FLOWREC

PROTOTYPING SESSION-BASED RECOMMENDER SYSTEMS IN STREAMING MODE

MALMÖ UNIVERSITET

Dimitris Paraschakis
dimitris.paraschakis@mau.se

Bengt J. Nilsson
bengt.nilsson.ts@mau.se

BACKGROUND

Recent trends in RecSys community

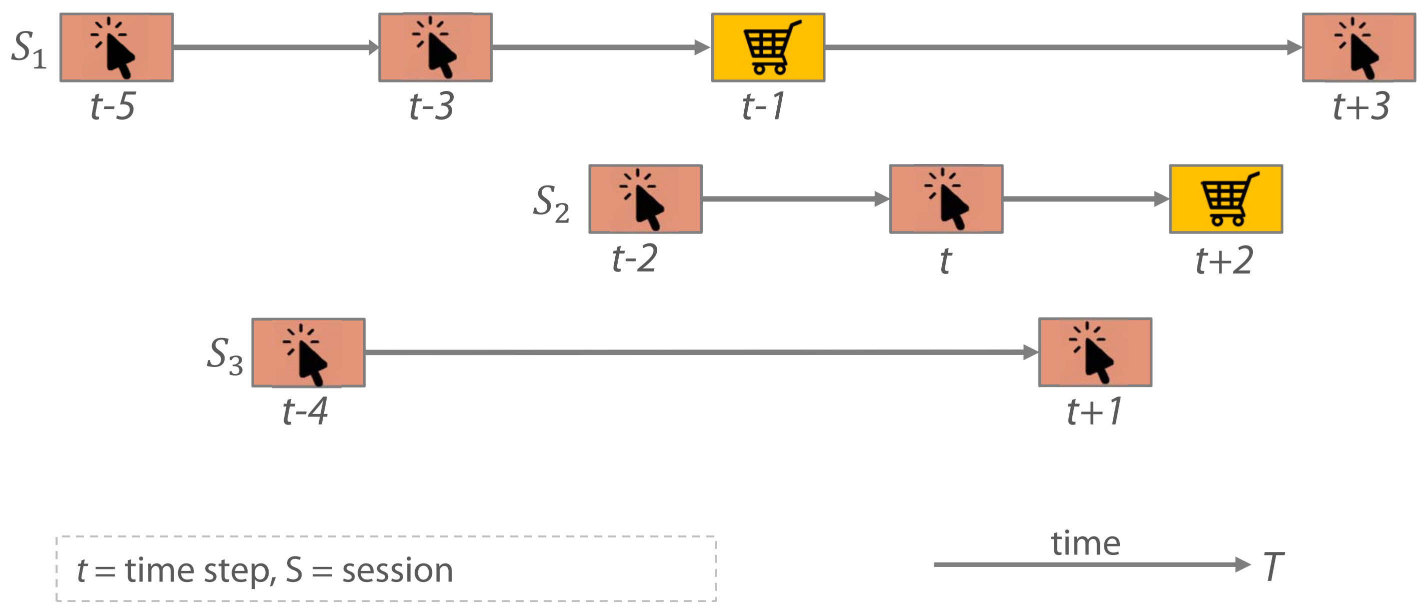
- Paradigm shift from matrix completion abstraction to sequence and session modelling
- Streaming recommendations

"As long as the matrix exists, the human race will never be free"

- Morpheus

BACKGROUND

- Sequence-aware recommendation: taking advantage of the sequential order of user interactions that are relevant in inferring his/her intent
- Very useful for session-based recommendations, where the goal is to find items that are relevant in the context of a current session



BACKGROUND

Streaming session data

- Temporally ordered
- Continuous
- High velocity
- Heterogeneous

Stream-based recommenders

- Incremental (sequential) updates
- Online learning, anytime prediction
- Below 100 ms / recommendation
- Event-specific training

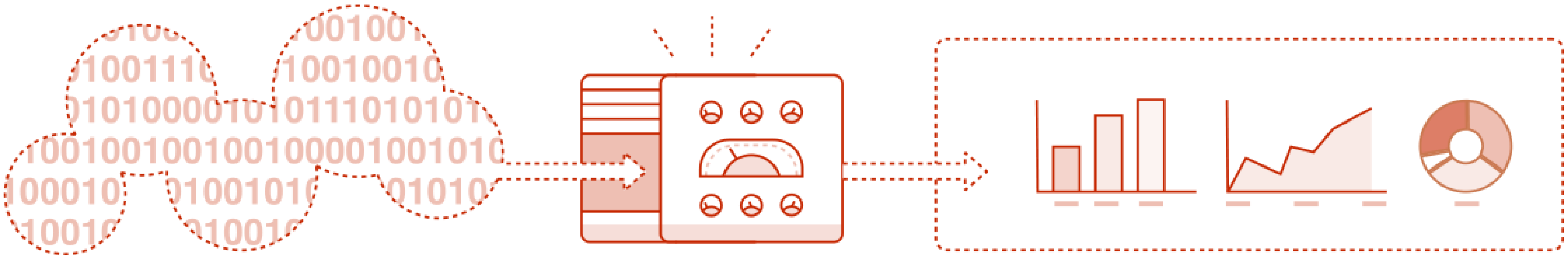
dimitris.paraschakis@mau.se

MOTIVATIONS

Current challenges

- Many existing recommender systems need pre- or re-training
- Dominated by RNN methods ➔ often outperformed by simpler models
- Lack of evaluation frameworks and standards for streaming session data
- Existing stream-based recommenders rely on matrix completion setups

Can we utilize existing data stream algorithms and frameworks for the recommendation task?



