Automated Machine Learning: AutoML

Hugo Jair Escalante, Eduardo Morales

hugojair@inaoep.mx

Overview

- The goal of this lecture is to give the reader an overview of AutoML, from the fundamentals to main approaches and challenges

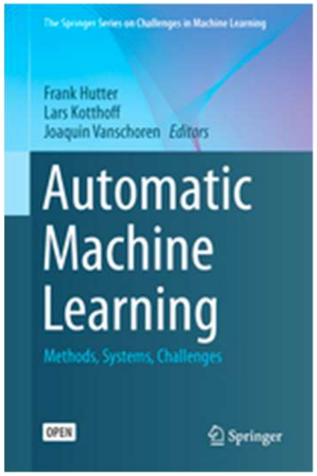
Contents

- Introduction to AutoML
- Definitions
- Brief history
- Early years
- Second wave
- Recent progress on AutoML

AutoML

- Automatic Machine Learning*
  - Research area that targets progressive automation of machine learning
  - Field of research focusing on the development of autonomous methods for solving a variety of machine learning problems

- Motivation.
  - Large amounts of data readily available everywhere
  - Lack of domain and/or ML experts who can advise/supervise the development of ML-based systems
  - Need to tune ML models



https://www.springer.com/us/book/9783030053178

* We will focus on supervised learning

Machine learning

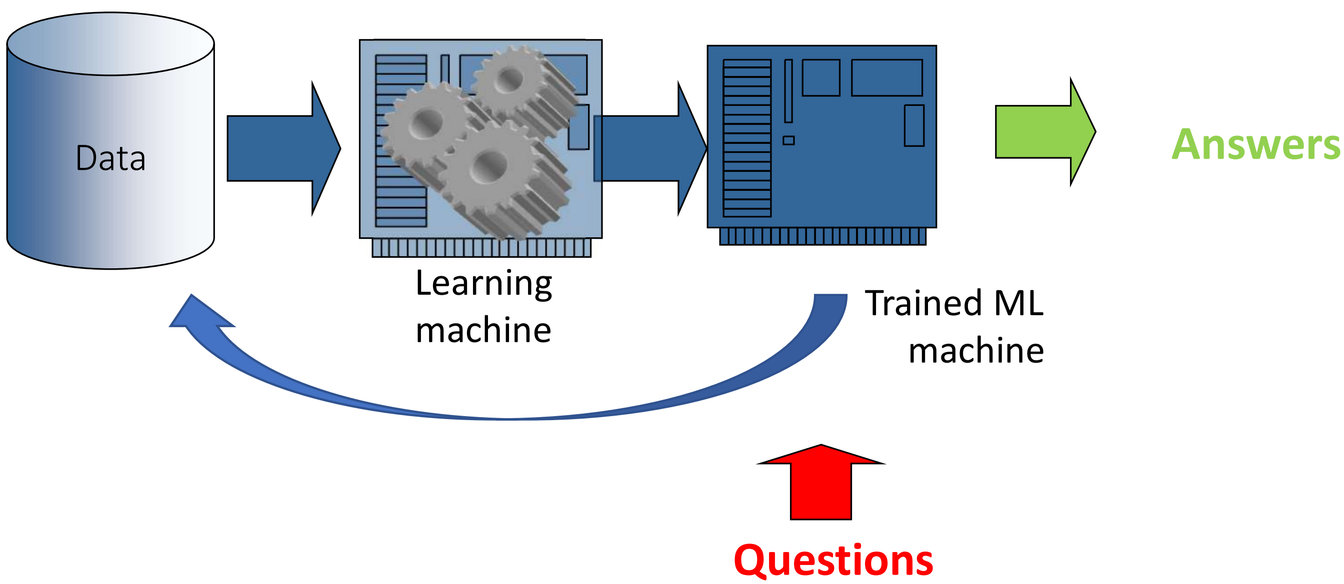
- Subfield of AI focusing on the development of computer programs that:
  - Adapt their behavior automatically from data
  - Are capable of learning without being explicit programmed

How can we build computer programs that learn from experience?

What are the fundamental laws governing learning processes?

