NIPS 2018 AutoML for Lifelong Machine Learning Challenge

AutoGBT: Automatically Optimized Gradient Boosting Trees for Classifying Large Volume High Cardinality Data Streams under Concept Drift

Jobin Wilson[1,2], Amit Meher[2], Bivin Vinodkumar Bindu[1,2], Manoj Sharma[3], Vishakha Pareek[3] , Prof. Santanu Chaudhury[1] and Prof. Brejesh Lall[1]

[1]Indian Institute of Technology Delhi, [2]Flytxt, [3]CSIR-CEERI

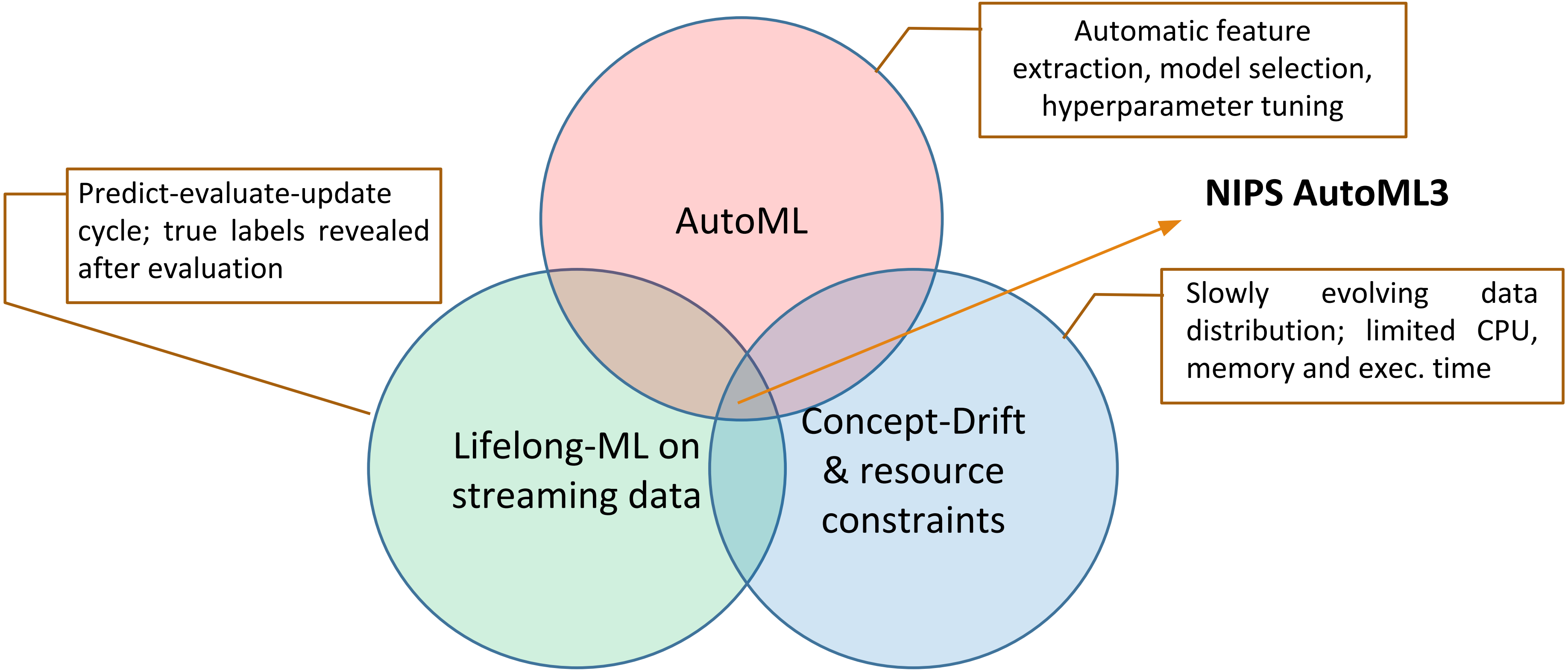
Agenda

- Problem Overview & Unique Aspects
- Existing Approaches
- AutoGBT Model Architecture
- Experiments & Results
- Conclusion

Motivation

- Lack of ML/domain experts; need self-maintaining autonomous end-to-end ML pipelines
- Real-world data arrives as streams/batches ordered in time
- Data distributions evolve; need to handle non-i.i.d data
  - Ability to retain knowledge, adapt to changes and transfer knowledge across time; subject to limited resources
- Relaxing AutoML constraints; large data volume, slow concept-drift, lifelong machine learning