SCIKIT-MULTIFLOW

scikit-multiflow

A Python machine learning framework for data streams

- Contains stream learners, change detectors, and evaluation methods
- Supports multi-output/multi-label learning
- Inspired by **MOA** and **Meka** frameworks
- Follows **Scikit-learn**'s syntax and philosophy

How to adapt this framework for recommendations?

FlowRec

powered by Scikit-Multiflow

Extends Scikit-Multiflow for streaming session-based recommendations

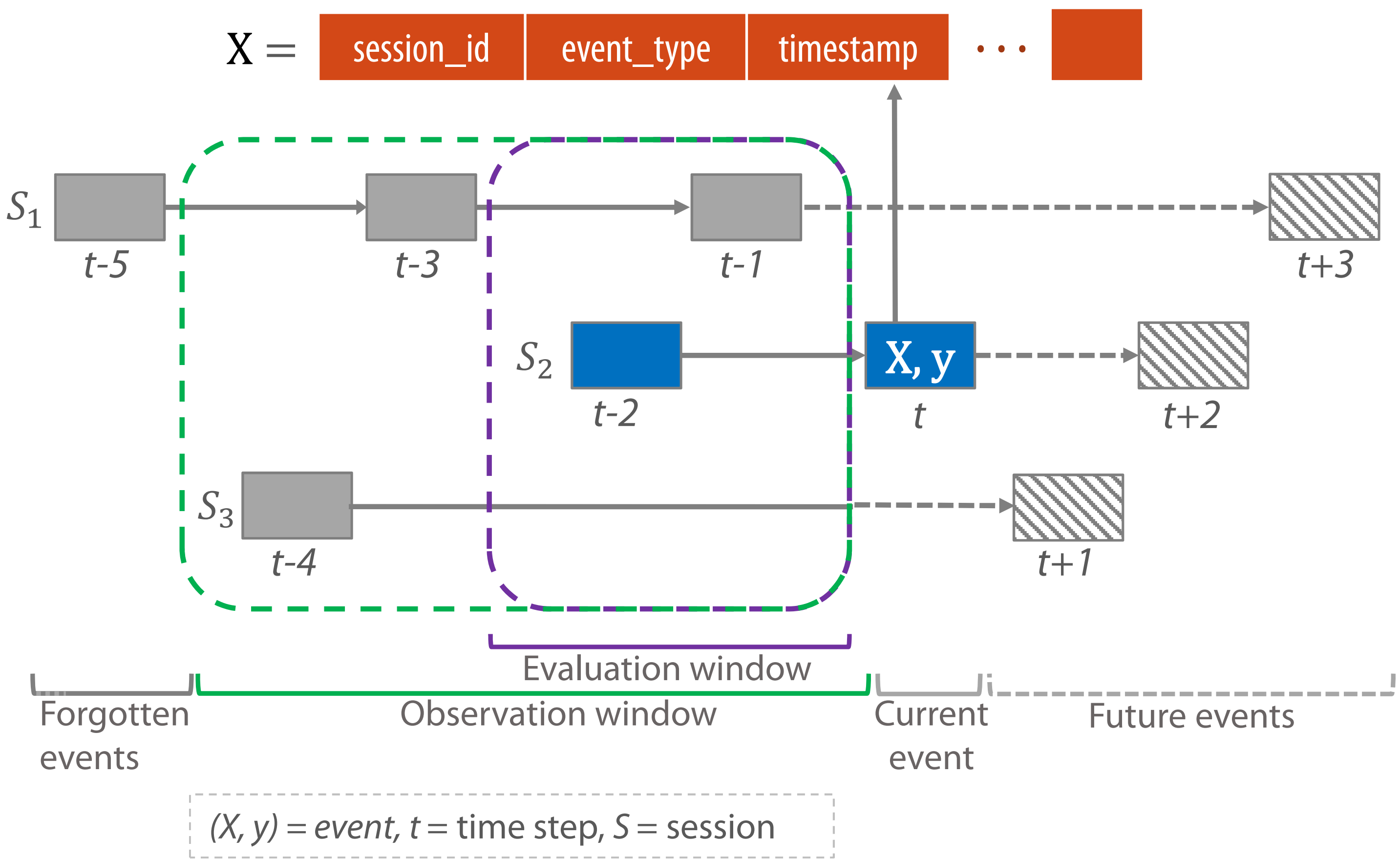
Features:

- Prequential evaluation
- A number of baseline recommenders + wrapper for Hoeffding Tree learner
- Support for multiple types of input data
- Adjustable sliding window for evaluation and training
- Support for reminders and repeated recommendations