T. M. Mitchell. **The Discipline of Machine Learning.** Tech. Report. CMU-ML-06-108, 2006

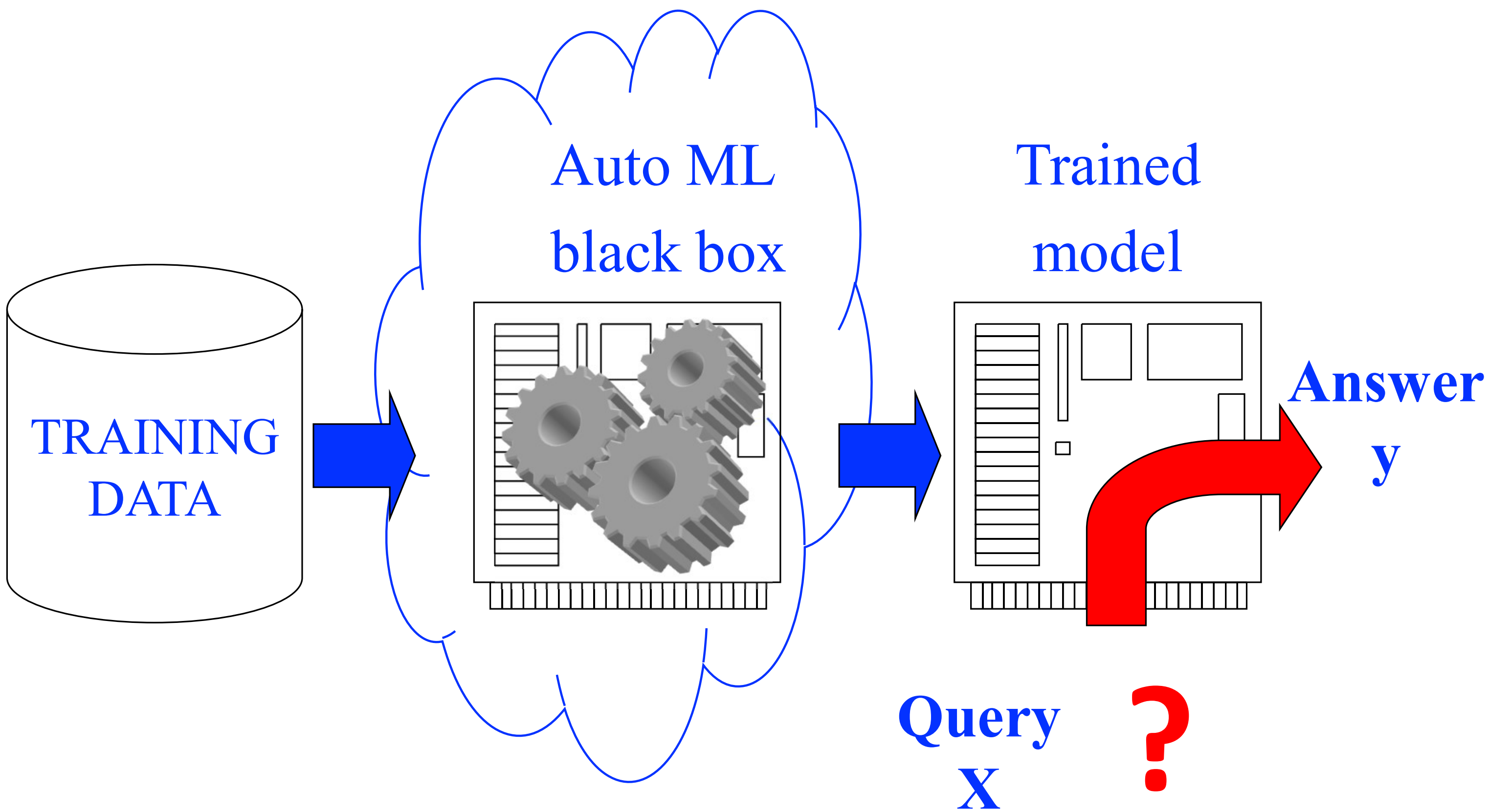
ML = Representation + Evaluation + Optimization



