AMAZON RESEARCH DAYS

A DIVERSE **NETWORK**
A UNITED COMMUNITY

Squeezing the last DRiP:

AutoML for Cost-constrained Product Classification

Abhishek Divekar
India Machine Learning, Amazon

For questions, please contact adivekar@amazon.com

Background

- Automated Machine Learning (AutoML): Given a training set, automatically discovers best model / ensemble.
  - Useful for non-tech users (PMs, associates) and for scientists to quickly establish benchmarks.
  - First proposed by Auto-WEKA (2013) as “combined algorithm selection and hyperparameter optimization” (CASH) problem.
- Neural Architecture Search (NAS): finds good NNs automatically.
  - Very expensive.
  - **CASH is still king** for most practical AutoML use-cases.

Background

- Popular AutoML systems use tricks to automatically build ML pipelines:
  - Auto-WEKA (Thornton et al., 2013): automatic hyperparameter tuning
  - TPOT (Olson et al., 2016): flexible expression-tree + genetic programming
  - AutoGluon (2019): multi-layer stacking ensemble
  - AutoSklearn2.0 (Feurer et al., 2020): portfolio-based meta-learning, budget allocation using successive halving
  - H2O.ai AutoML (2019): multiple tricks, can run on most cloud platforms
  - Many more (Auto-keras, mljar, hyperopt-sklearn, Ludwig, …)
- Popular Benchmarks:
  - OpenML AutoML Benchmark
  - ChaLearn AutoML Challenges

Unmasking the problem

- Current AutoML systems: reduce cost of **discovery** process
  - e.g. “find best model within 30 mins”
- Most models are trained, used in production for months, refreshed frequently.
  - Costs of running model in production > one-time cost to discover best model > periodic refresh cost.
- Heuristics to reduce inference cost:
  - Distillation (model distillation, knowledge distillation for deep nets)
  - Reduce training time limit (leads to smaller ensembles)
- **Question: Can AutoML systems discover high-performing models under explicit cost budget?**
  - “Cost” might be inference latency, RAM used, dollar value for loss in recall.

Unmasking the problem

- Compared to FastText, AutoGluon gives 120 bps ROC-AUC uplift on Amazon product classification datasets*
- …but, AutoGluon incurs **~40x inference cost for single-prediction use-cases** (e.g. predicting Alexa utterances) and ~8x inference cost for batch of 10,000.

Algorithm	Mean ROC-AUC	Rank (Champion)	Per-row inference time (ms/row) and cost ($/MM) $B = 1$		$B = 10^2$		$B = 10^4$	
FastText	96.98±2.37	6.97 (1)	**46.1**	**$2.95**	**1.6**	**$0.11**	0.51	**$0.03**
VowpalWabbit	97.54±2.16	5.21 (2)	70.2	$4.48	2.0	$0.13	0.73	$0.05
WideAndDeep	97.20±2.62	5.97 (2)	244.3	$15.61	3.7	$0.23	0.50	**$0.03**
XGBoost	97.87±1.85	4.105 (3)	87.3	$5.57	26.2	$1.67	1.29	$0.08
BERT-B	97.28±2.28	6.23 (0)	69.9	$14.29	8.3	$1.69	8.74	$1.79
ELECTRA-S	96.88±2.17	7.92 (1)	66.8	$13.65	3.1	$0.63	1.95	$0.4
DeBERTa-L	94.00±9.27	6.55 (4)	99.9	$20.43	29.8	$6.09	29.7	$6.08
RoBERTa-L	94.47±11.61	6.29 (2)	79.6	$16.27	23.5	$4.81	23.7	$4.85
AutoGluon	**98.18±1.67**	**2.21 (23)**	2465.2	$157.50	27.9	$1.78	3.83	$0.24
AutoGluon-D	97.54±1.91	5.71 (0)	387.5	$24.75	6.3	$0.40	2.10	$0.13

*Average performance numbers from a set of product classification tasks, not representative of production environment.

Credit to: Andrew Borthwick, Oleg Kim, Fayaz Ahmed Farooque, Ethan Xu, Kate Kang, Kee Kiat Koo

2021 AMAZON RESEARCH DAYS

DRiP Framework

(Discover-Refine-Productionize)