FLOWREC

Problem setting: time series framed as supervised learning

- A stream of events of the form (X, y), where:
  y is an item *(hidden)*, and X is a context vector describing item y



dimitris.paraschakis@mau.se

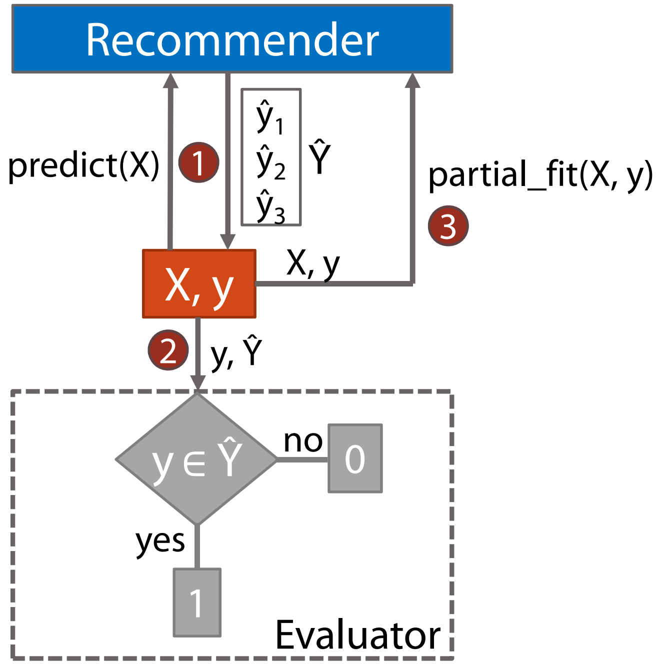
EVALUATION

Prequential evaluation protocol

Metrics:

$$\boldsymbol{Recall@N} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}(y_t = \hat{y}_i)$$

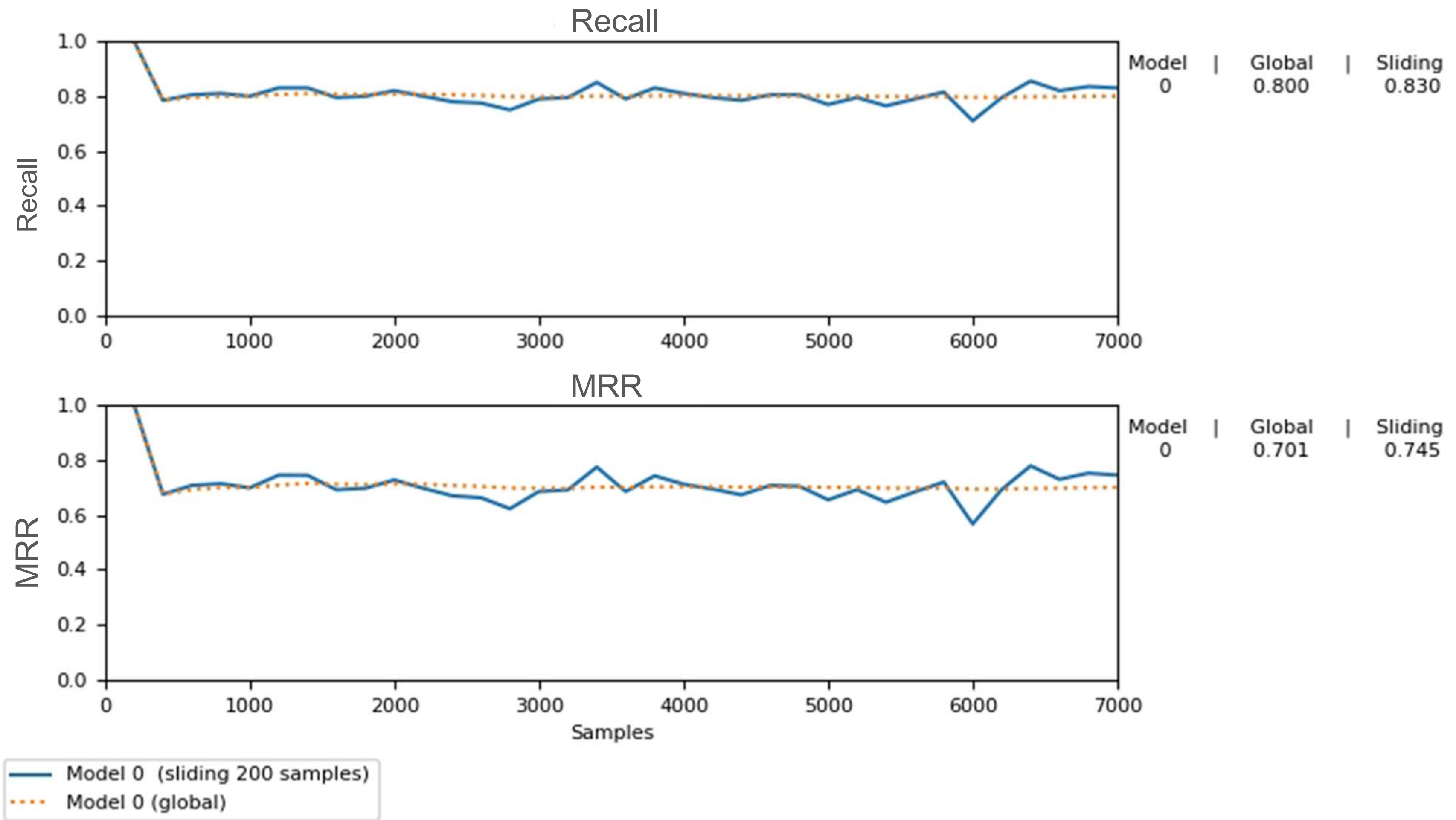
$$\boldsymbol{MRR@N} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{1}{i}\mathbb{I}(y_t = \hat{y}_i)$$