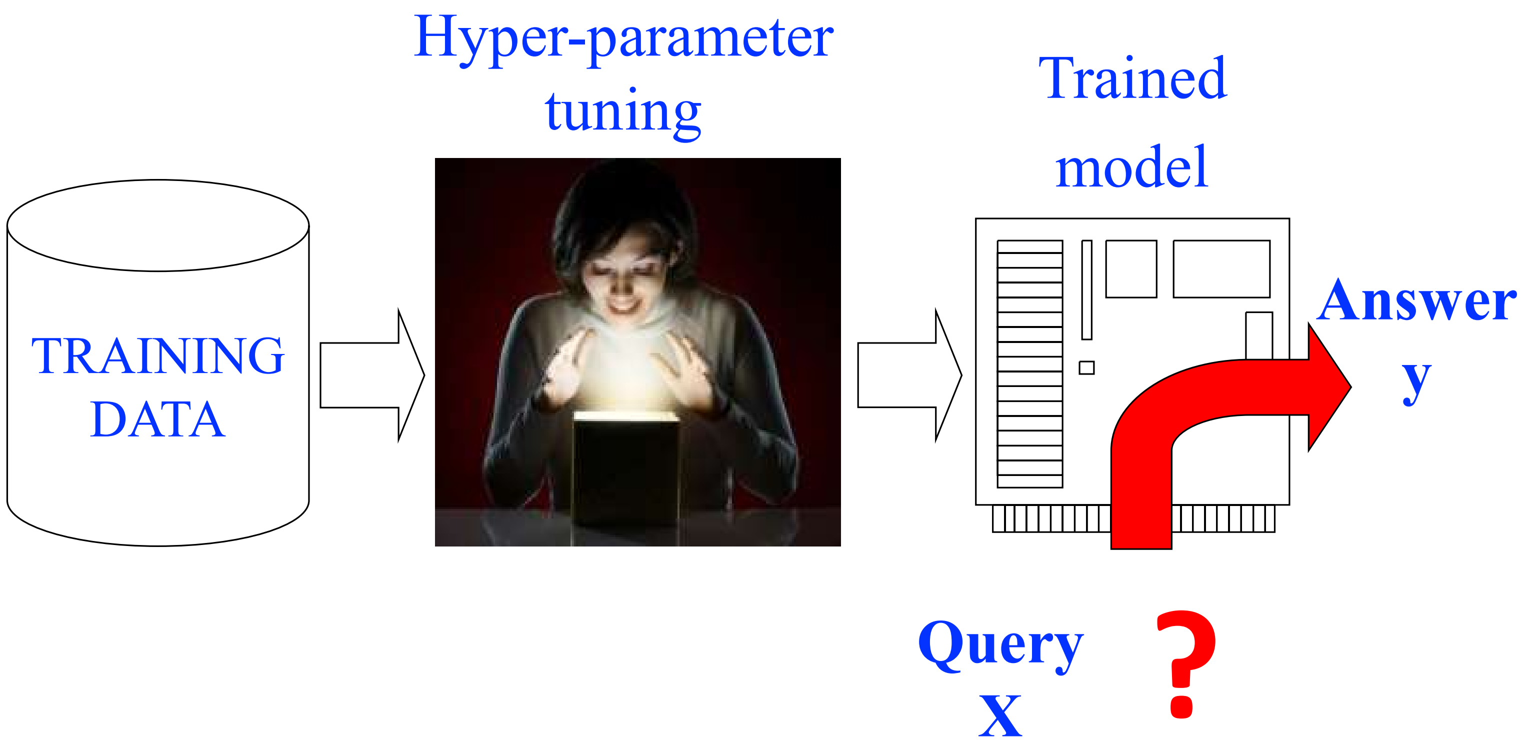
Pedro Domingos. **A Few Useful Things to Know about Machine Learning**. Communications of the ACM, 55(10):78--87, 2012

Isabelle Guyon. **A Practical Guide to Model Selection.** In Jeremie Marie, editor, Machine Learning Summer School 2008, Springer Texts in Statistics, 2011. (slide from I.Guyon's)

