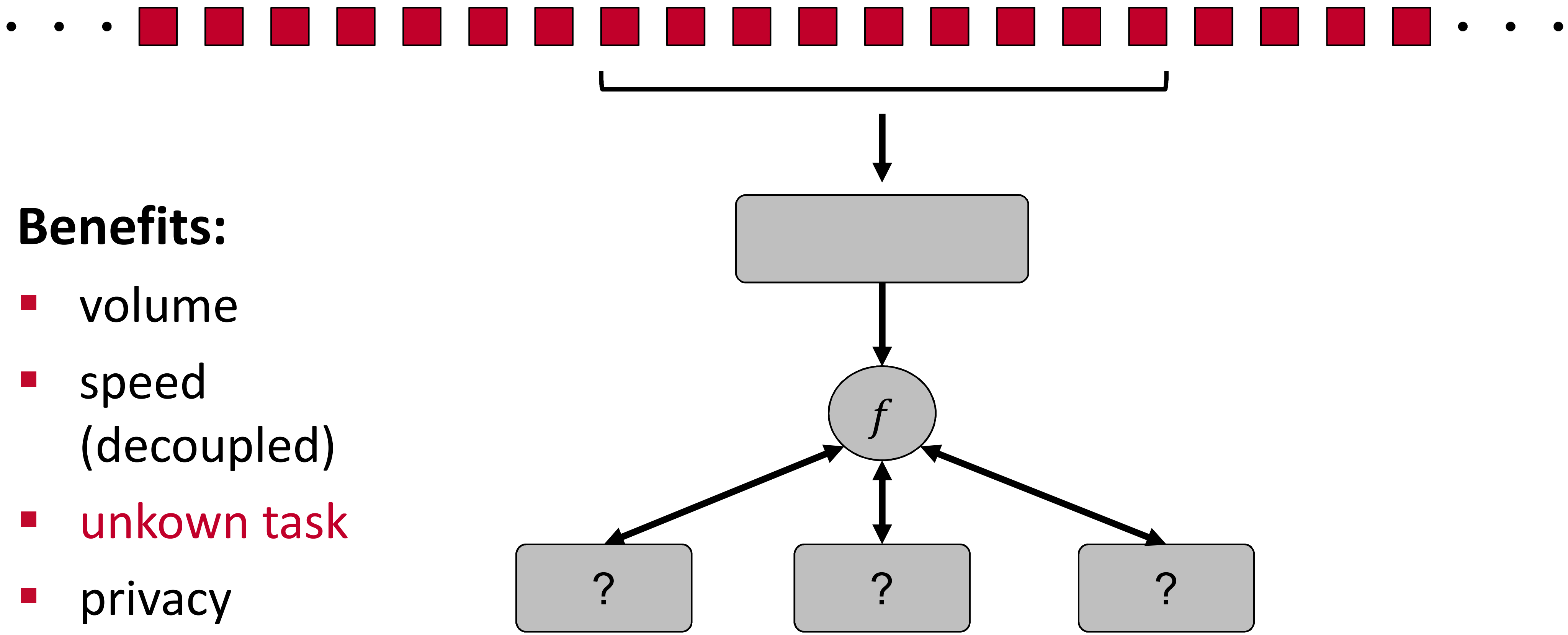
Benefits



Benefits:

- volume
- speed (decoupled)
- unkown task
- privacy

Benefits



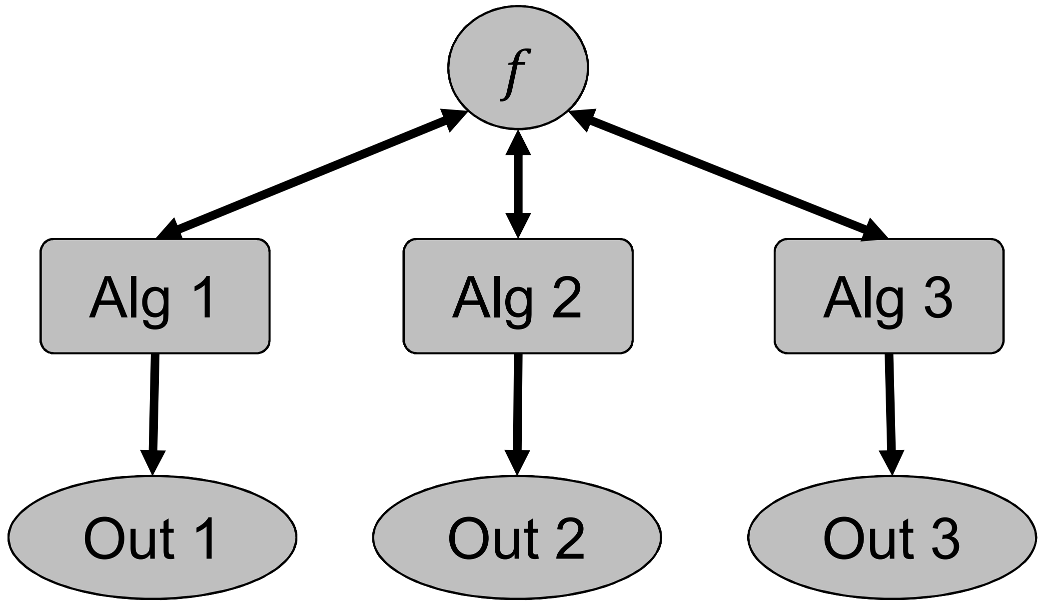
Benefits:

- volume
- speed (decoupled)
- unkown task
- privacy

Benefits

Benefits:

- volume
- speed (decoupled)
- unkown task
- privacy



Problem Statement

Given: nominal variables $X_1, X_2, \ldots, X_n$, an unknown discrete joint density $f(X_1, X_2, \ldots, X_n)$, and an infinite stream of data that is distributed according to f

Find: A density estimate $\hat{f}$ for f in an online fashion, i.e., in an instance- or batch-incremental way.

Estimators need to be consistent and enable inference tasks (hard evidence, e.g., $f(X_1, X_2, X_4 | X_3 = b, X_5 = c)$ or soft evidence or frequency queries, e.g., $f(X_1, X_2, X_3, X_4, X_5) > \theta$).