1. For each sample **(X,y)** arriving from the stream do:
2. Produce top-N recommendations given context **X**
3. Evaluate recommendations against item **y**
4. Update model with new observation **(X,y)**

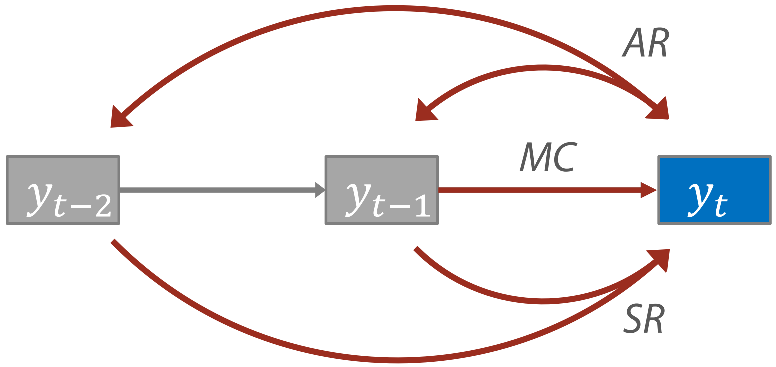
EVALUATION

Modes of evaluation: **global** (entire run), **sliding window** (most recent data)



BASELINES

- **Rule-based models**
  1. Association rules (AR): $y_{t'} \leftrightarrow y_t, t \neq t'$
  2. Markov chains (MC): $y_t \rightarrow y_{t+1}$
  3. Sequential rules (SR): $y_{t'} \rightarrow y_t, t > t'$, with weight $1/(t - t')$



- **Session-kNN** (Jannach & Ludewig, 2017)
- **BEER[TS]**: bandit ensemble (Brodén et al., 2019)
- Popularity baseline

PROTOTYPING

- Recommendations as a multi-class classification problem
- Each class is an item

How to build a recommender? Implement the following methods:

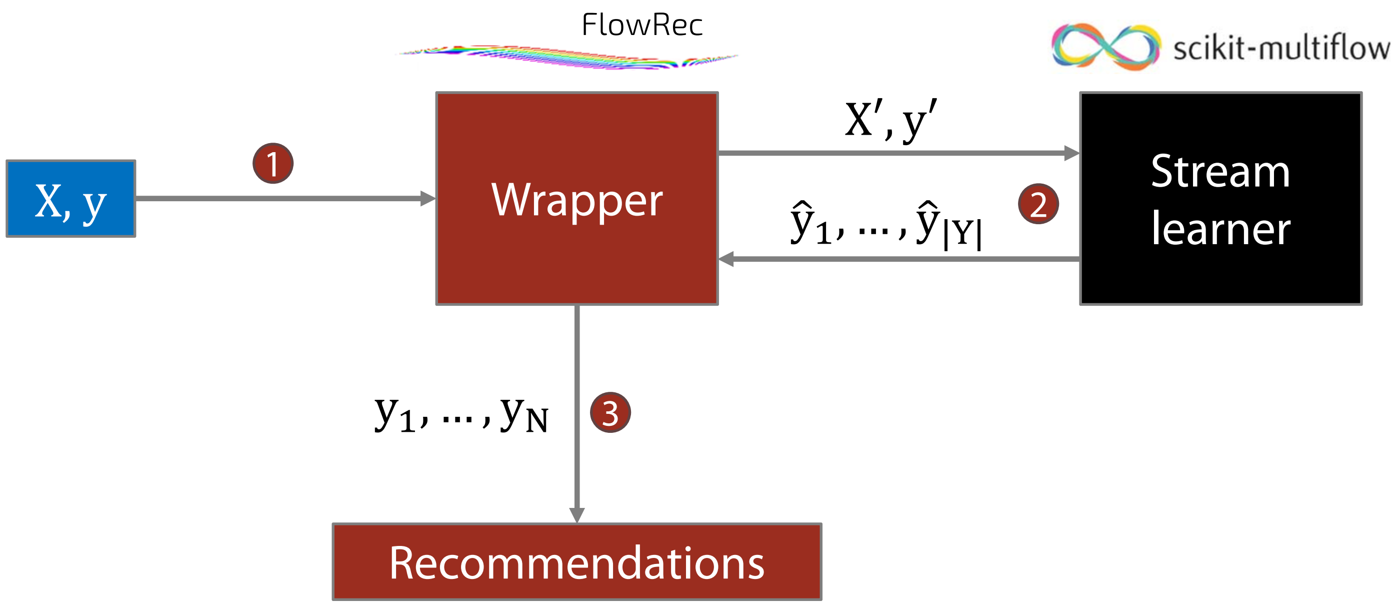
- **partial_fit(X, y)** – incrementally train a streaming model
- **predict(X)** – generate top-N predictions of the target's class
- **predict_proba(X)** – calculate the probabilities of a target belonging to each of the available classes (optional)

- Compliant with scikit learn

WRAPPERS

What does a **wrapper** do?

