NYU

Introduction to Data Science

Center for Data Science
Iddo Drori, Spring 2019

Agenda

- Term project
- Automatic machine learning
- Automatic data science

Hyperparameter Optimization

Hyperparameter Optimization

Source: Auto-sklearn

Neural networks:

- Learning rate alpha
- Momentum term beta
- Minibatch size
- # of layers
- # of hidden units
- Learning rate decay

...

name	$\#\lambda$	cat (cond)	cont (cond)
AdaBoost (AB)	4	1 (-)	3 (-)
Bernoulli naïve Bayes	2	1 (-)	1 (-)
decision tree (DT)	4	1 (-)	3 (-)
extreml. rand. trees	5	2 (-)	3 (-)
Gaussian naïve Bayes	-	-	-
gradient boosting (GB)	6	-	6 (-)
kNN	3	2 (-)	1 (-)
LDA	4	1 (-)	3 (1)
linear SVM	4	2 (-)	2 (-)
kernel SVM	7	2 (-)	5 (2)
multinomial naïve Bayes	2	1 (-)	1 (-)
passive aggressive	3	1 (-)	2 (-)
QDA	2	-	2 (-)
random forest (RF)	5	2 (-)	3 (-)
Linear Class. (SGD)	10	4 (-)	6 (3)

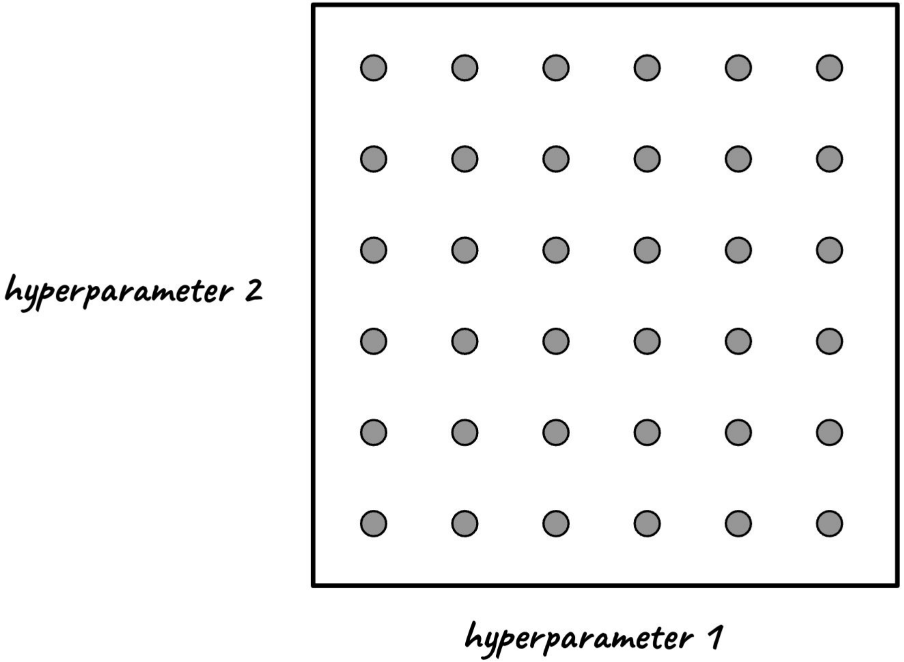
Hyperparameter Optimization

- Given dataset D find hyperparameters which minimize the loss of a model generated by algorithm A trained on D_{train} and evaluated on D_{valid}

$$\theta^* = arg\min_{\theta} \mathbb{E}_{(D_{train}, D_{valid}) \sim D} V(\mathcal{L}, A_\theta, D_{train}, D_{valid})$$