A. Divekar, G. Manchanda, P. Raj, A. Das, K. Tanwar, A. Jagatap, V. Puranik, J. Srinivasan, R. Nalam, N. Rasiwasia

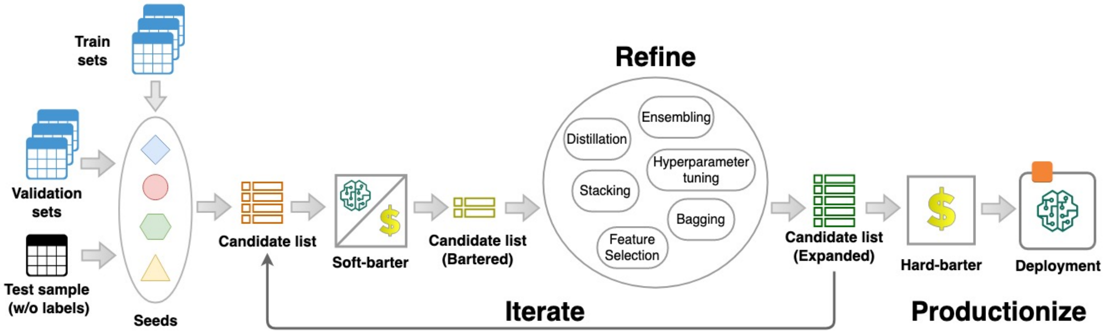
India Machine Learning, Amazon

For questions, please contact adivekar@amazon.com

DRiP framework

(Discover-Refine-Productionize)

1. Train a **seed set** of candidate models
2. Iteratively **refine** (i.e. apply ML techniques to current candidates) and **barter** (filter bad candidates) to improve cost-performance tradeoff of the current list of candidates
3. Select the final candidate as having the lowest K-fold cross-validation loss, and cost within budget α (**hard-barter**). Cost is measured on unlabelled sample of test universe, $\mathcal{D}_{unlb}$
4. Finally, **Productionize** selected model.



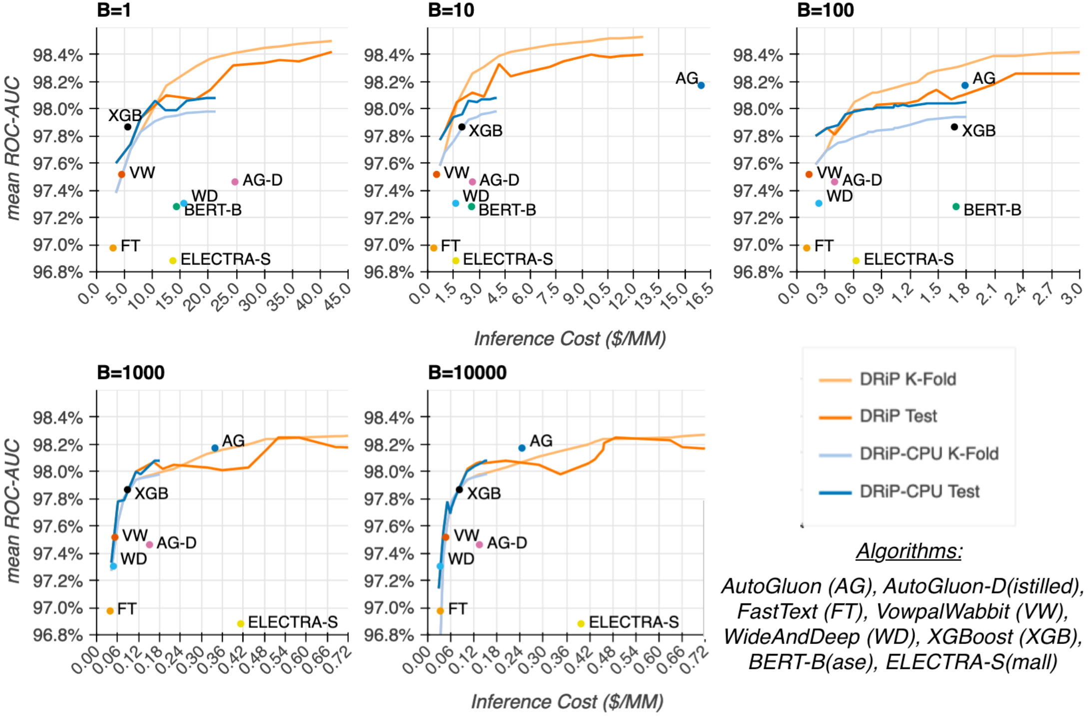
DRiP AutoML system

- Our implementation of DRiP framework.
- Select **heterogeneous seed algorithms**.
    - CPU-based: XGBoost, VowpalWabbit, FastText,
    - GPU-based (Text Transformers): BERT, RoBERTA, DeBERTa, ELECTRA
- Use cost-optimized ML techniques:

1. **Pretrain** BERT, ELECTRA on Amazon Product text.
2. **Feature Ranking:** reduce feature space (and cost) with min. performance loss.
3. **Bayesian Optimization:** automatically find hyperparams with high performance.
4. **Cost-constrained ensembling:** combine probability scores of candidates, select best combination **within budget** by *K*-fold performance.

Results

- DRiP produces a **cost-performance tradeoff curve*** for users to choose the appropriate point.
- Single algorithms and other AutoML systems only produce a single point on this curve.
- As seen, inference cost varies widely for different batch-sizes/use-cases.
- By accepting a user-defined budget, DRiP can find the best model within the budget (materializes a single point on the curve) for that batch size.