Modeling Discrete Densities using Classifier Chains

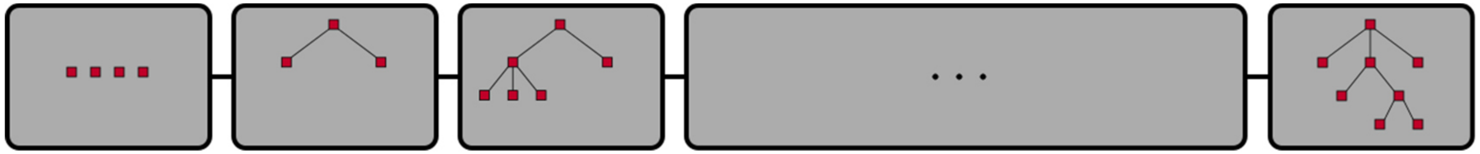
Applying the product rule to $f(X_1, X_2, \ldots, X_n)$ yields

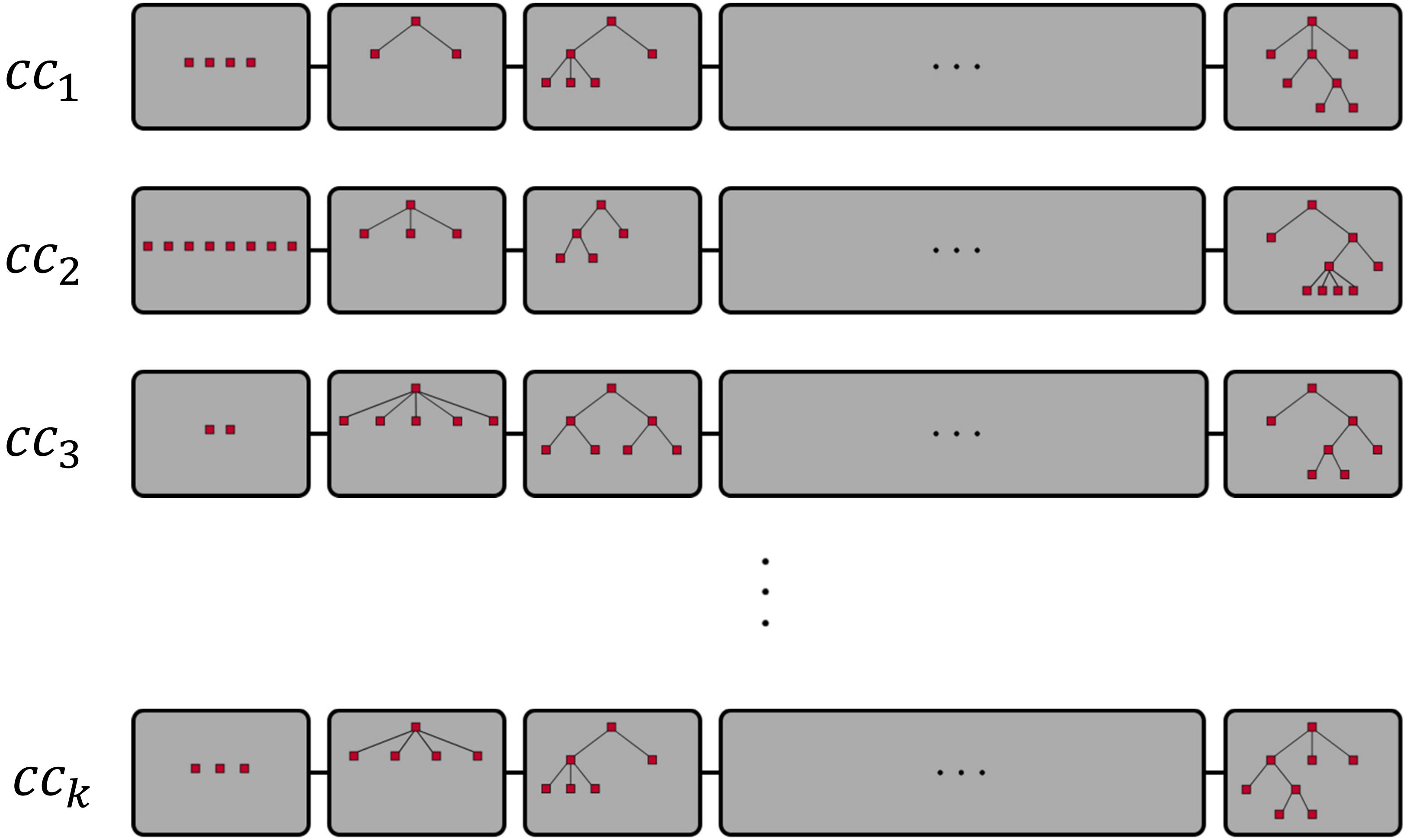
$$f_1(X_1) \cdot f_2(X_2 \mid X_1) \cdot \ \ldots \ \cdot f_n(X_n \mid X_1, X_2, \ldots, X_{n-1})$$

Classifier	
Majority class	for $f_1(X_1)$
Hoeffding trees	for $f_i(X_n \mid X_1, X_2, \ldots, X_{i-1})$

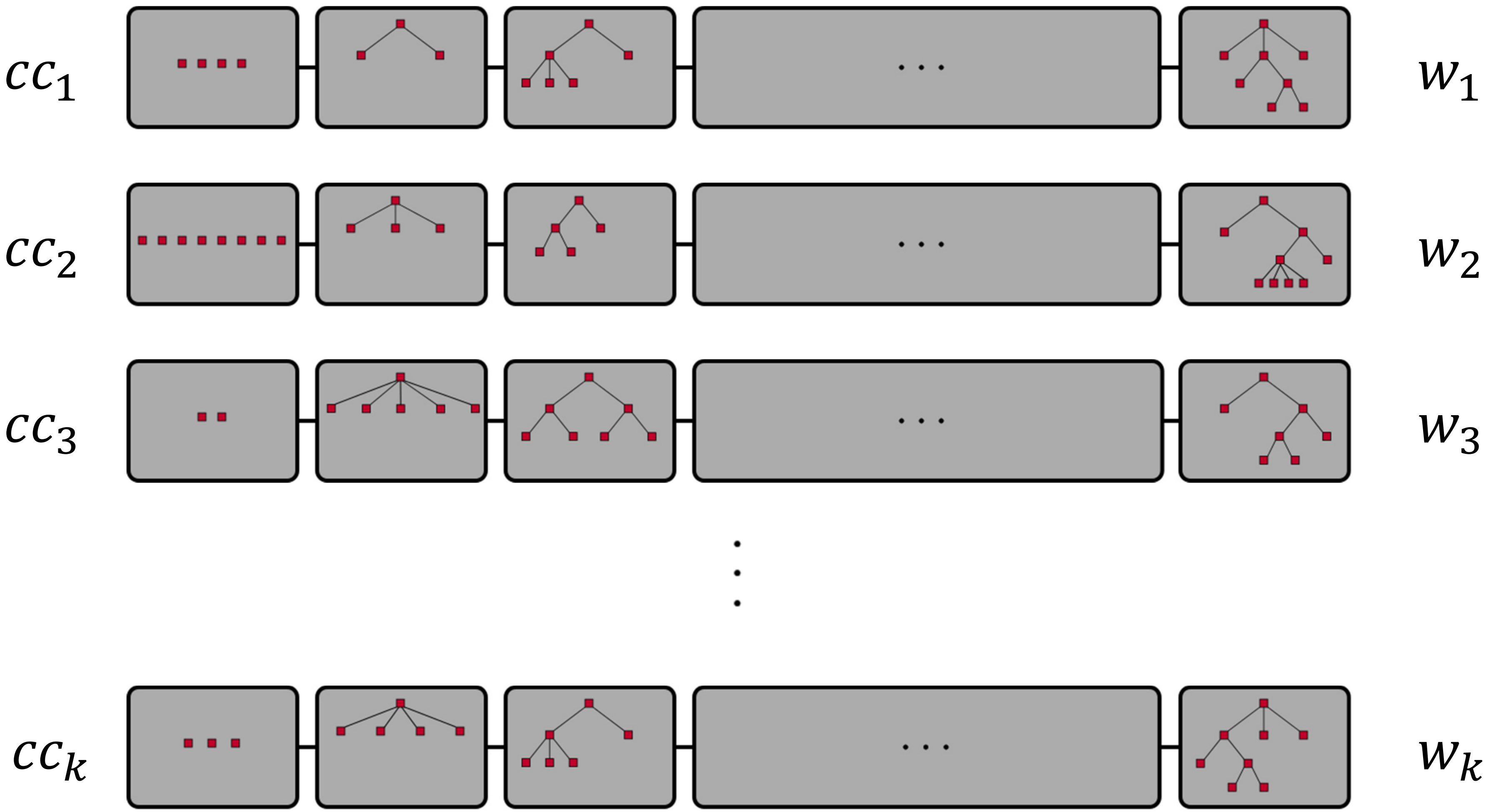
- Both allow us to estimate the density in an online fashion.
- Take *permutations* for *ensembles of chains* (performance may vary!)