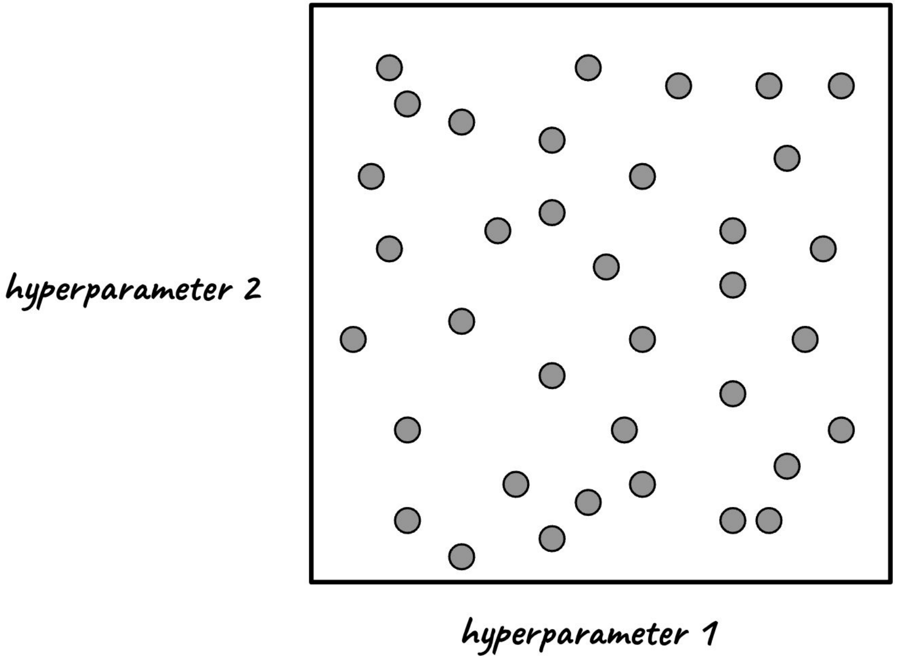
Grid Search

- Regularly sample grid
- Test grid values

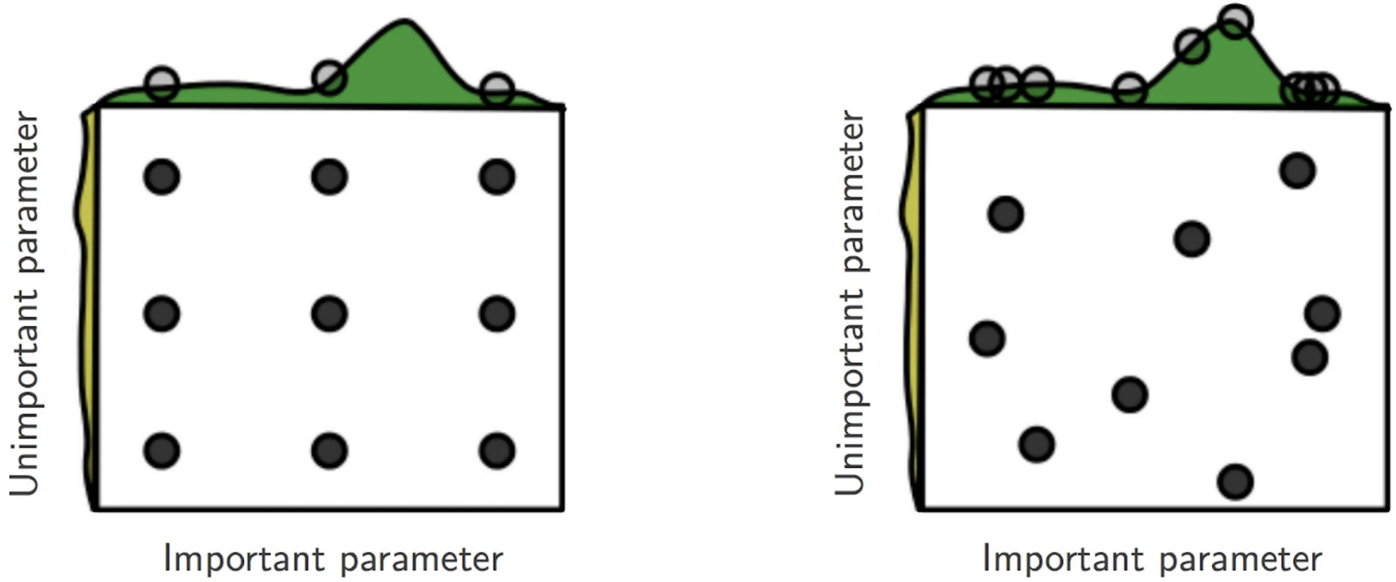


Random Search

- Randomly sample grid
- Test random values



Grid Search vs. Random Search



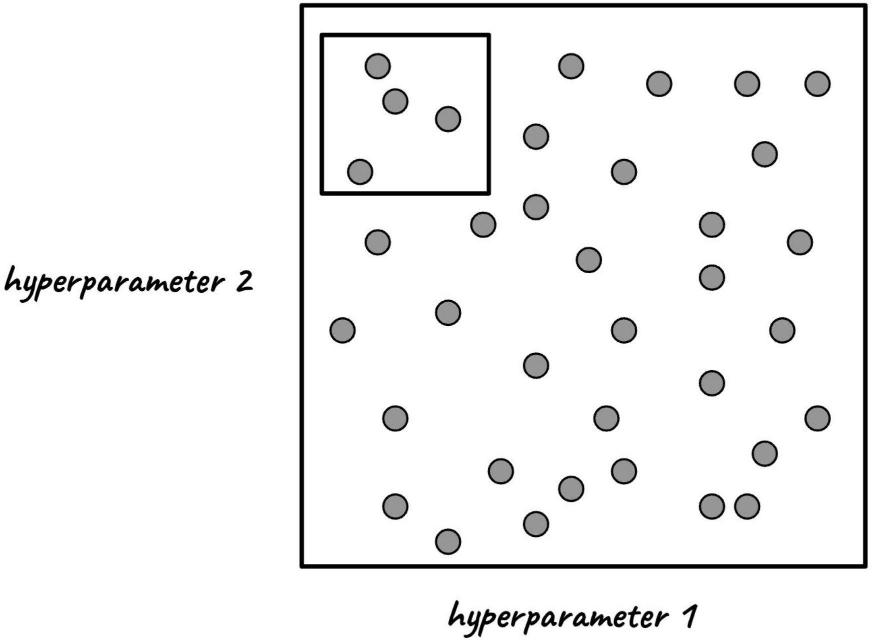
Source: Random search for hyper-parameter optimization, James Bergstra and Yoshua Bengio, JMLR 2012

