How Developers Iterate on Machine Learning Workflows

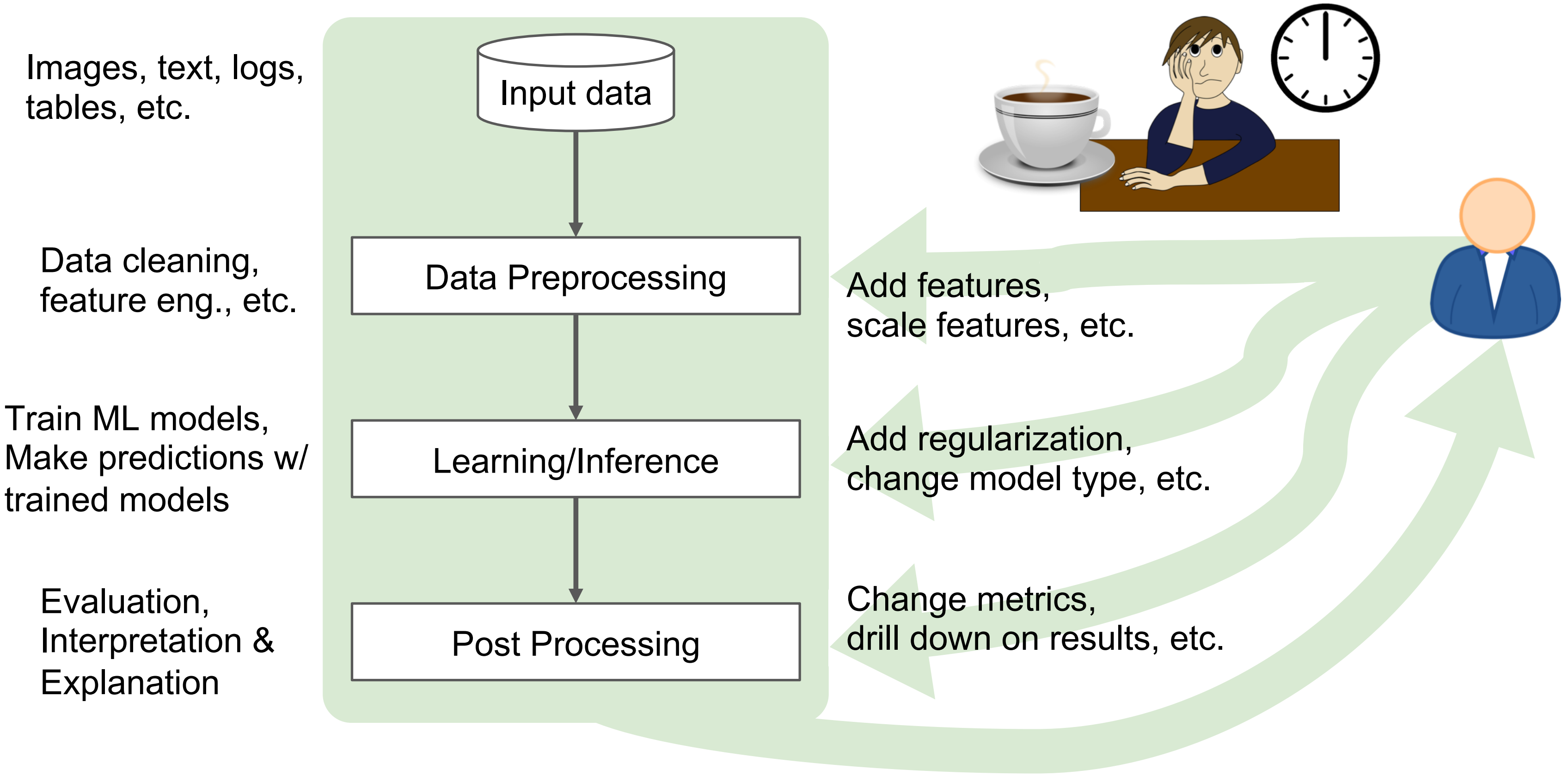
-- A Survey of the Applied Machine Learning Literature