Challenge Scope



Unique Aspects

- Algorithm scalability; varying data volume & feature count
- Varied features: numeric, categorical, multi-categorical, temporal & binary; missing values; dataset skew
- Categorical/multi-categorical features with large number of values, following power law
- Absence of domain information; slow concept-drift
- Resource constraints (CPU, RAM, Disk & time budget)
- Lifelong setting; predict-evaluate-update over multiple batches

Prediction Problem

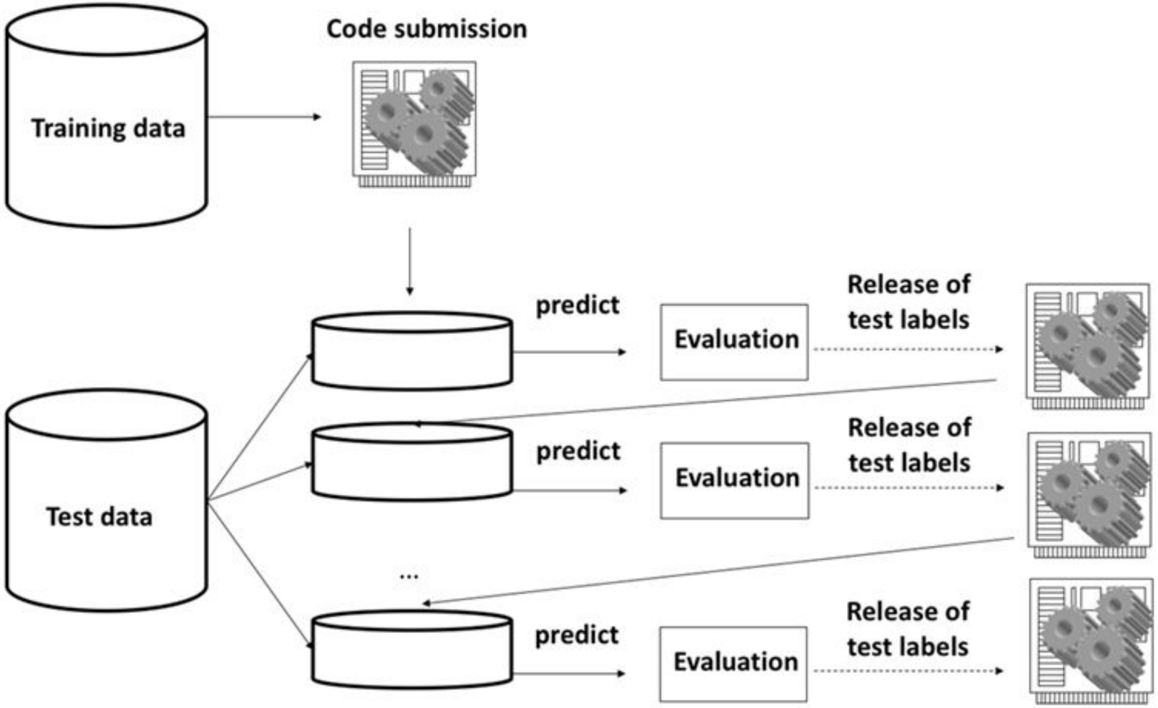


Image source : https://www.4paradigm.com/competition/nips2018

Evaluation Metric : Average AUC across all batches

Existing Approaches

- PoSH Auto-sklearn [2] (Feurer, Matthias, et al. 2018)
  - Automatically pre-selected portfolio, ensemble building and Bayesian optimization with successive halving
  - ML Pipeline optimization using Sequential Model Based Algorithm Configuration (SMAC)
  - Doesnt handle concept-drift, multicategorical features, categorical features with large number of unique values with power-law distribution
  - Not designed for very large data volume and lifelong learning setting