cc_1

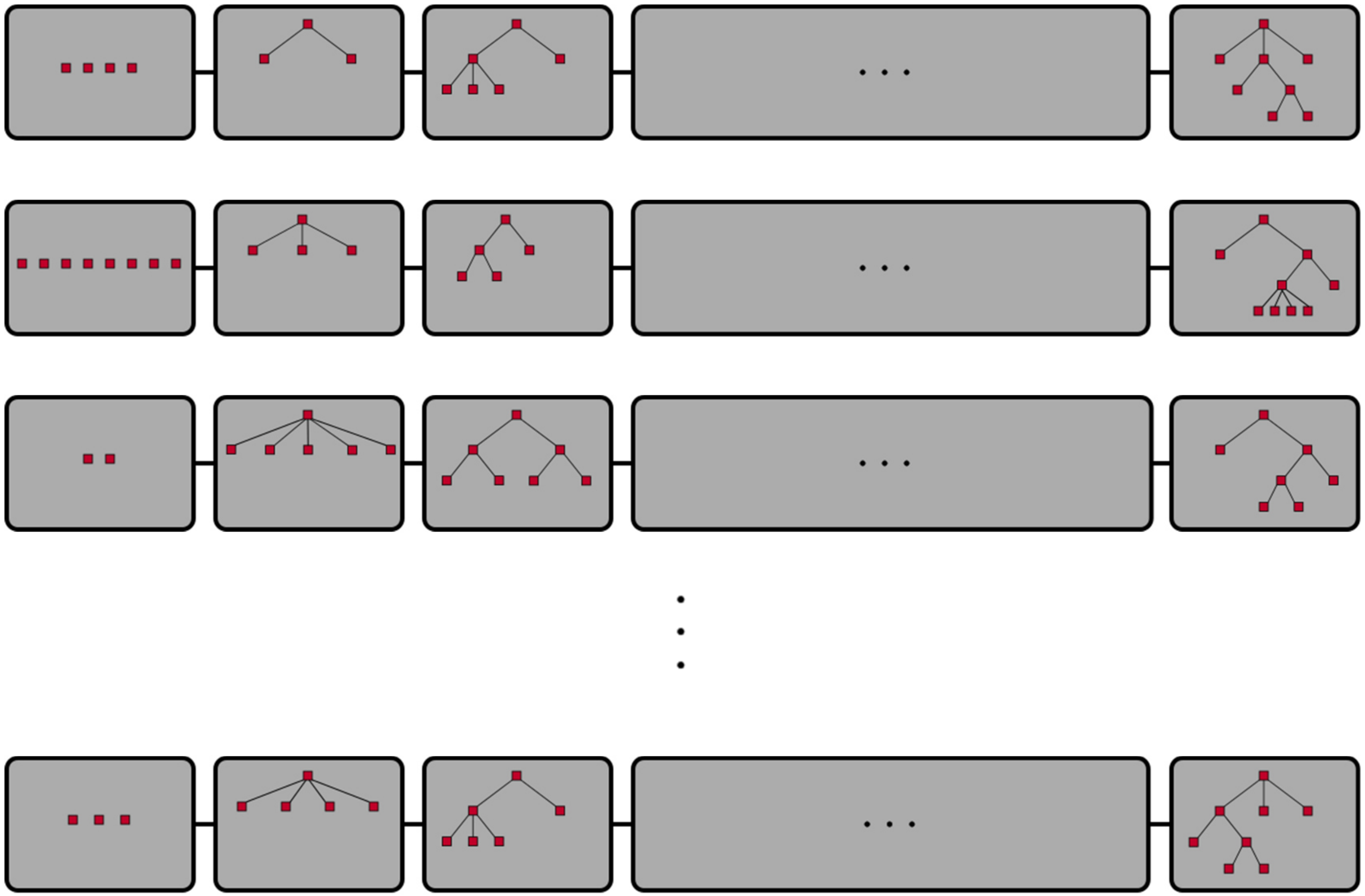



cc_1
cc_2
cc_3
cc_k

Ensembles of Weighted Classifier Chains (EWCC)



Ensembles of Weighted Classifier Chains (EWCC)