Doris Xin[1], Litian Ma[1], Shuchen Song[1], Rong Ma[2], Aditya Parameswaran[1]

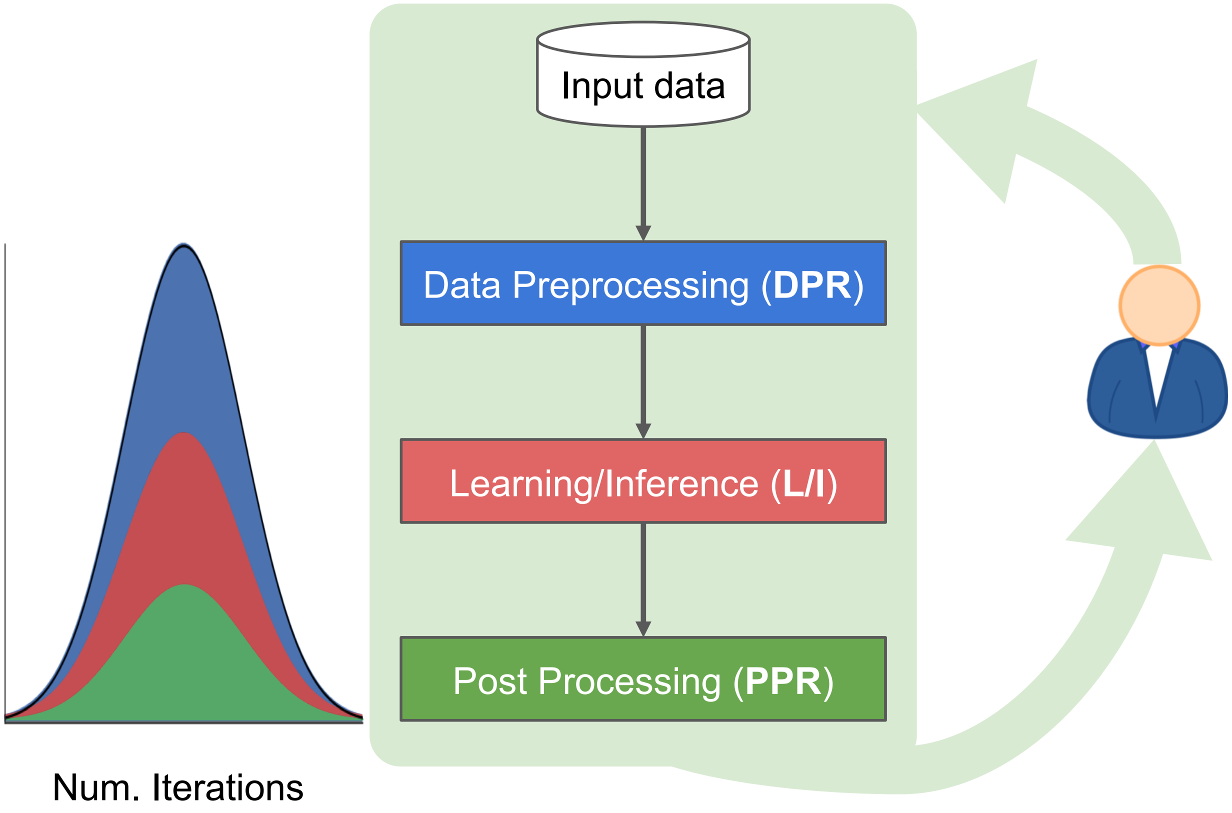
[1] University of Illinois at Urbana-Champaign