Coarse to Fine Optimization

- Efficient optimization
- Sample examples
- Sample features

Adaptive Coarse to Fine Sampling

- Zoom in
- Perform dense search in small region of relevant values



Guided Search

- Suitable for n boolean params good/bad without interactions

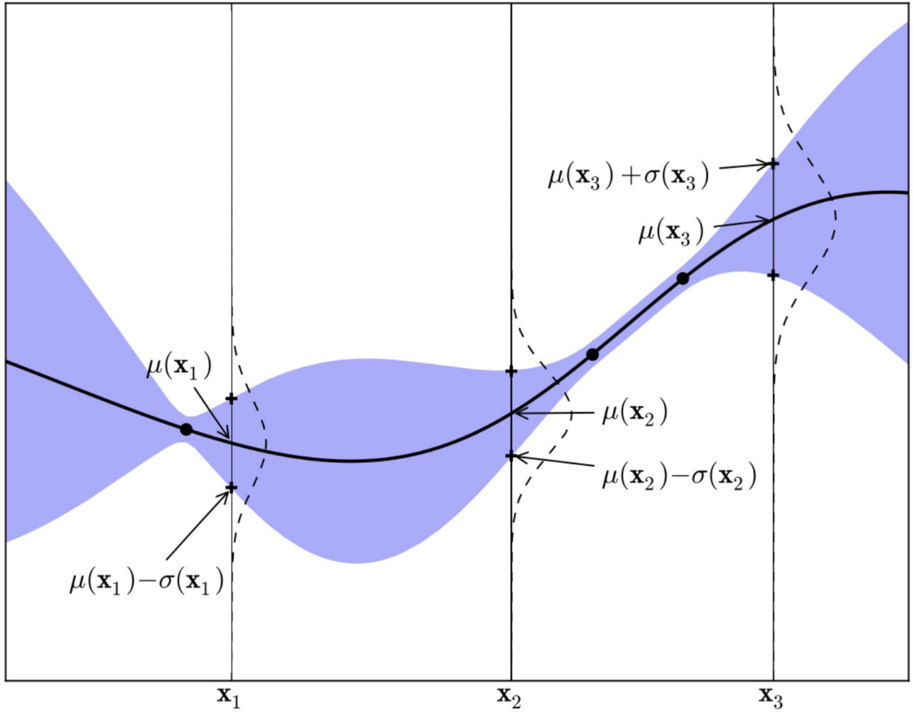
Training Strategies

- Supervised training of a single model: human in the loop
- Unsupervised training of multiple models in parallel: automatic

Covariance Matrix Adaptation

- Sample new configurations
- Reorder configurations based on fitness
- Update state variables, covariance, based on ordered solutions

Gaussian Processes



Source: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, Eric Brochu, Vlad M. Cora, Nando de Freitas, 2010.

Bayesian Optimization

- Build probabilistic model of objective
- Compute posterior distribution: Gaussian processes
- Optimize cheap surrogate function rather than expensive objective

Bayesian Optimization

Algorithm 1 Bayesian optimization

1: **for** $n = 1, 2, \ldots$ **do**

2: select new $\mathbf{x}_{n+1}$ by optimizing acquisition function α

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

3: query objective function to obtain y_{n+1}

4: augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$

5: update statistical model

6: **end for**

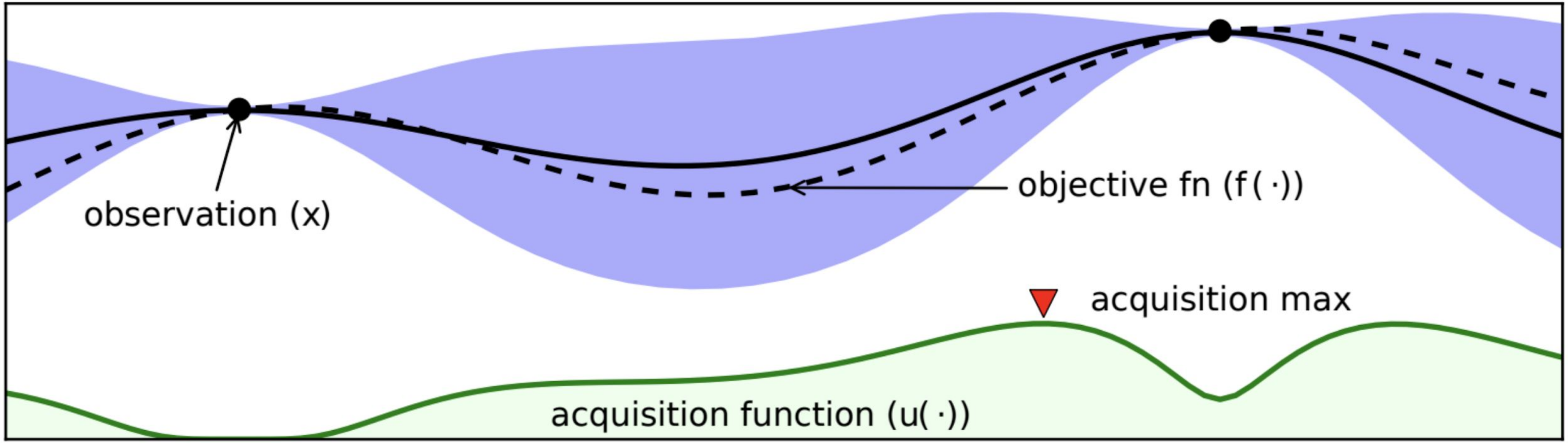
Source: Taking the human out of the loop: A review of Bayesian optimization, Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas, IEEE, 2016.

Bayesian Optimization

- Mean and confidence intervals estimated with a probabilistic model of objective function