- Transforms a sample to the format accepted by a learner
- Calls the **predict_proba(X)** method of a learner and manipulates its return values to generate top-N recommendations



dimitris.paraschakis@mau.se

HT WRAPPER

Hoeffding Tree (HT)

scikit-multiflow

- Incremental, anytime decision tree inducer for data stream learning
- Naïve Bayes prediction at the leaves + sample weighting

HT wrapper

FlowRec

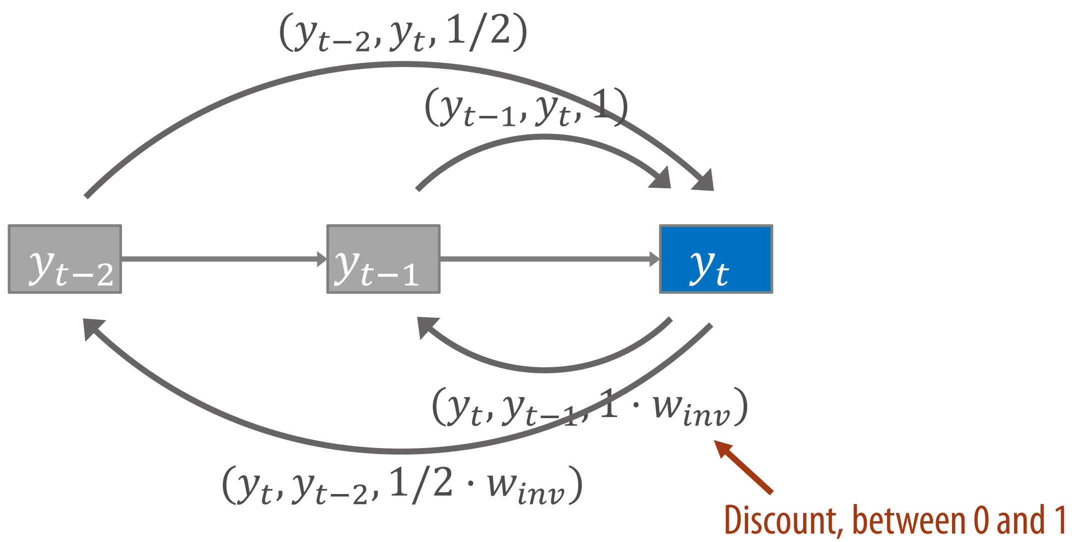
- Reduce HT to a decision stump, with nodes representing input items, and leaves containing item predictions obtained via Naïve Bayes
- Encode sequential and co-occurrence patterns of rule-based methods in a single learner
- Utilize the entire user session to make recommendations

HT WRAPPER

Training procedure

input (*feature*) output (*label*)

- At each time step t, call **partial_fit(item1, item2, weight)**
- Two weighted fits for each pair containing the current item:
  - Straight
  - Inverse

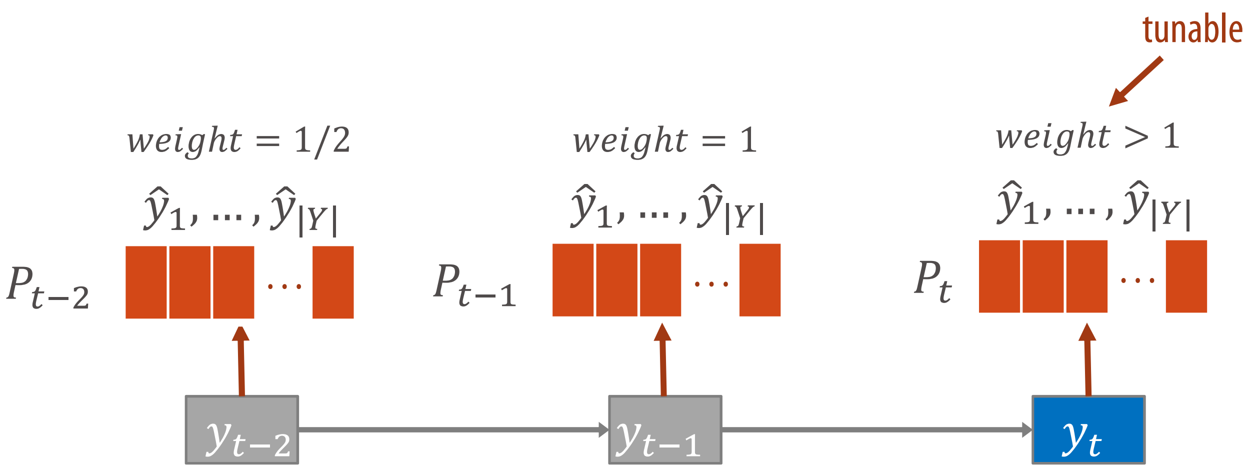


dimitris.paraschakis@mau.se

HT WRAPPER

Prediction procedure

- Combine the predictions of all tree nodes (i.e. items) in a session in an ensemble-like manner
- Invoke the **predict_proba(item)** method of Hoeffding Tree learner



Final prediction vector: $P = P_{t-2} + P_{t-1} + P_t$ → extract top-N items

EXPERIMENTS

Setup

- Next-item recommendation: predict item to appear in the next event
- Evaluation from pure cold-start, i.e. no model pre-training
- 1M events per simulation, 10K events for evaluation window, 50K events for training window, 100K events for hyperparameter tuning