[2] Peking University

Developing Machine Learning Applications is **Iterative**



Developing Machine Learning Applications is **Interactive!**

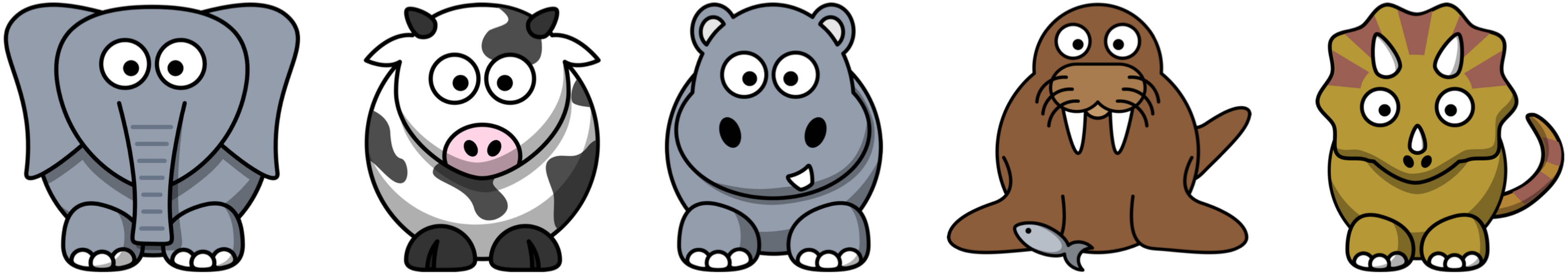


Creating systems to enhance interactivity requires ***a statistical characterization of how developers iterate on ML workflows***.

How Do Developers Iterate on Machine Learning Workflows?

How Do Developers Iterate on Machine Learning Workflows?

Our approach: study iterations by collecting statistics from applied ML papers **grouped by application domains**.



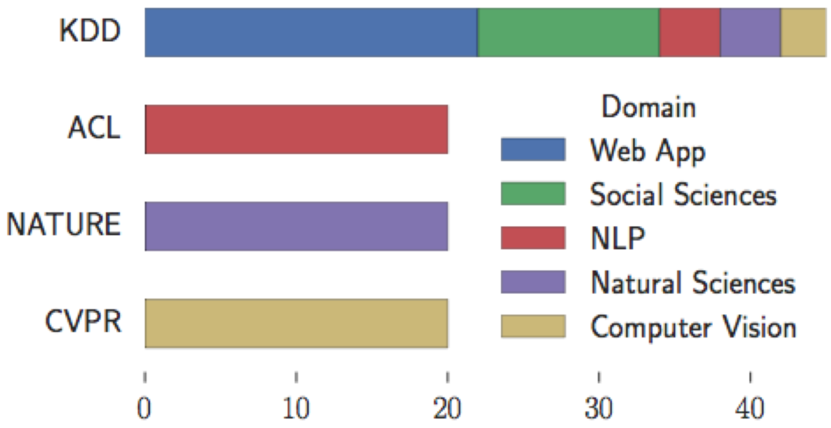
Outline

- Data & Limitations
- Methodology
  - Statistics
  - Estimation
- Results
- Conclusion & Future Work

Outline

- **Data & Limitations**
- Methodology
  - Statistics
  - Estimation
- Results
- Conclusion & Future Work

Corpus: 105 Papers from 2016



Limitations

- Incomplete picture of iterations
  - Focus on ML and omit DPR
- Results presented side-by-side
  - Can’t determine the order
- # papers / domain is small
  - May lead to spurious results

Remedies

- Multiple surveyors to reduce chance of spurious results
- Iteration estimators that do not rely on order

Outline

- Data & Limitations
- **Methodology**
  - Statistics
  - Estimation
- Results
- Conclusion & Future Work

Collecting Statistics

	Data Prep.			ML Model Class			ML Tuning			Evaluation Metrics			
	norm.	impute	...	LSTM	SVM	...	Reg.	Learn. Rate	...	AUC	...	# tables	# figs
												5	2
												5	2
												5	2
Aggregate												5	2

Open source dataset at https://github.com/helix-ml/AppliedMLSurvey

Estimating Iterations

	Data Prep.			ML Model Class			ML Tuning			Evaluation Metrics			
	norm.	impute	...	LSTM	SVM	...	Reg.	Learn. Rate	...	AUC	...	# tables	# figs
Aggregate												5	2