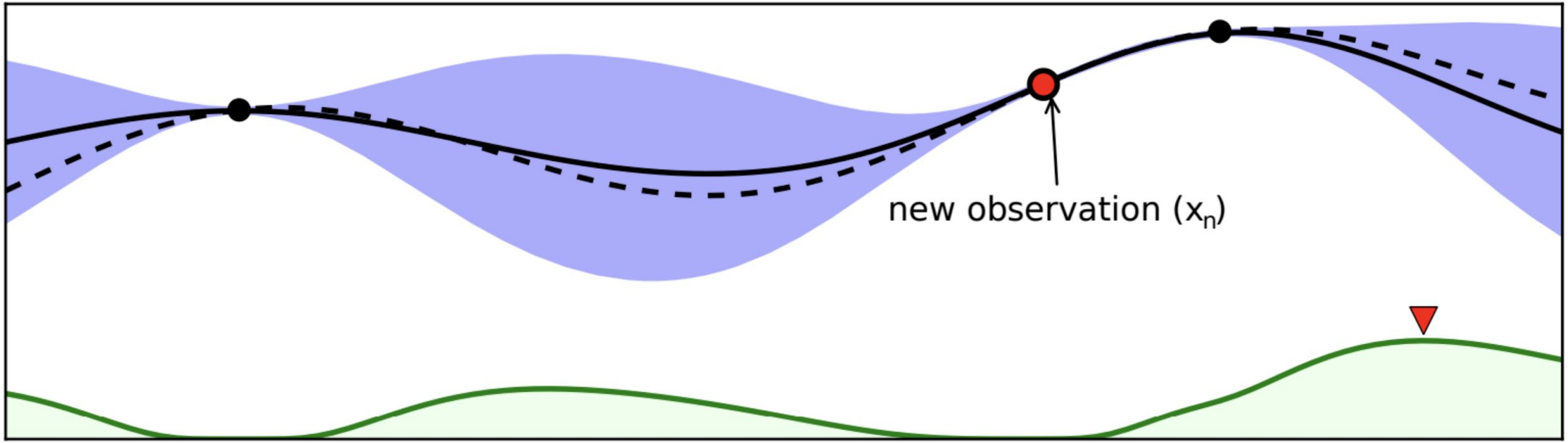
- High acquisition where model predicts both:
  High objective (exploitation)
  Prediction uncertainty is high (exploration)

Bayesian Optimization



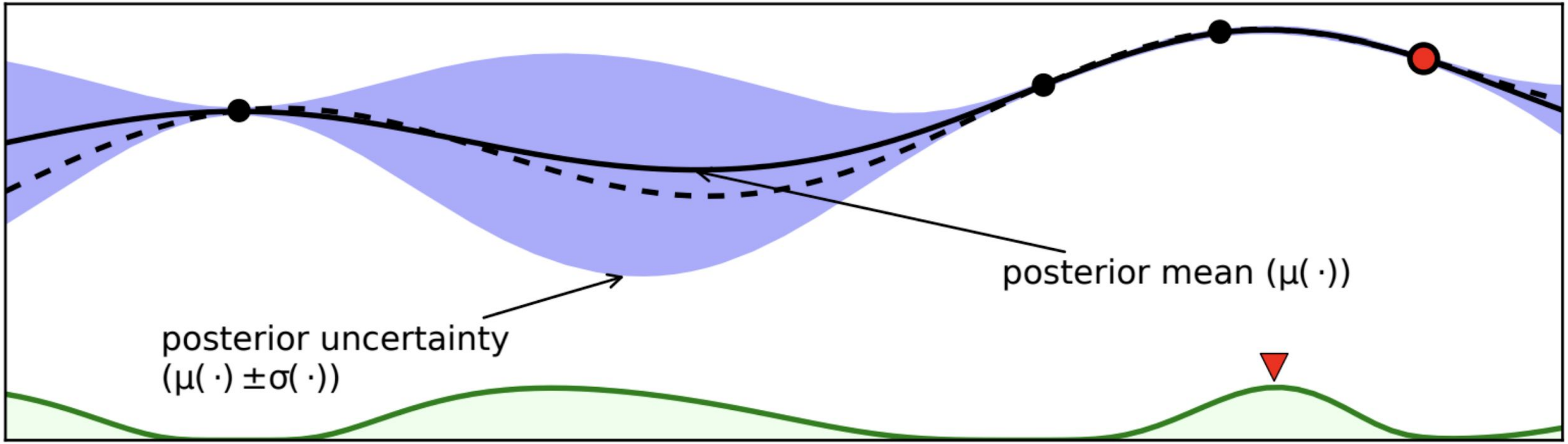
Source: Taking the human out of the loop: A review of Bayesian optimization, Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas, IEEE, 2016.

Bayesian Optimization



Source: Taking the human out of the loop: A review of Bayesian optimization, Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas, IEEE, 2016.

Bayesian Optimization



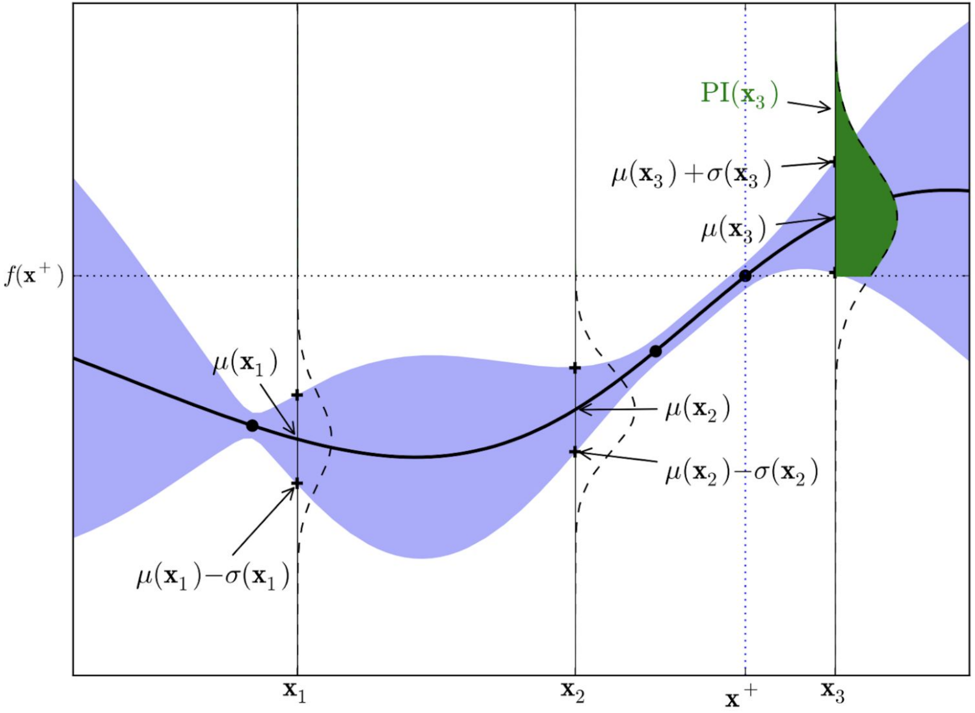
Source: Taking the human out of the loop: A review of Bayesian optimization, Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas, IEEE, 2016.