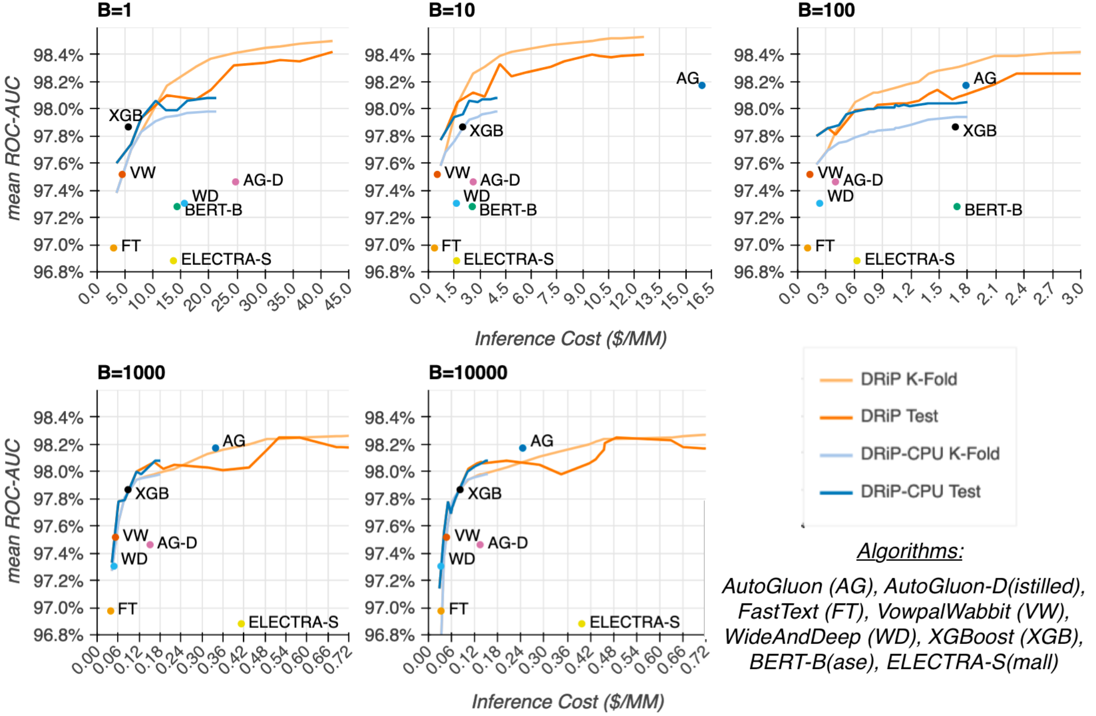
Algorithms:

AutoGluon (AG), AutoGluon-D(istilled), FastText (FT), VowpalWabbit (VW), WideAndDeep (WD), XGBoost (XGB), BERT-B(ase), ELECTRA-S(mall)

*Average performance numbers from a set of product classification tasks, not representative of production environment.

Credit to: Andrew Borthwick, Oleg Kim, Fayaz Ahmed Farooque, Ethan Xu, Kate Kang, Kee Kiat Koo

Results



Algorithms:

AutoGluon (AG), AutoGluon-D(istilled), FastText (FT), VowpalWabbit (VW), WideAndDeep (WD), XGBoost (XGB), BERT-B(ase), ELECTRA-S(mall)

Compared to AutoGluon (AG)*:

- **On-par performance at low cost:**
  DRiP achieves 67% reduction in inference cost at 4bps drop in performance compared to AutoGluon (mean ROC-AUC of 98.14 vs 98.18).
- **Minimizing cost:**
  Low-budget DRiP requires just 44% inference cost, yet improves ROC-AUC by 21bps compared to distilled AG (97.75 DRiP vs 97.54 AG-Distilled).
- **Maximum performance:**
  Without a budget, "Unrestricted" DRiP delivers **98.42 ROC-AUC** vs 98.18 for AG (24bps lift).

*Average performance numbers from a set of product classification tasks, not representative of production environment.

Credit to: Andrew Borthwick, Oleg Kim, Fayaz Ahmed Farooque, Ethan Xu, Kate Kang, Kee Kiat Koo

Conclusion

Cost-constrained AutoML is possible and crucial for business use-cases which are sensitive to cost.

Future work:

- Reduce cost of discovery process (multi-objective cost; bartering schemes).
- Faster Transformer alternatives for larger batch sizes.
- Benchmark on external multiclass/regression problems.
- Alternative ensemble methods (e.g. stacking, bagging, boosting).

2021 AMAZON RESEARCH DAYS

Thank you!

For questions, please contact adivekar@amazon.com

A DIVERSE **NETWORK**
A UNITED COMMUNITY

**AMAZON
RESEARCH
DAYS**