cc_1 — $L_1 := -log_2\, p_1$

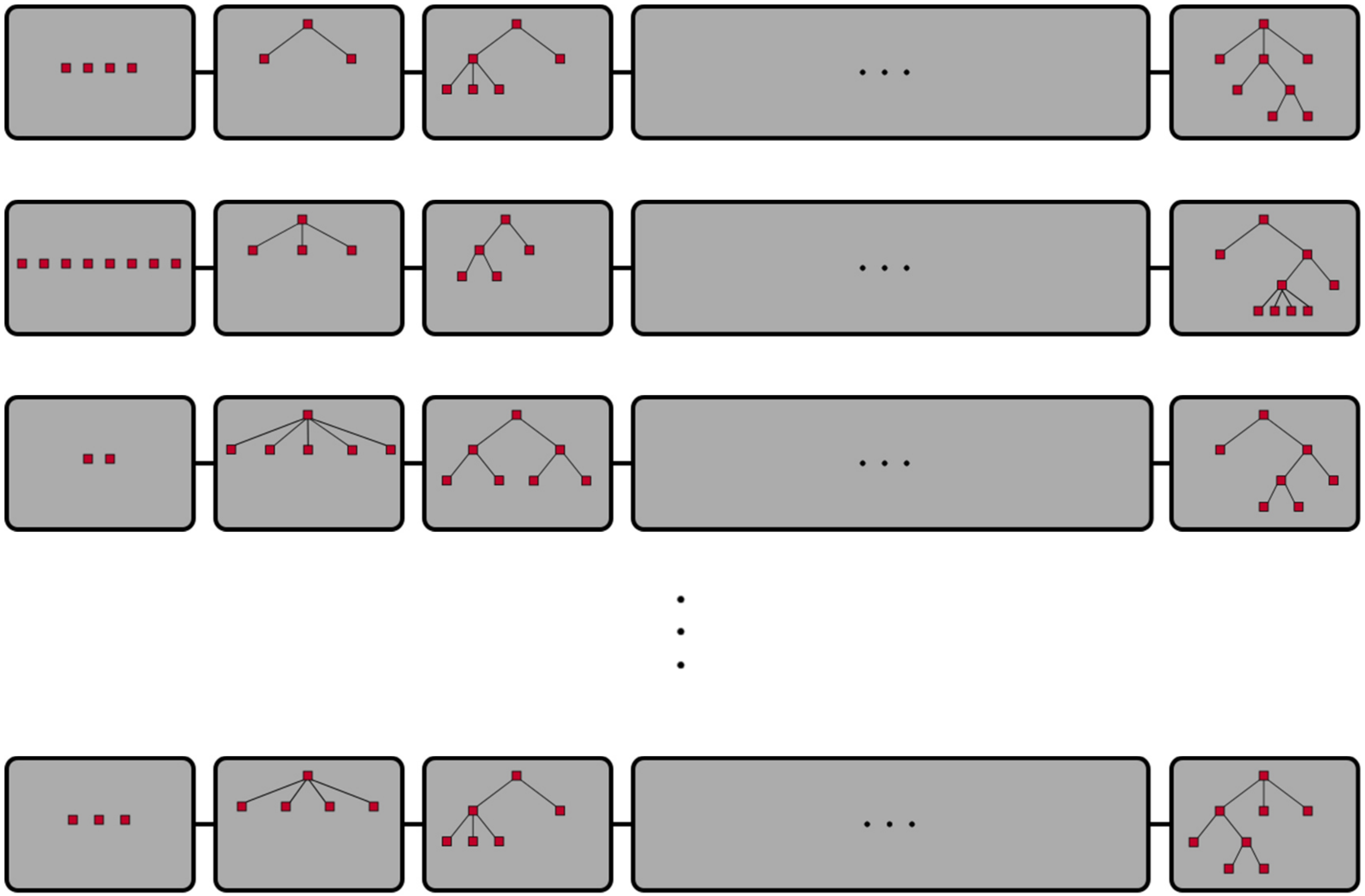
cc_2 — $L_2 := -log_2\, p_2$

cc_3 — $L_3 := -log_2\, p_3$

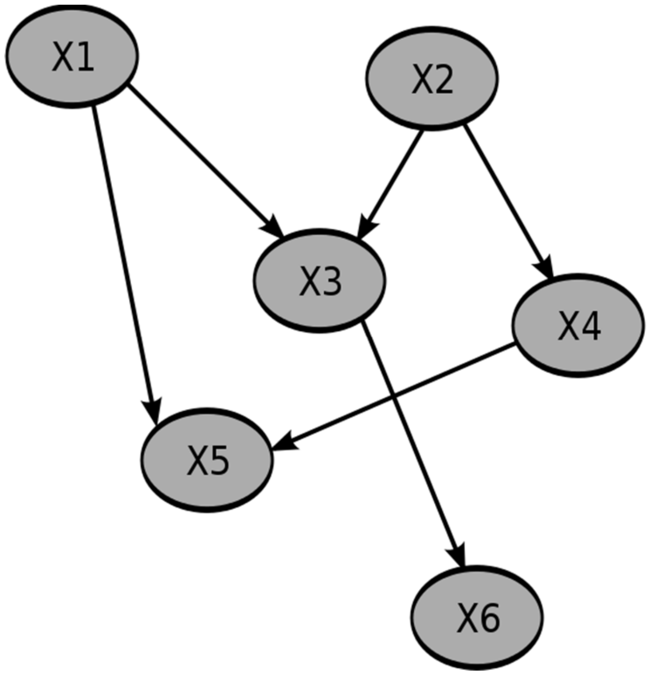
⋮

cc_k — $L_k := -log_2\, p_k$

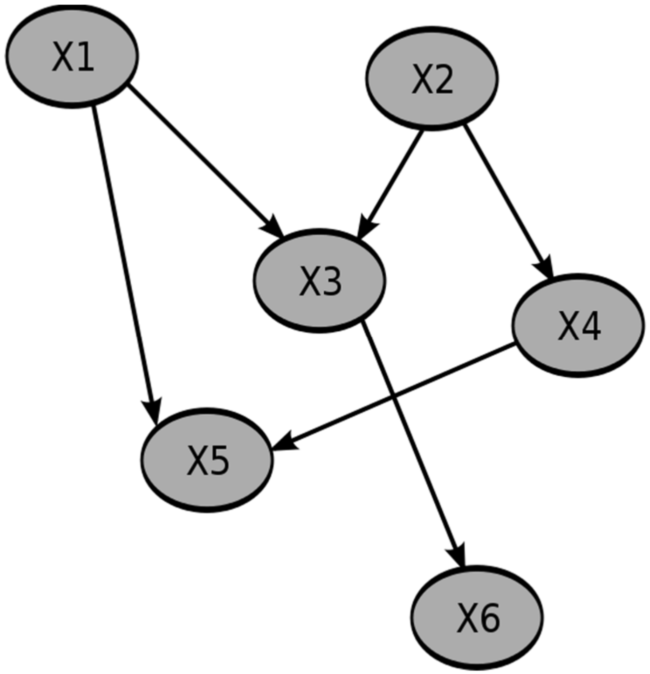
Ensembles of Weighted Classifier Chains (EWCC)

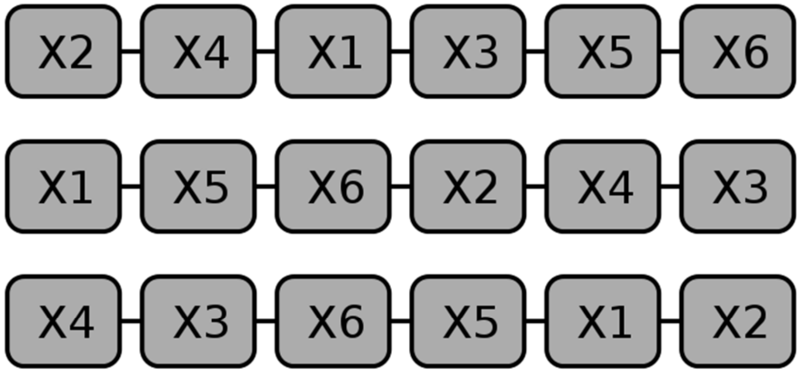


Ensembles of Classifier Chains (ECC)