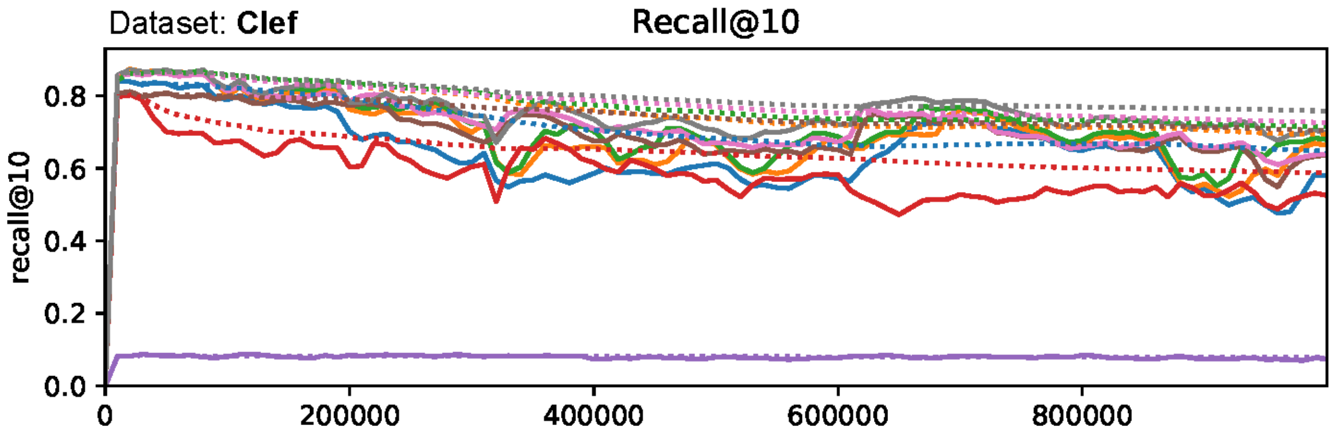
Datasets

Dataset	Contest	Domain (action)	Items	Sessions	Avg. session size
Clef	NewsReel '15	News (impressions)	109	305703	3.27 events
Yoochoose	RecSys '15	E-commerce (clicks)	21300	255166	3.92 events
Trivago	RecSys '19	Travel (clickouts)	243714	521677	1.92 events

dimitris.paraschakis@mau.se

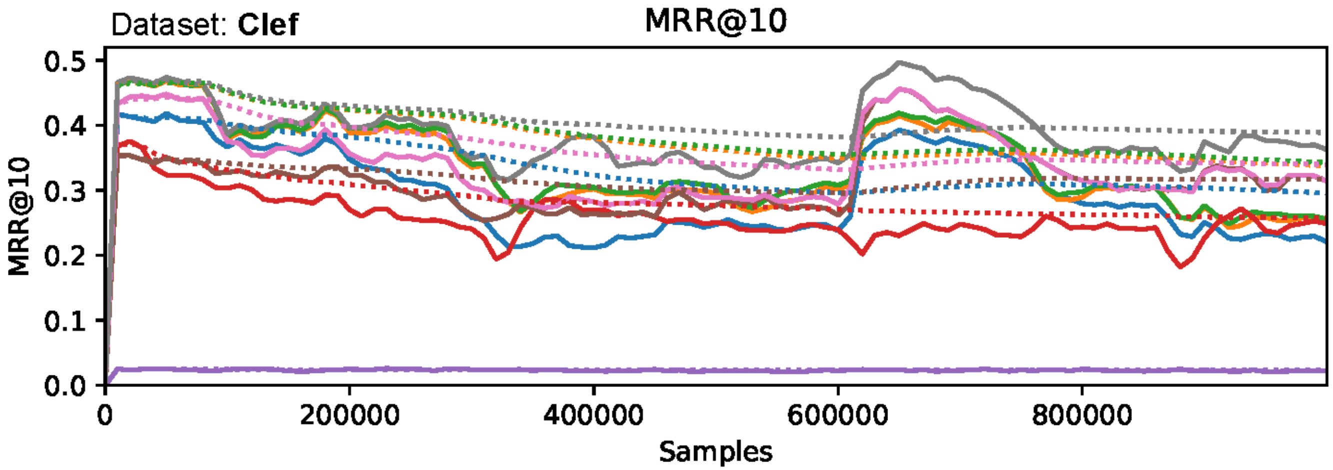
RESULTS

Clef (news articles)



Dataset: **Clef**

Model	Global	Sliding
AR	0.6483	0.5809
SR	0.6948	0.6636
MC	0.7157	0.6795
POP	0.5871	0.5256
RAND	0.0787	0.0747
S-KNN	0.7044	0.6376
BEER[TS]	0.7260	0.6396
HT *	0.7585	0.6961