Bayesian Optimization

Acquisition functions:

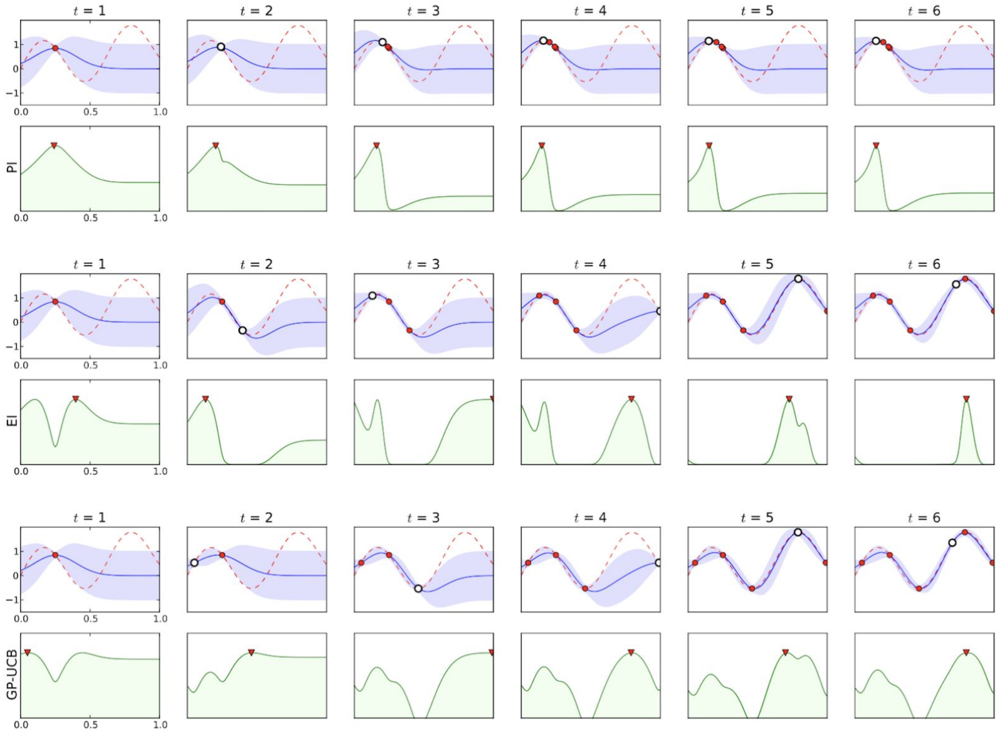
- Maximum probability of improvement: exploitation
- Expected improvement
- Entropy search
- Upper confidence bound (UCB)

Bayesian Optimization



Source: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, Eric Brochu, Vlad M. Cora, Nando de Freitas, 2010.

Bayesian Optimization



Source: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, Eric Brochu, Vlad M. Cora, Nando de Freitas, 2010.

Scale

- Log scale
- 1 - value
- Beta distribution: replace range by parameters of beta distribution, optimize those

Multiple Objectives

- Performances
- Time
- Memory
- Constraints

Algorithm Selection and Hyperparameter Optimization

Algorithm Selection and Hyperparameter Optimization

- Replace user's selection of algorithm and hyperparameters
- Given dataset D find the algorithm and its hyperparameters which minimize the loss of a model generated by algorithm A trained on D_{train} and evaluated on D_{valid}

$$A^*{}_{\theta^*} = arg \min_{A,\theta} \sum \mathcal{L}(A_\theta, D_{train}, D_{valid})$$

Algorithm Selection and Hyperparameter Optimization

- Source: AutoWeka

algorithm	# of hyperparameters		
Base Learners			
BayesNet	2	NaiveBayes	2
DecisionStump*	0	NaiveBayesMultinomial	0
DecisionTable*	4	OneR	1
GaussianProcesses*	10	PART	4
IBk*	5	RandomForest	7
J48	9	RandomTree*	11
JRip	4	REPTree*	6
KStar*	3	SGD*	5
LinearRegression*	3	SimpleLinearRegression*	0
LMT	9	SimpleLogistic	5
Logistic	1	SMO	11
M5P	4	SMOreg*	13
M5Rules	4	VotedPerceptron	3
MultilayerPerceptron*	8	ZeroR*	0
Ensemble Methods			
Stacking	2	Vote	2
Meta-Methods			
LWL	5	Bagging	4
AdaBoostM1	6	RandomCommittee	2
AdditiveRegression	4		
AttributeSelectedClassifier	2	RandomSubSpace	3
Feature Selection Methods			
BestFirst	2	GreedyStepwise	4