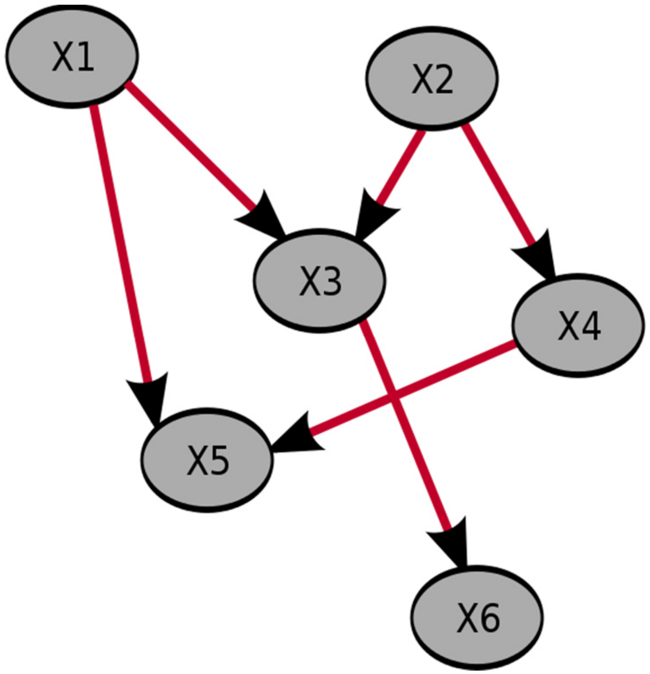


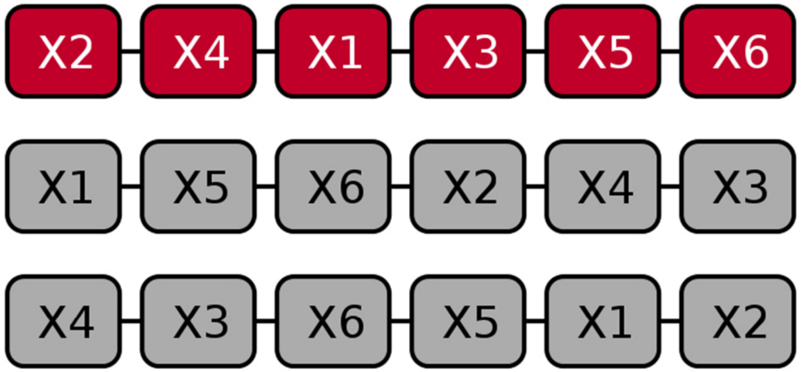
Ensembles of Classifier Chains (ECC)





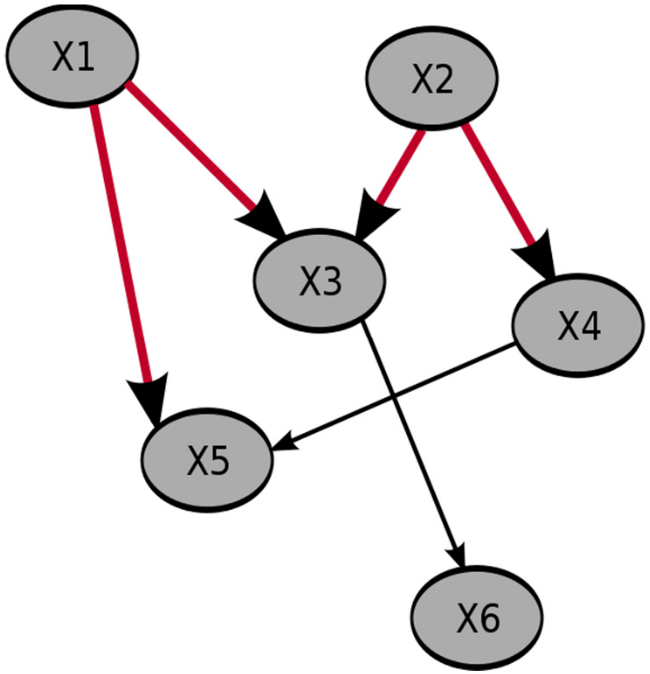
Ensembles of Classifier Chains (ECC)

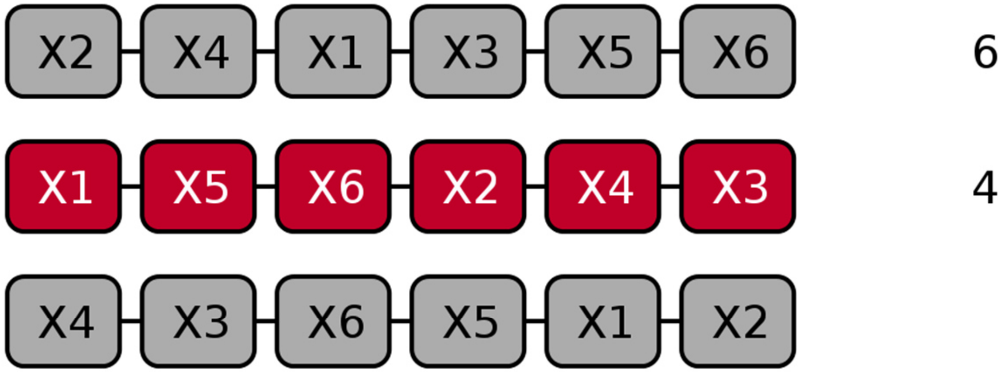




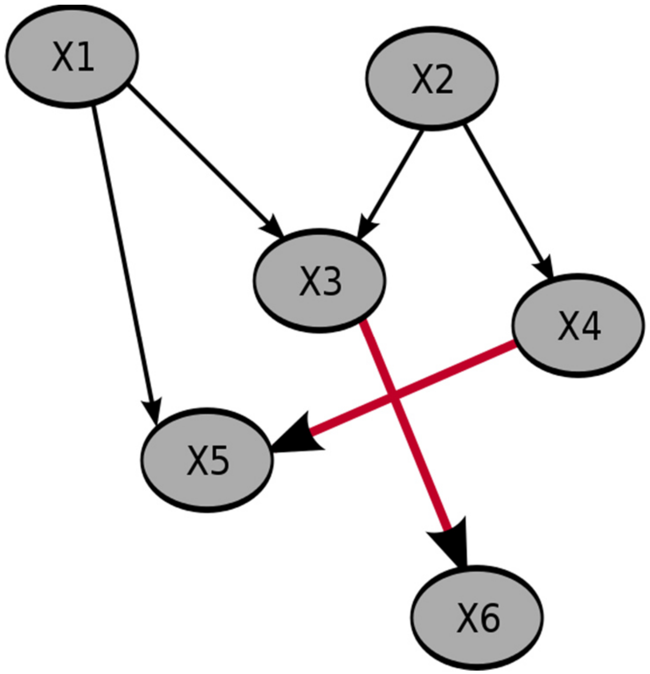
6

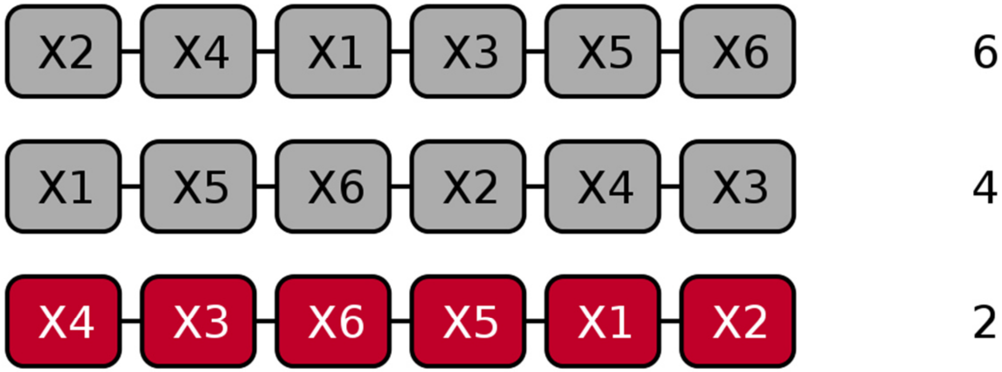
Ensembles of Classifier Chains (ECC)