Number of data prep. iterations t_{DPR}

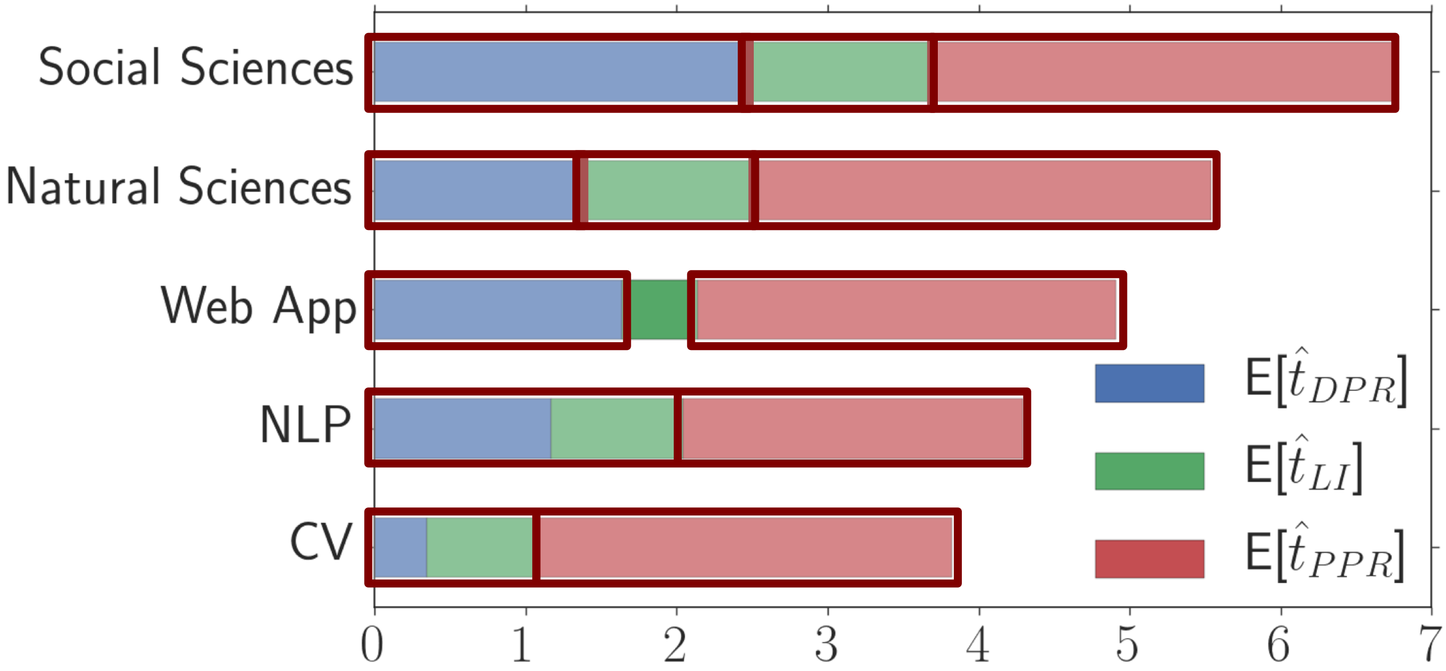
Number of ML iterations t_{LI}

Number of post proc. iterations t_{PPR}

Outline

- Data & Limitations
- Methodology
  - Statistics
  - Estimation
- **Results**
- Conclusion & Future Work

Mean Iteration Count by Domains



Data Preprocessing

Social Sciences	Natural Sciences	Web Apps	NLP	Computer Vision
Join (31.0%)	**Feat. Def.** (40.6%)	**Feat. Def.** (36.1%)	**Feat. Def.** (32.1%)	**Feat. Def.** (37.5%)
Feat. Def. (27.6%)	Univar. FS (18.8%)	**Join** (22.2%)	**BOW** (17.9%)	**BOW** (25.0%)
Normalize (17.2%)	**Normalize** (12.5%)	**Normalize** (13.9%)	**Join** (14.3%)	Interaction (25.0%)
Impute (6.9%)	PCA (9.4%)	Discretize (8.3%)	**Normalize** (10.7%)	**Join** (12.5%)

- Feat. Def. = human defined features from raw attributes
  - e.g. adult=true if age >=18