Automatic Data Science
Meta Learning

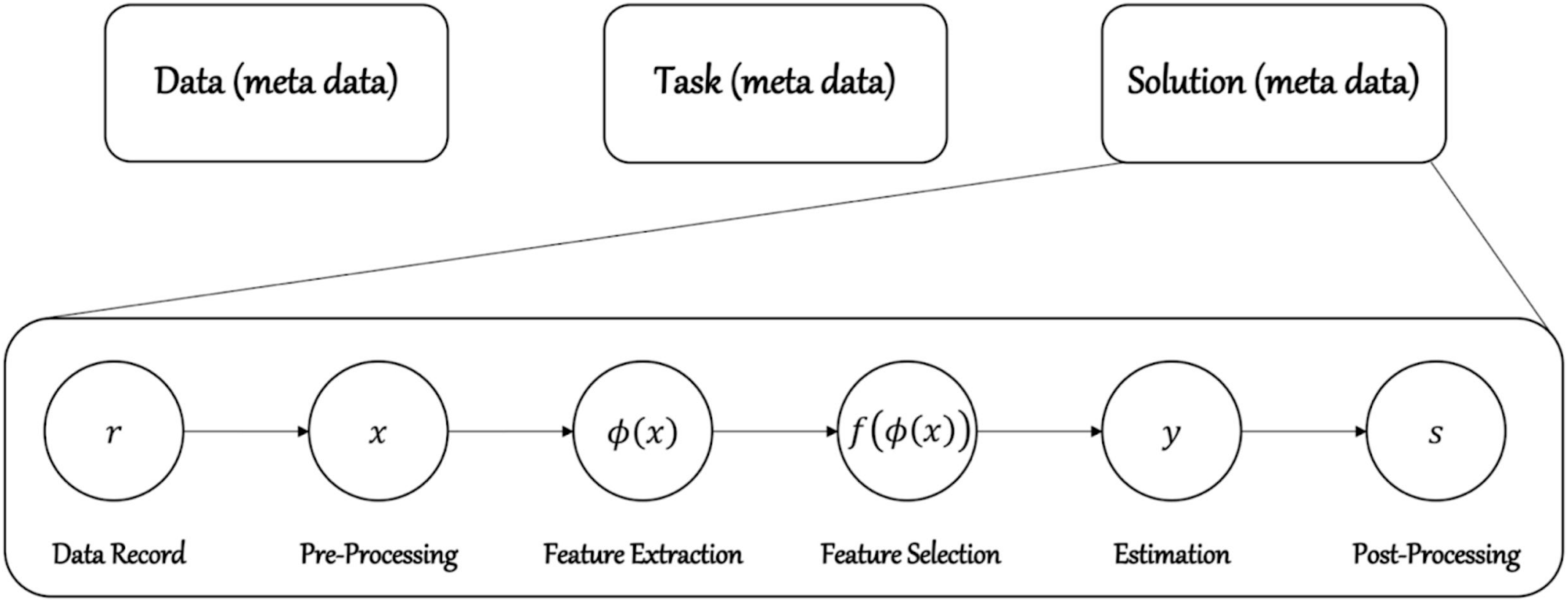
Meta Learning

- Learning to learn across tasks
- Experience vs. starting from scratch
- Given datasets with tasks, find machine learning pipelines optimizing performance and time.

Transfer Learning

- Learn model on a large dataset
- Modify part of model based on new dataset

Machine Learning Pipelines



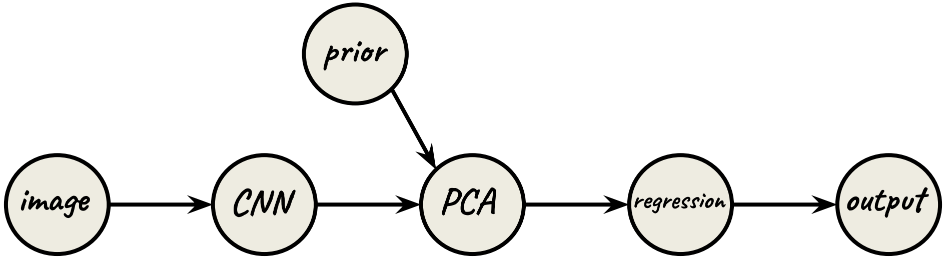
Source: Drori et al, 2018

Meta Data

- Data
- Task
- Solution

Gradient Based Methods

- Differentiable primitives
- Form a directed acyclic graph (DAG)
- Differentiable programming: optimize end-to-end
  End-to-end training of differentiable pipelines across machine learning frameworks, Mitar et al, 2017.



Meta Learning

- Source: Auto-sklearn

estimation

name	#λ	cat (cond)	cont (cond)
AdaBoost (AB)	4	1 (-)	3 (-)
Bernoulli naïve Bayes	2	1 (-)	1 (-)
decision tree (DT)	4	1 (-)	3 (-)
extreml. rand. trees	5	2 (-)	3 (-)
Gaussian naïve Bayes	-	-	-
gradient boosting (GB)	6	-	6 (-)
kNN	3	2 (-)	1 (-)
LDA	4	1 (-)	3 (1)
linear SVM	4	2 (-)	2 (-)
kernel SVM	7	2 (-)	5 (2)
multinomial naïve Bayes	2	1 (-)	1 (-)
passive aggressive	3	1 (-)	2 (-)
QDA	2	-	2 (-)
random forest (RF)	5	2 (-)	3 (-)
Linear Class. (SGD)	10	4 (-)	6 (3)

feature processing

name	#λ	cat (cond)	cont (cond)
extreml. rand. trees prepr.	5	2 (-)	3 (-)
fast ICA	4	3 (-)	1 (1)
feature agglomeration	4	3 ()	1 (-)
kernel PCA	5	1 (-)	4 (3)
rand. kitchen sinks	2	-	2 (-)
linear SVM prepr.	3	1 (-)	2 (-)
no preprocessing	-	-	-
nystroem sampler	5	1 (-)	4 (3)
PCA	2	1 (-)	1 (-)
polynomial	3	2 (-)	1 (-)
random trees embed.	4	-	4 (-)
select percentile	2	1 (-)	1 (-)
select rates	3	2 (-)	1 (-)
one-hot encoding	2	1 (-)	1 (1)
imputation	1	1 (-)	-
balancing	1	1 (-)	-
rescaling	1	1 (-)	-