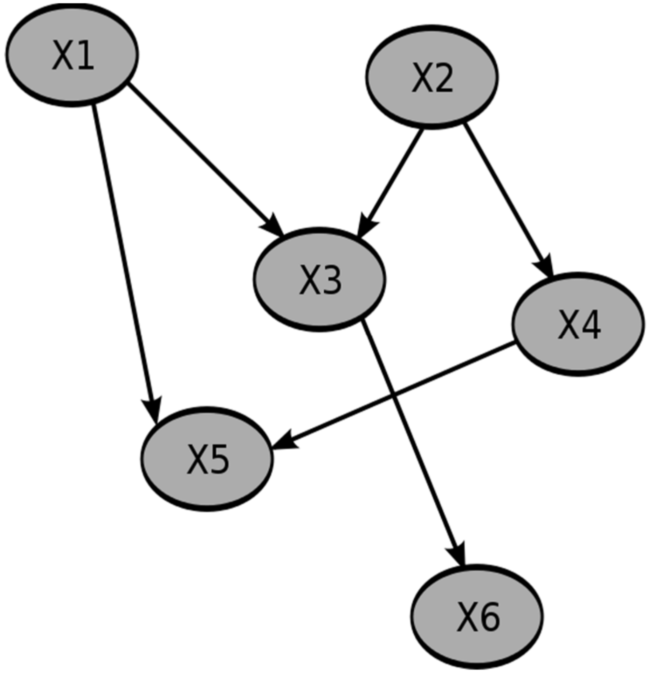


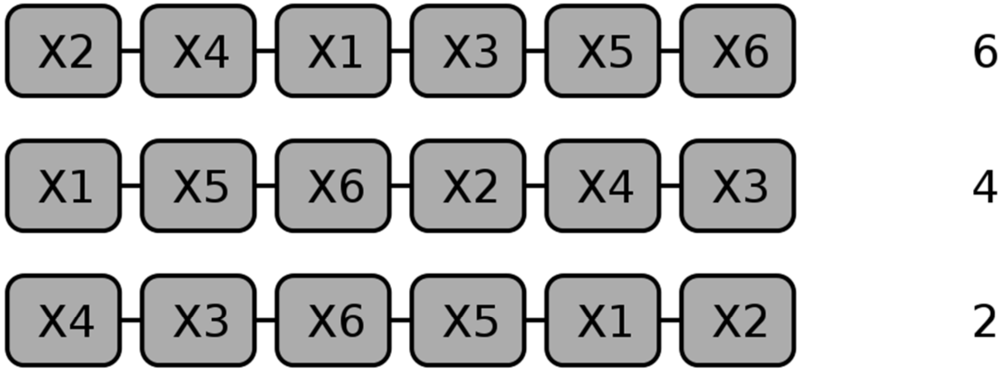
Ensembles of Classifier Chains (ECC)





Ensembles of Classifier Chains (ECC)



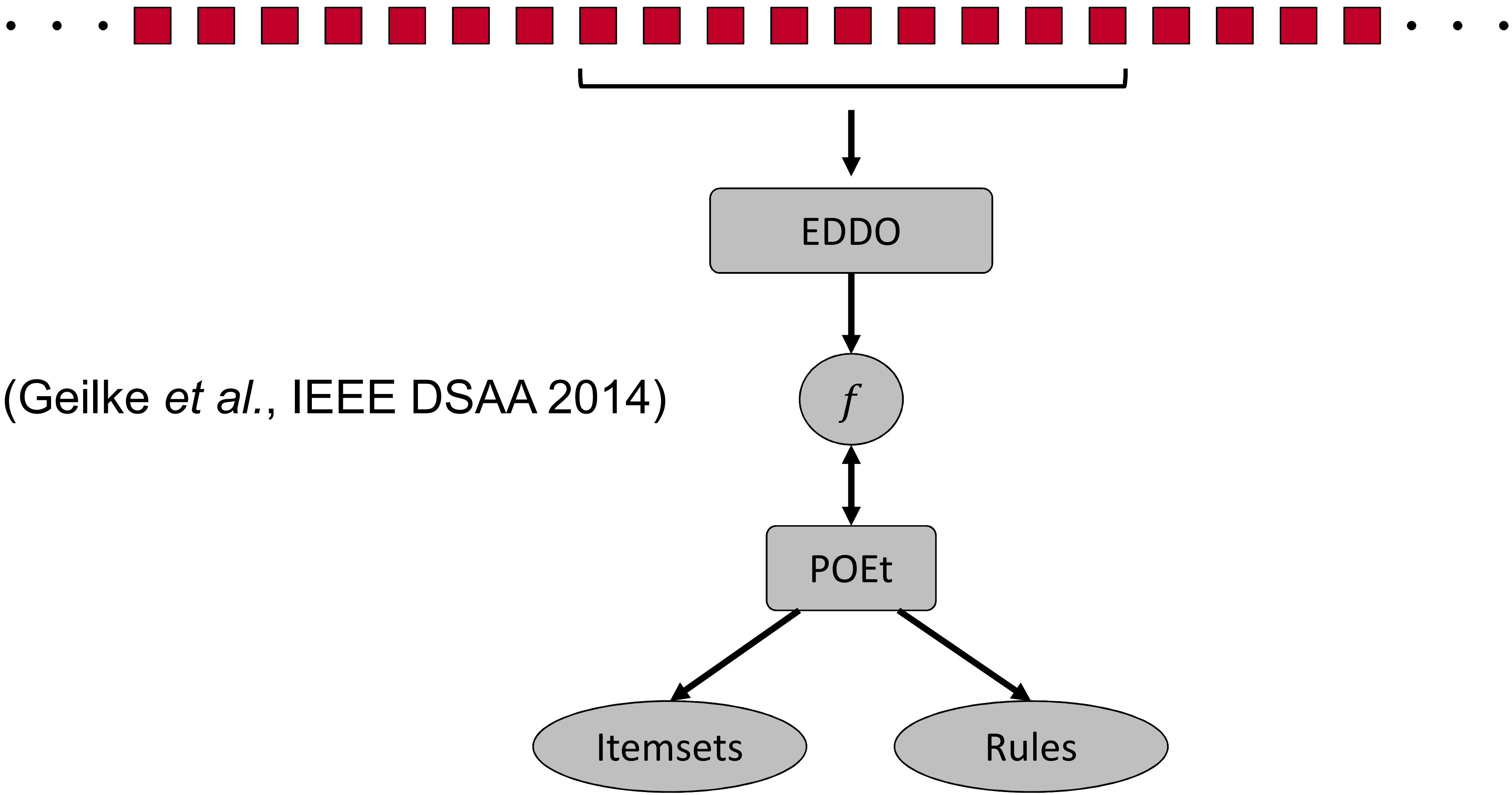


Hence, to increase robustness, we

- sample chains at random from the set of possible variable orderings,
- and average over the density estimates obtained.