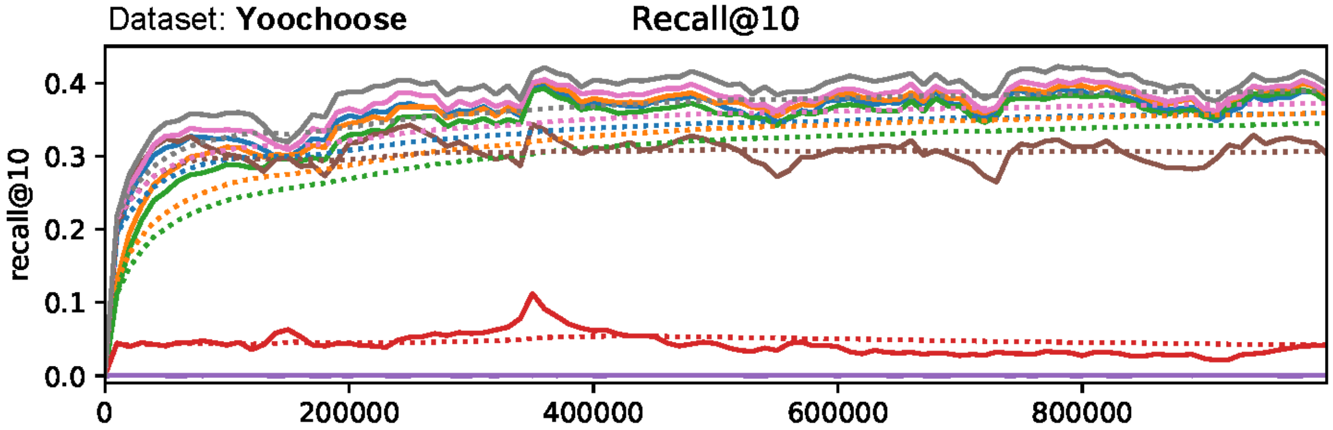


Dataset: **Clef**

Model	Global	Sliding
AR	0.2960	0.2208
SR	0.3375	0.2519
MC	0.3427	0.2557
POP	0.2578	0.2486
RAND	0.0230	0.0218
S-KNN	0.3173	0.3145
BEER[TS]	0.3398	0.3139
HT *	0.3890	0.3629

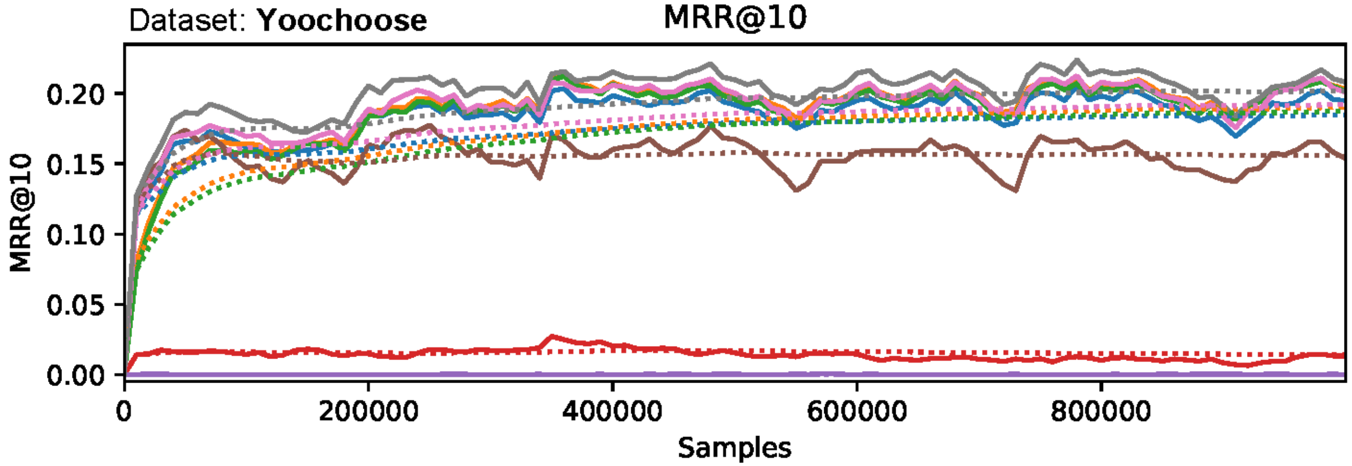
RESULTS

Yoochoose (e-commerce)



Dataset: **Yoochoose**

Model	Global	Sliding
AR	0.3586	0.3772
SR	0.3588	0.3830
MC	0.3450	0.3745
POP	0.0427	0.0417
RAND	0.0004	0.0003
S-KNN	0.3061	0.3037
BEER[TS]	0.3720	0.3867
HT *	0.3891	0.3989