Neural Architecture Search

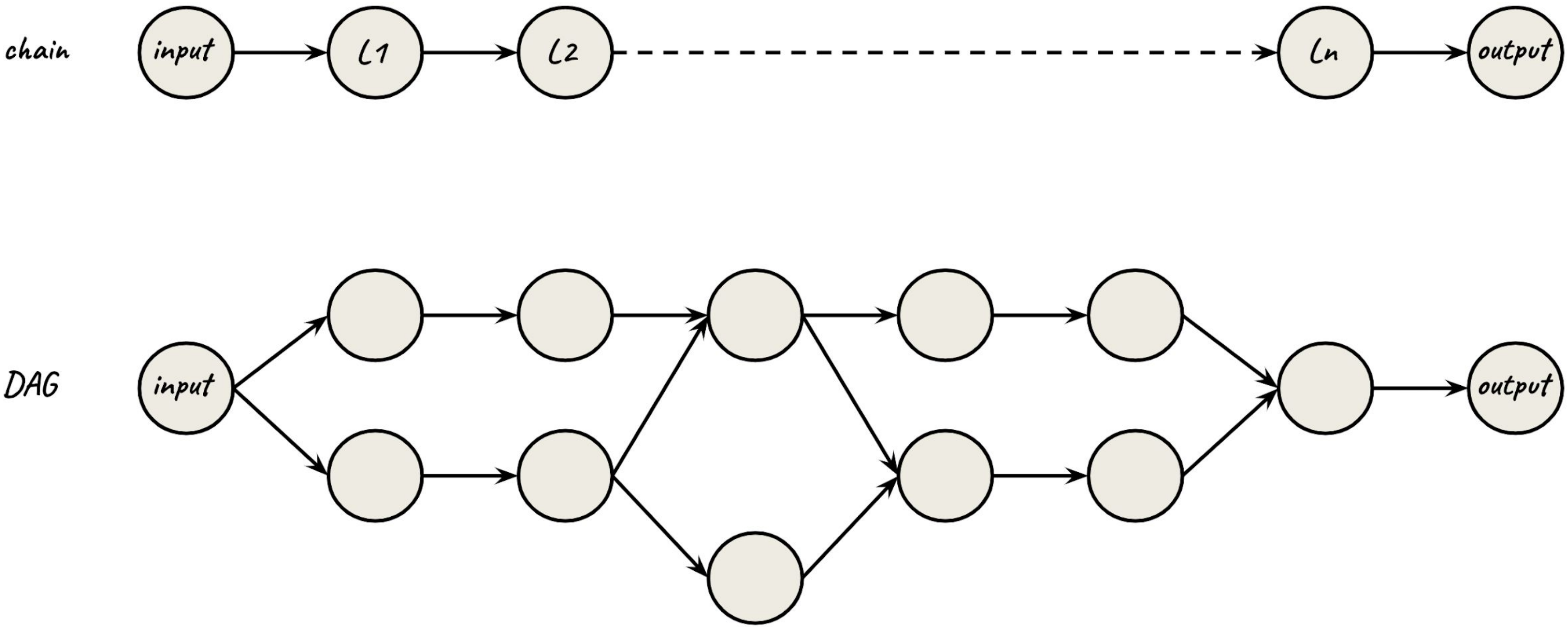
Neural Architecture Search

- Developing novel neural architectures manually is time consuming, error prone

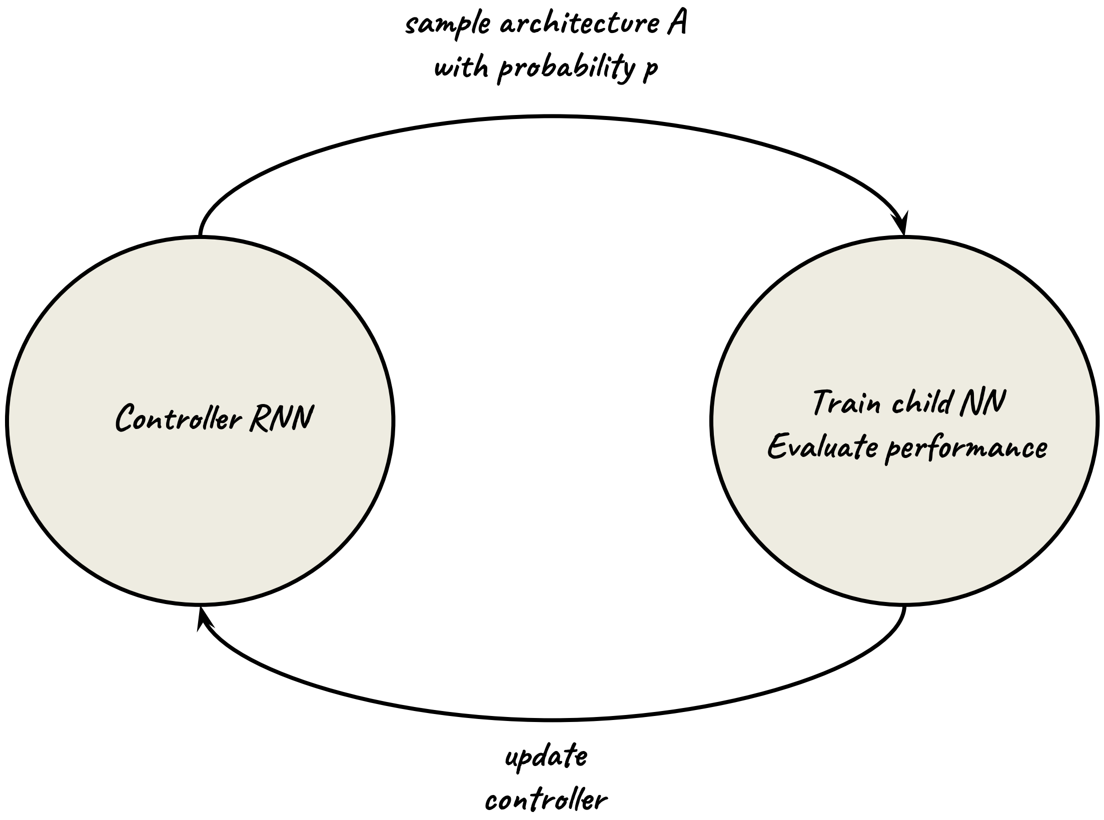
- Automatic methods for searching for neural network architectures