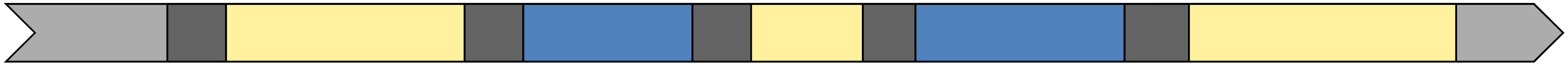
On batch data, this outperforms 12 BN structure learners (Geilke *et al.*, IEEE ICDM 2013).

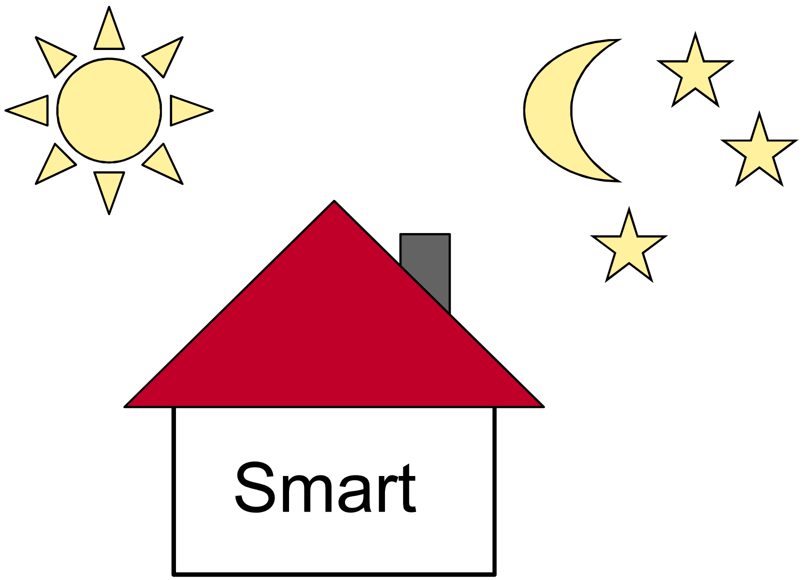
Pattern Mining



(Geilke *et al.*, IEEE DSAA 2014)

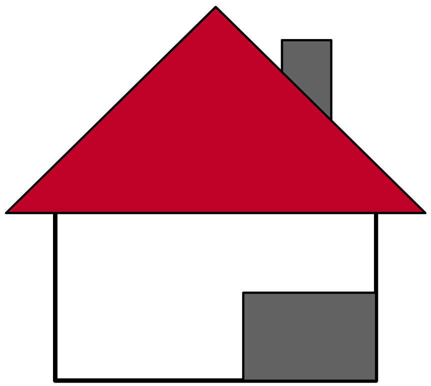

EDO
f
Smart





Recurrences

- day and night
- working days and weekends
- seasons



Recurrences

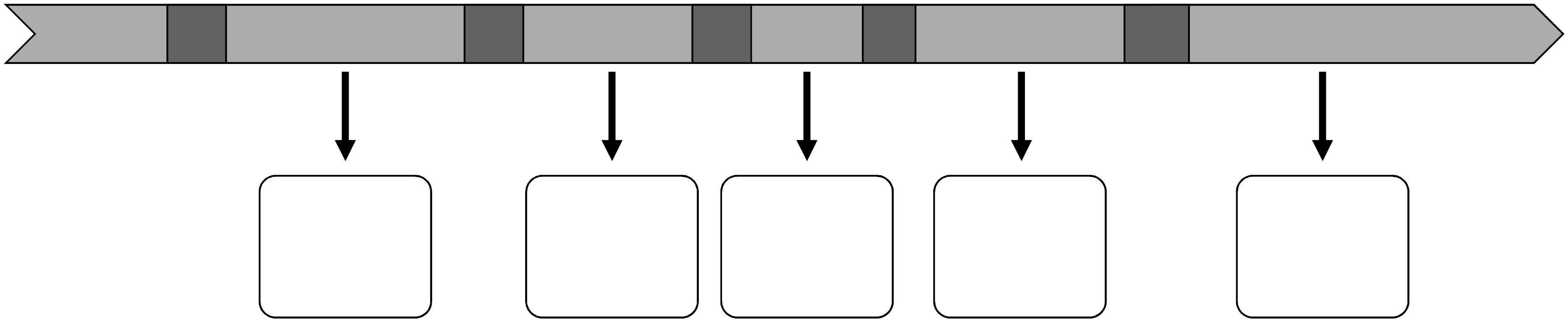
- pattern could be more complex
- may only affect a part of the house

Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities

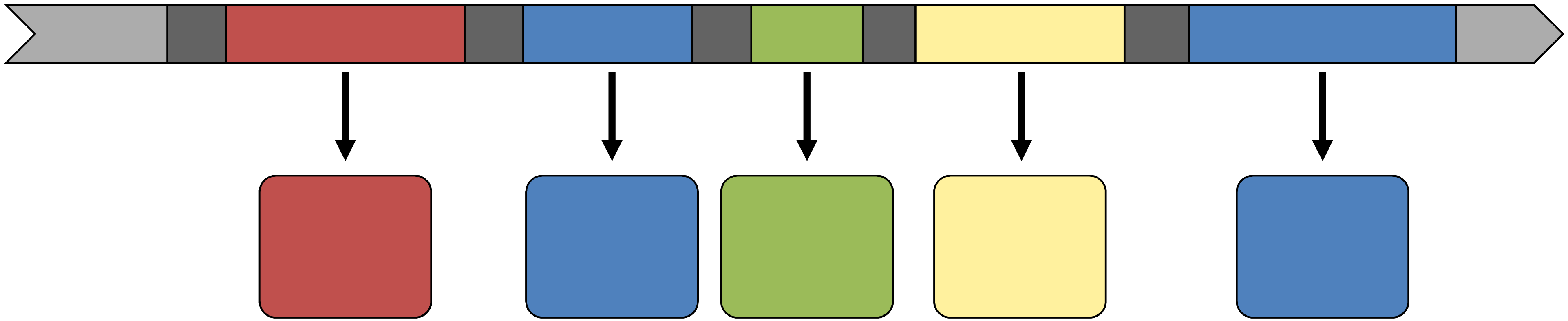
Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities