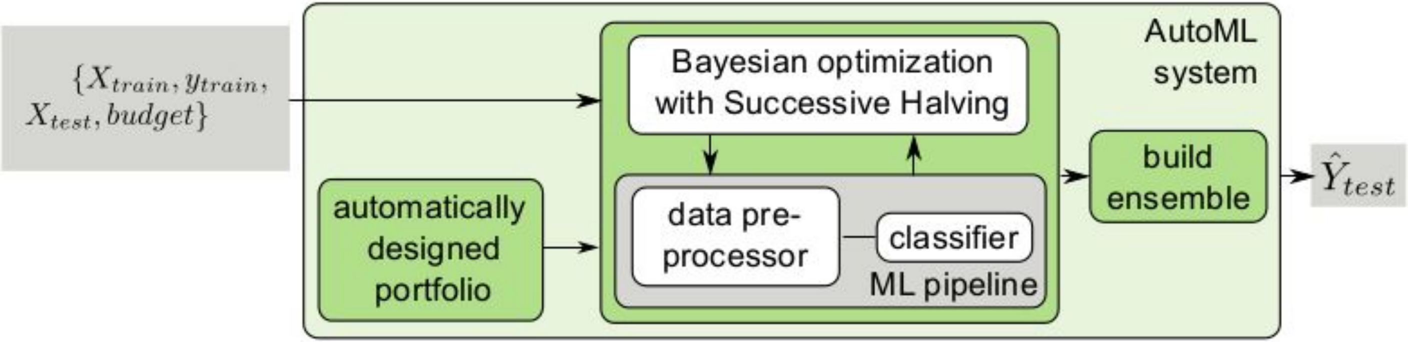


Image source:[2], Feurer, Matthias, et al. 2018

Existing Approaches

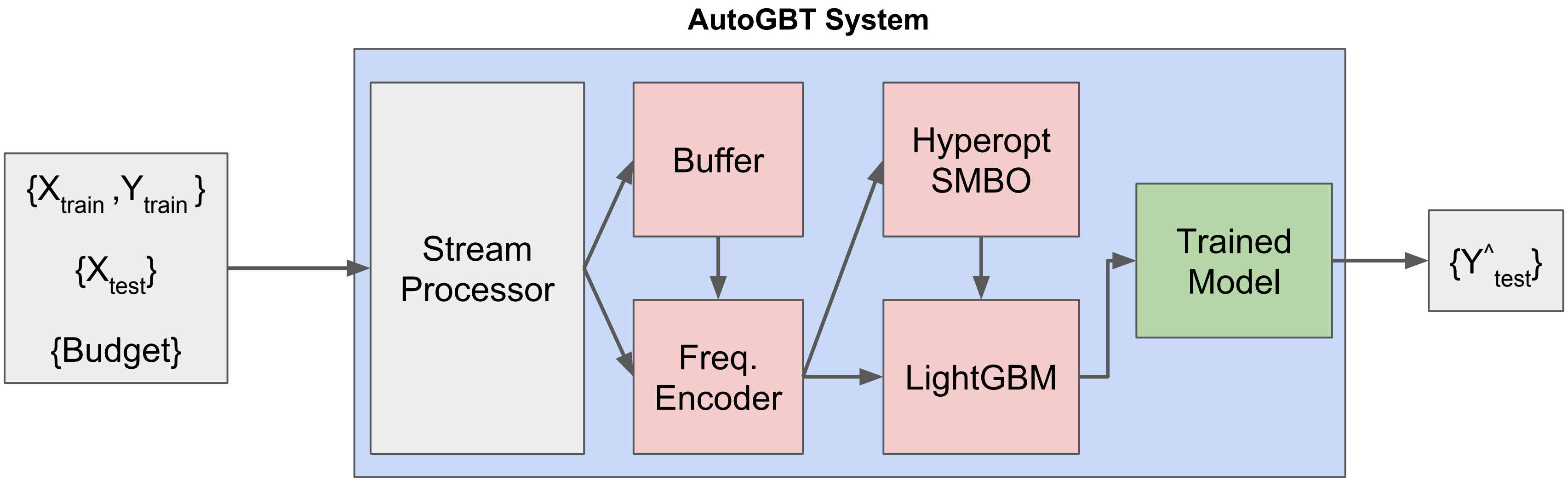
- "LML extension for Auto-sklearn" [1] (Jorge G. Madrid, et al. 2018)
  - Auto-sklearn extended by adding explicit drift detection using Fast Hoeffding Drift Detection Method (FHDDM)
  - Doesnt explicitly handle multi-categorical features and categorical features with large number of unique values with power-law distribution
  - Not designed for very large data volume; performance dependency on type of drift