ML Model Classes

Social Sciences	Natural Sciences	Web Apps	NLP	Computer Vision
GLM (36.0%)	SVM (32.7%)	GLM (37.0%)	RNN (32.4%)	CNN (38.2%)
SVM (28.0%)	GLM (15.4%)	SVM (11.1%)	GLM (14.7%)	SVM (17.6%)
RF (20.0%)	RF (13.5%)	RF (11.1%)	SVM (11.8%)	RNN (17.6%)
Decision Tree (12.0%)	DNN(13.5%)	Matrix Factor. (11.1%)	CNN (8.8%)	RF (5.9%)

- Generalized linear models: logistic regression, linear regressions, etc.
- SVMs are popular (especially in natural sciences!) possibly due to kernels
- Deep learning is only popular in NLP and computer vision so far

ML Model Tuning

Social Sciences	Natural Sciences	Web Apps	NLP	Computer Vision
Regularize(40.0%)	Cross Val. (31.8%)	Regularize(41.2%)	Learn Rate(39.4%)	Learn Rate(46.2%)
Cross Val. (30.0%)	Learn Rate(22.7%)	Learn Rate(23.5%)	Batch Size(24.2%)	Batch Size(30.8%)
Learn Rate(10.0%)	DNN Arch.(18.2%)	Batch Size(11.8%)	DNN Arch.(18.2%)	DNN Arch.(11.5%)
Batch Size(10.0%)	Kernel (9.1%)	Cross Val. (11.8%)	Kernel (6.1%)	Regularize(11.5%)

- Learning Rate + Batch Size → looking for faster training

Post Processing

Social Sciences	Natural Sciences	Web Apps	NLP	Computer Vision
Prec/Rec (25.7%)	Accuracy (28.6%)	Accuracy (20.8%)	Prec/Rec(29.2%)	Visualiz. (33.3%)
Accuracy (20.0%)	Prec/Rec(18.6%)	Prec/Rec(20.8%)	Accuracy(27.1%)	Accuracy (29.8%)
Feat. Contrib. (17.1%)	Visualiz. (15.7%)	Case Studies (13.2%)	Case Studies (14.6%)	Prec/Rec(17.5%)
Visualiz. (14.3%)	Correlation (11.4%)	DCG (9.4%)	Human Eval (8.3%)	Case Studies (12.3%)

- Precision/Recall & Accuracy → coarse-grained evaluation
- Case Studies & Visualization → fine-grained evaluation

Takeaways

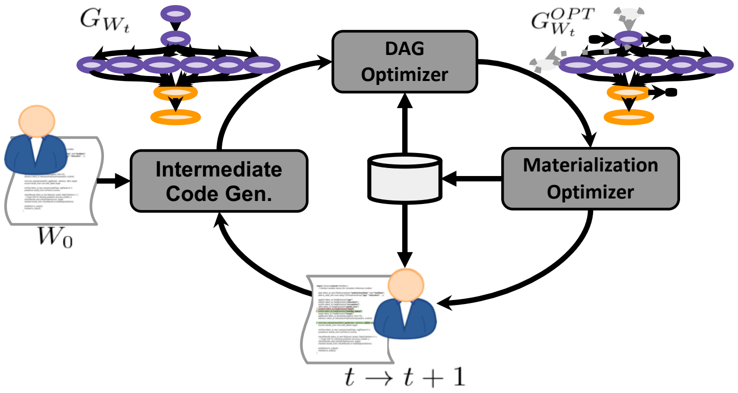
- Study iteration using **empirical evidence** from applied ML papers
    - Grouping by domains gives better insights
- Lessons from results
    - **Data prep**: fine-grained feature engineering, efficient joins
    - **ML**: explainable models and fast training
    - **Eval**: fine-grained evals are as common as coarse-grained metrics
- Open source dataset at https://github.com/helix-ml/AppliedMLSurvey

Future Work

- Refine statistics and estimators
- Develop insights and trends into a benchmark
- Look at code repositories (e.g. Kaggle) for a more complete picture

Helix

Accelerate Iterative Execution via Intermediates Reuse



- ❑ Address user needs discovered in our survey
- ❑ Selectively materialize intermediate results for reuse in future iterations

https://helix-ml.github.io

More on Helix in the technical report @ ***http://data-people.cs.illinois.edu/helix-tr.pdf***


T. HANKS