- Search space of architectures: chain, directed acyclic graph (DAG)
- Search strategy: exploration and exploitation trade-off
- Efficient performance estimation

Neural Architecture Search



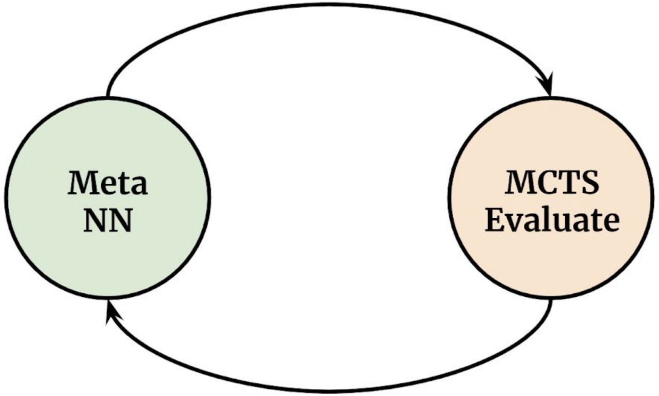
Neural Architecture Search



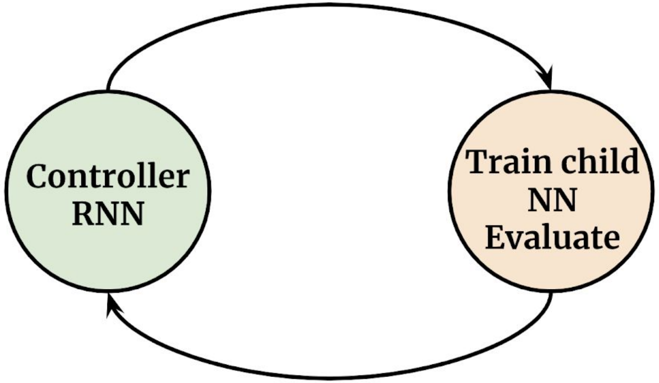
DARPA Data Driven Discovery of Models (D3M)

- Goal: solve any well defined ML task on any dataset specified by a user.
- Broad set of computational primitives as building blocks.
- Automatic systems for machine learning, synthesize pipeline and hyperparameters to solve a previously unknown data and problem.
- Human in the loop: user interface that enables users to interact with and improve the automatically generated results.
- Pipelines: pre-processing, feature extraction, feature selection, estimation, post-processing, evaluation

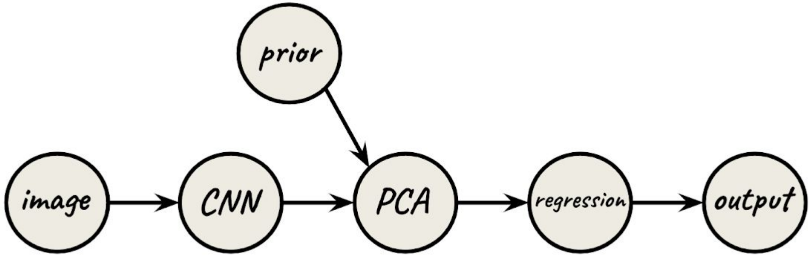
Automatic Machine Learning



Machine learning pipeline synthesis, Drori et al, 2018



Neural Architecture Search, Zoph and Le, 2016



End-to-end differentiable programming, Milutinovic et al, 2017

Automatic Machine Learning Systems

- Bayesian optimization, hyperparameter tuning:
  Autosklearn (Feurer et al, NIPS 2015)
  AutoWEKA (Kotthoff et al, JMLR 2017)
- Tree search of algorithms and hyperparameters, multi-armed bandit
  Auto-Tuned Models (Swearingen et al, Big Data 2017)
- Deep reinforcement learning: expert iteration
  AlphaD3M (Drori et al, AutoML 2018)
- Evolutionary algorithms
  TPOT (Olson et al, ICML 2016) machine learning pipelines as trees
  Autostacker (Chen et al, GECCO 2018) ML pipelines as stacked layers.
- Collaborative filtering
  OBOE (Yang et al, 2018).
- Neural architecture search
  AutoKeras (Jin et al, 2018).

NYU

Thank you