Data: $D(X, y)$ examples
Take a batch D'_t of size n, $D'_t(X', y') \in D(X, y)$; $T_t \leftarrow$ learn a model with auto-sklearn using D'_t
while *there is data in D* **do**
  Take next batch D'_{t+1}; $\hat{y} \leftarrow$ Make predictions with T_t; **for** $y_j \in \hat{y}$ **do**
    drift_detected = Detector($y_j == y'_j \in y'$) //drift will be detected with model performance
  end
  if *drif_detected* **then**
    $T_{t+1} = adapt(T, D'_{t+1})$; Detector.reset(); $t = t + 1$
  else
    $T_{t+1} = T_t$; $t = t + 1$
  end
end

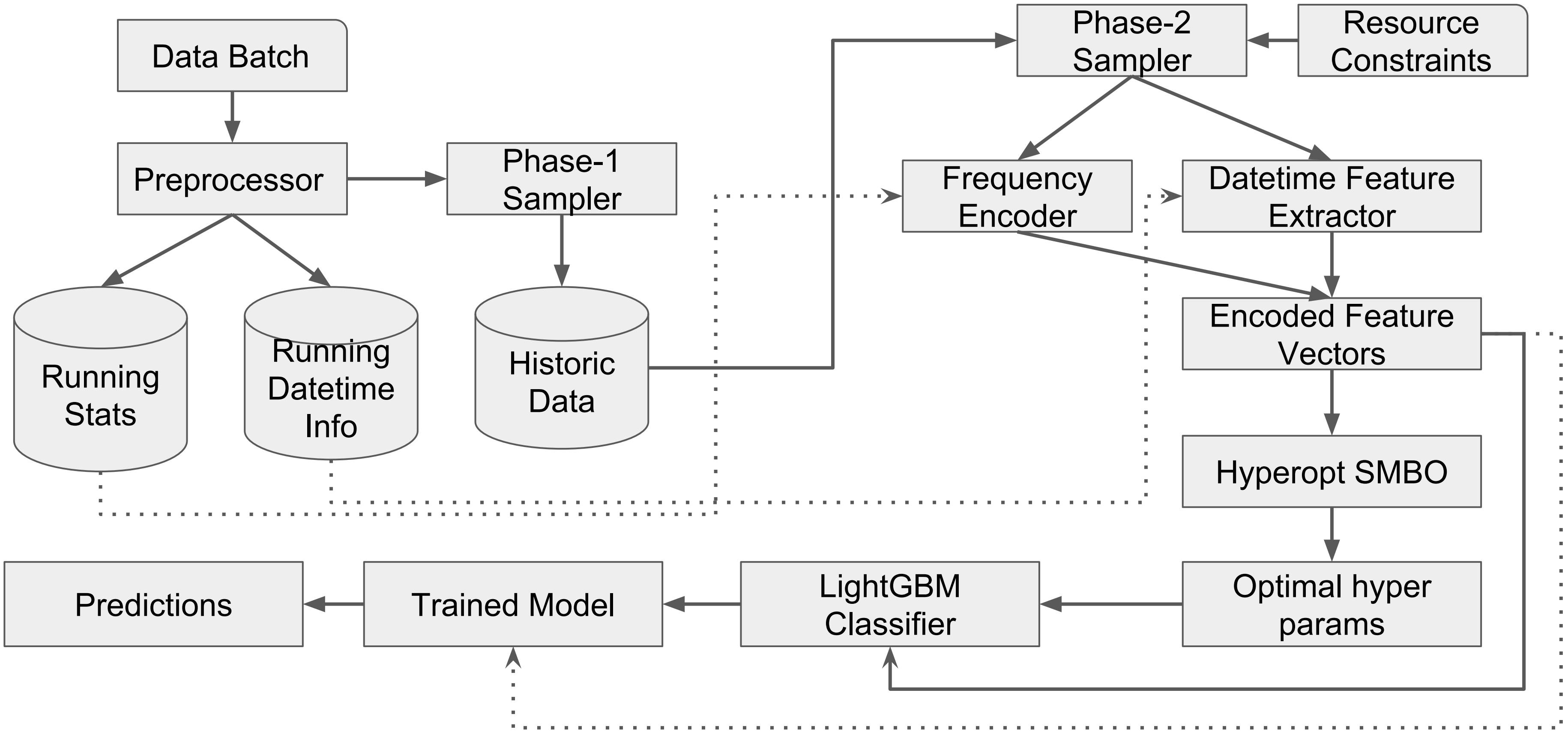
Source: [1], Jorge G. Madrid, et al. 2018