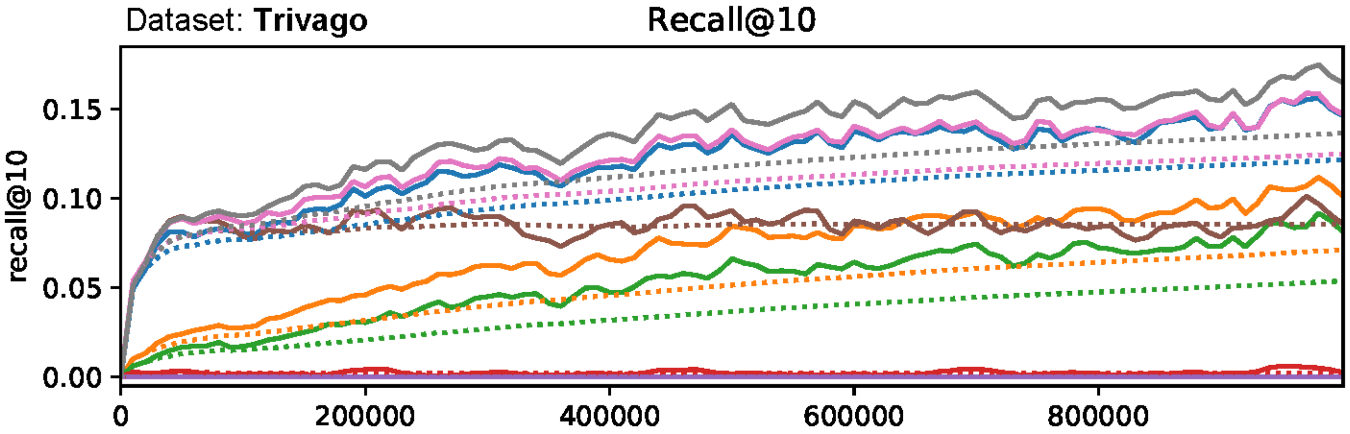


Dataset: **Yoochoose**

Model	Global	Sliding
AR	0.1847	0.1949
SR	0.1902	0.2031
MC	0.1874	0.2014
POP	0.0141	0.0132
RAND	0.0001	0.0000
S-KNN	0.1559	0.1539
BEER[TS]	0.1921	0.2010
HT *	0.2019	0.2080

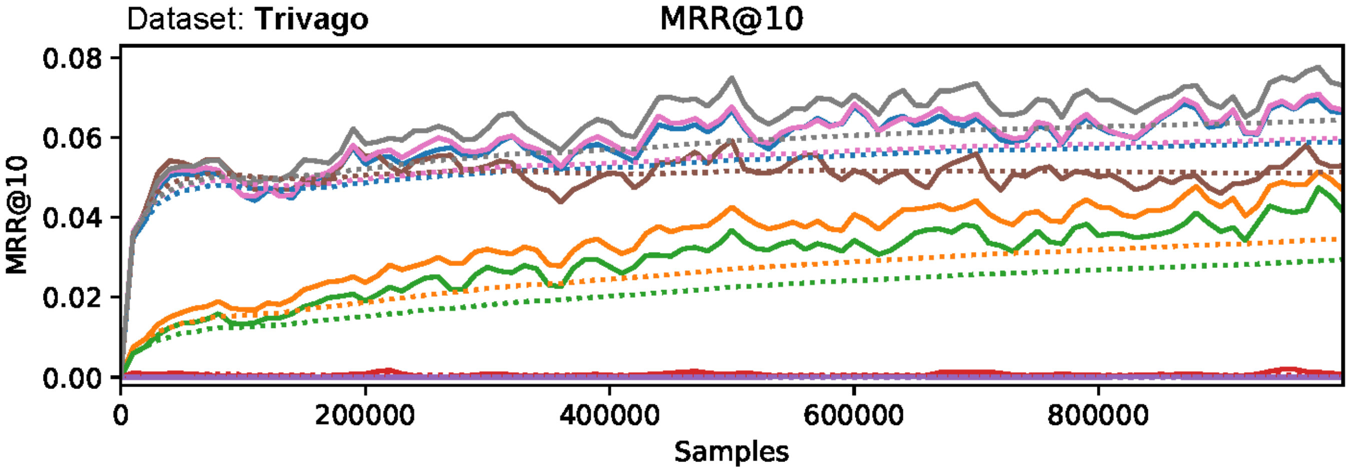
RESULTS

Trivago (travel)



Dataset: **Trivago**

Model	Global	Sliding
AR	0.1217	0.1463
SR	0.0712	0.1011
MC	0.0538	0.0812
POP	0.0023	0.0025
RAND	0.0000	0.0000
S-KNN	0.0857	0.0856
BEER[TS]	0.1250	0.1471
HT *	0.1367	0.1648



Dataset: **Trivago**

Model	Global	Sliding
AR	0.0589	0.0661
SR	0.0346	0.0469
MC	0.0294	0.0415
POP	0.0007	0.0006
RAND	0.0000	0.0000
S-KNN	0.0513	0.0529
BEER[TS]	0.0599	0.0669
HT *	0.0644	0.0730

dimitris.paraschakis@mau.se

Running time

Average recommendation time (*msec*) per model

Dataset	Model							
	AR	SR	MC	POP	RAND	S-kNN	BEER[TS]	HT
Clef	0.270	0.213	0.204	0.187	0.204	26.543	27.858	2.542
Yoochoose	0.744	0.623	0.476	3.365	0.313	2.434	13.331	9.629
Trivago	4.276	4.093	4.082	11.380	1.150	5.675	85.163	23.036

All runtimes below 100ms!

CONCLUSIONS

We propose

- **FlowRec**: a prototyping/benchmarking tool for streaming session-based recommendations
  - Built on top of **Scikit-Multiflow** → access to existing code base
  - Real-time monitoring of algorithm's performance over time
- **HT wrapper**: a stream-based recommender based on Hoeffding Tree
  - Simple yet effective ensemble-like algorithm combining sequential and co-occurrence patterns in a session

Future work

- Extend the library with additional recommendation algorithms, evaluation protocols and metrics

THANK YOU!

Questions?