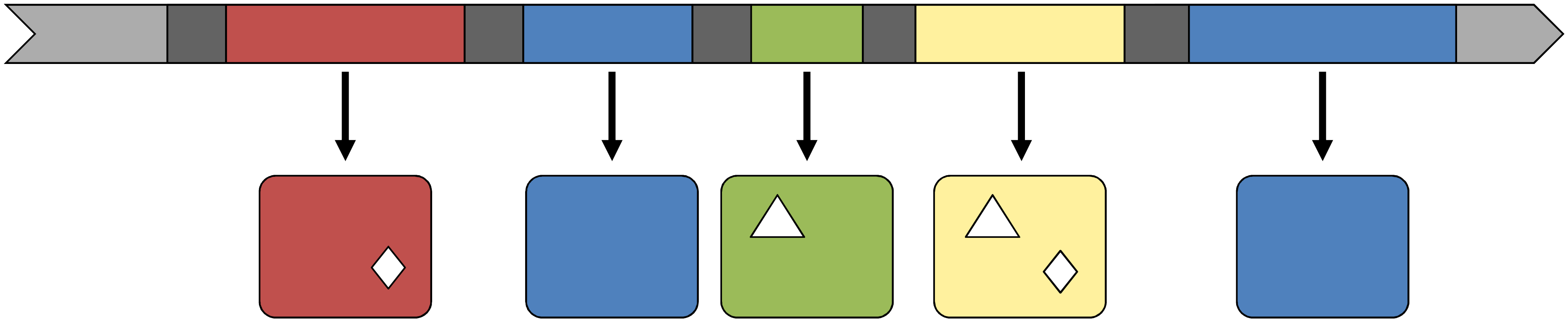
Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities



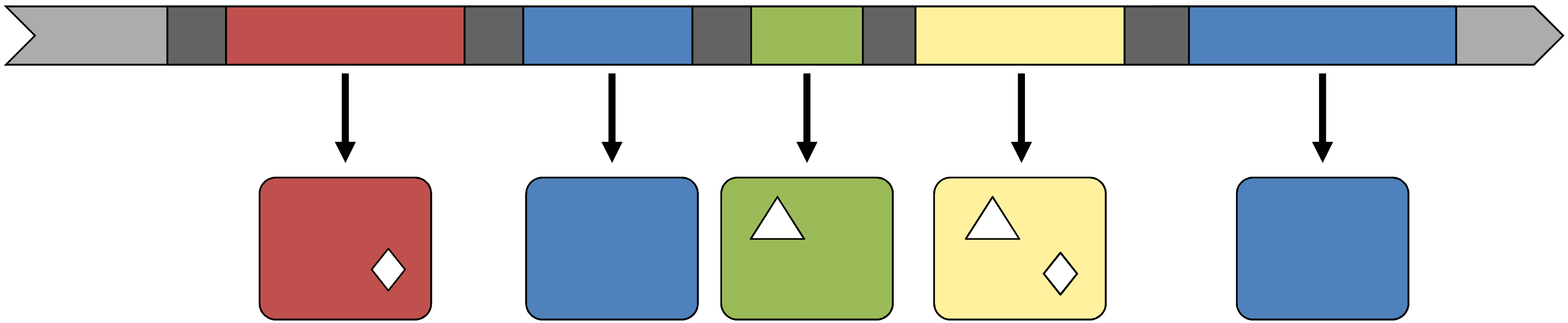
Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities



Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities

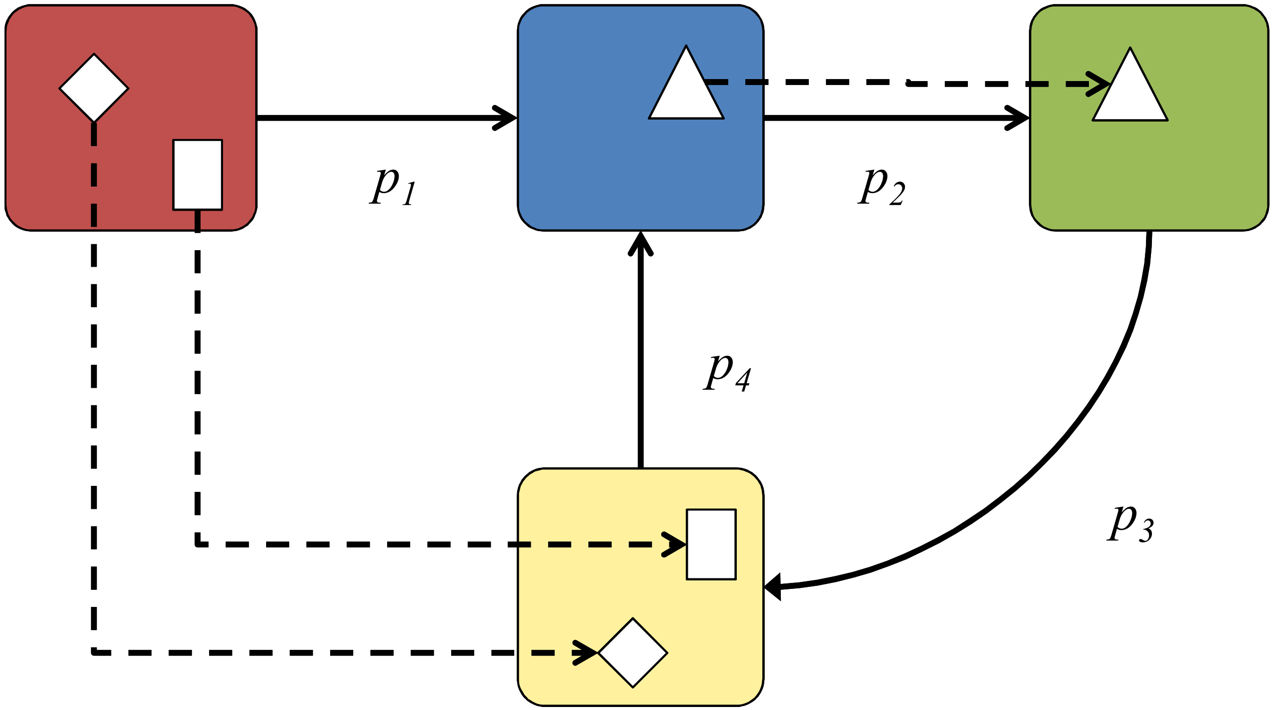


Tasks for proposed method

1. recognize regions of drift
2. represent density of data stream segments
3. identify recurrences on the density level
4. identify recurrences between parts of different densities

All done in an online fashion using a *bundle of methods*:
statistical tests like Wilcoxon, grouping variables by mutual information, graphical representation, ...

Result: Graph of Possible Worlds



- Pool of recurrent distributions plus recurrent “parts”
- Possible worlds connected by probabilistic transitions
- Queries over possible worlds require this structure plus inference on the densities (see (Geilke *et al.*, IEEE DSAA 2015))