AutoGBT

- Automatically Optimized Gradient Boosting Trees
  - Frequency encoding allows to exploit semantic similarity across batches to counter slow concept-drift through adaptation, without explicit drift detection.
  - Stream processing, sampling strategy, single model portfolio & lazy model training significantly reduces resource and time budget requirements; scales to large data volume
  - ML Pipeline optimization using hyperopt SMBO



AutoGBT Model Architecture



Experiments & Results (Feedback Phase Datasets)

Dataset	Instances#	Features#	Cat#	Num#	MVC#	Time#	Duration (Sec)	Splits #	Avg. AUC
A	~10 M	82	51	23	6	2	2112.20	10	0.5171
B	~1.9 M	25	17	7	1	0	219.52	10	0.3088
C	~2 M	79	44	20	9	6	610.41	10	0.5645
D	~1.5 M	76	17	54	1	4	427.27	10	0.4779
E	~17 M	34	25	6	1	2	1178.25	10	0.7273

Note: Based on logs from codalab server (4 Cores, 16 GB RAM); leaderboard position = 7th in feedback phase

Experiments & Results (Other Datasets)

Dataset	Instances#	Features#	Cat#	Num#	Time#	Duration (Sec)	Splits #	Avg. AUC
Avazu CTR Prediction	~40.42 M	22	21	0	1	950.68	9	0.4517
Amazon Challenge	~0.03 M	9	8	1	0	54.14	5	0.5757
Airline Dataset	~1.07 M	29	16	13	0	136.65	6	0.5175
Bank Marketing Dataset	~0.04 M	20	10	10	0	48.24	5	0.7303
RI-AutoML3 starter kit	~0.05 M	22	14	8	0	72.2	4	0.3974
Ada-AutoML3 starter kit	~0.17 M	48	0	48	0	61.71	4	0.8492

Note: Executed locally using codalab docker image on a workstation with 20 cores, 64 GB RAM

AutoML Phase Rankings

Bundle		Avg. rank	Set 1	Set 2	Set 3	Set 4	Set 5	Duration
Py3	autodidact.ai	2.2	2	4	1	2	2	5882.13
Py3	*harryfootball*	2.4	3	1	2	1	5	8700.47
Py3	*fanqiechaodan*	4.2	4	6	4	3	4	7912.14
Py3	*Ml-Intelligence*	4.2	1	3	6	10	1	9426.68
Py3	*linc326*	4.6	6	5	5	4	3	8843.15
Py3	*rcarson*	6.4	9	2	7	6	8	5471.59
Py3	*jimliu*	7.8	5	8	15	5	6	5581.74
Py3	*PGijsbers*	8.4	7	7	14	7	7	10427.18
Py3	*gxr_6666*	11.4	13	14	10	8	12	6674.08
Py2	*pipi_*	12	10	10	8	18	14	8334.86
Py3	*Jie_NJU*	12	11	23	3	13	10	8282.08
Py2	*nomo*	15	14	15	18	12	16	4165.99
Py3	*mlg.postech*	16.2	12	26	9	23	11	7357.6
Py2	*Cheng_Zi*	18.4	19	17	19	22	15	4532.53

Conclusion

- A simple adaptive pipeline having automatic hyperparameter tuning using SMBO is performing well, even though the model portfolio is limited
- Specialized preprocessing pipeline is essential to handle large number of categorical/multicategorical values following a power law distribution
- Our save-retrain strategy & frequency encoding performed reasonably well to counter slow concept-drift, even in the absence of explicit drift detection
- Heuristic checks proved to be helpful to handle budget constraints in our local experiments; its effect on blind phase performance is yet to be studied

References

1. Jorge G. Madrid et al. Towards AutoML in the presence of drift: First results. In: ICML 2018 Workshop on AutoML
2. Feurer, M. et al. Practical Automated Machine Learning for the AutoML Challenge 2018., In: ICML 2018 Workshop on AutoML
3. Feurer, Matthias, et al. Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems. 2015.
4. Ganin, Yaroslav, and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation." arXiv preprint arXiv:1409.7495 (2014).
5. Bergstra, James at al. "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms." Proceedings of the 12th Python in Science Conference. 2013.
6. Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in Neural Information Processing Systems. 2017.

Thank You