The AutoML dream



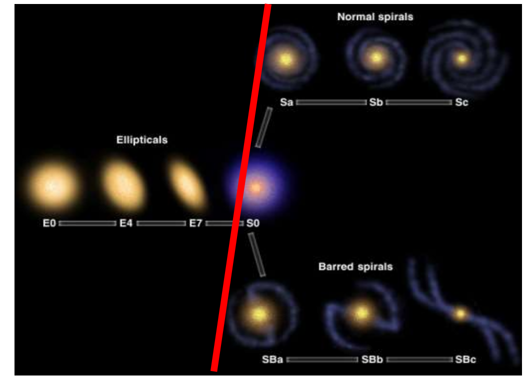
The REALITY

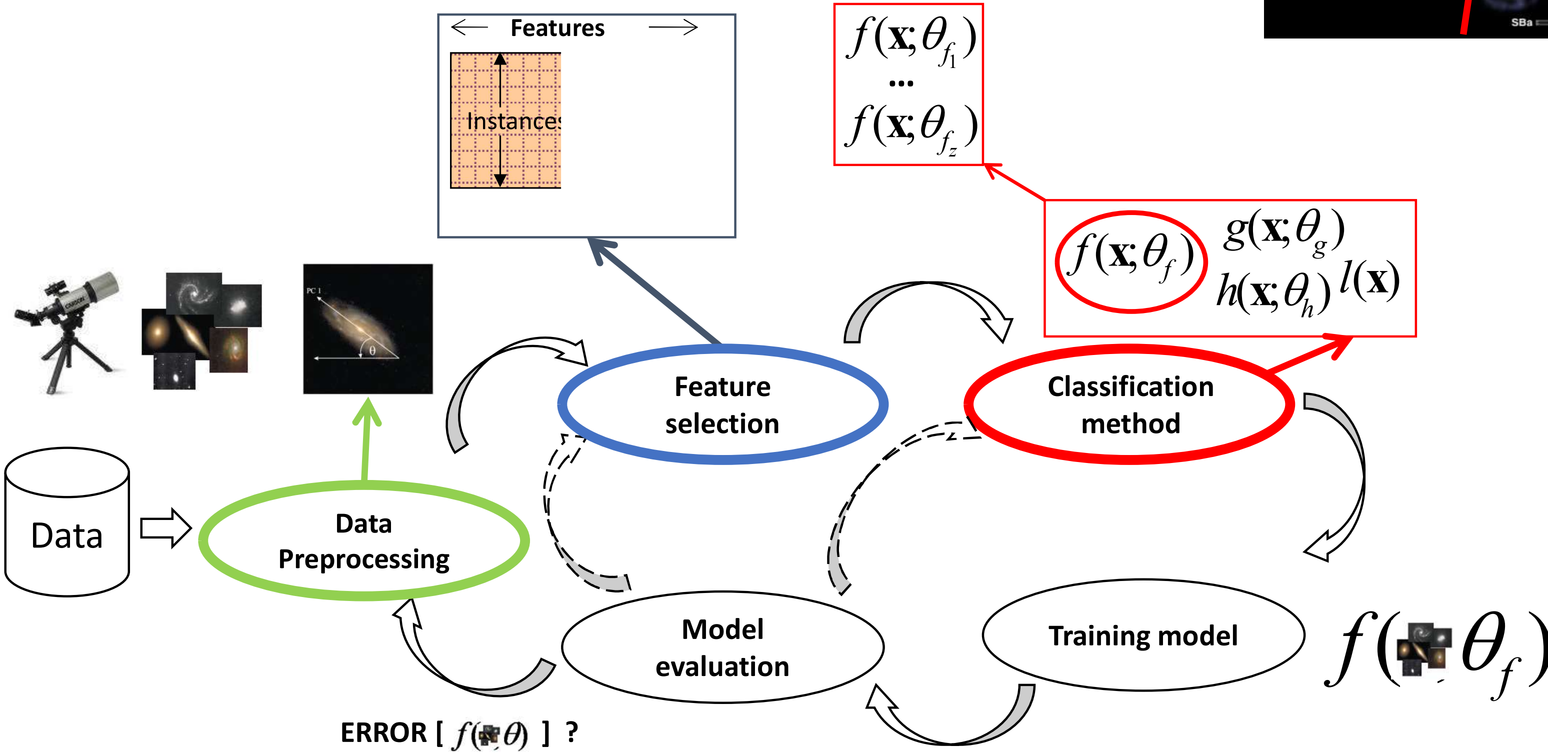

Hyper-parameter tuning
Trained model
TRAINING DATA
Answer y
Query X
?

Have you ever asked yourself…

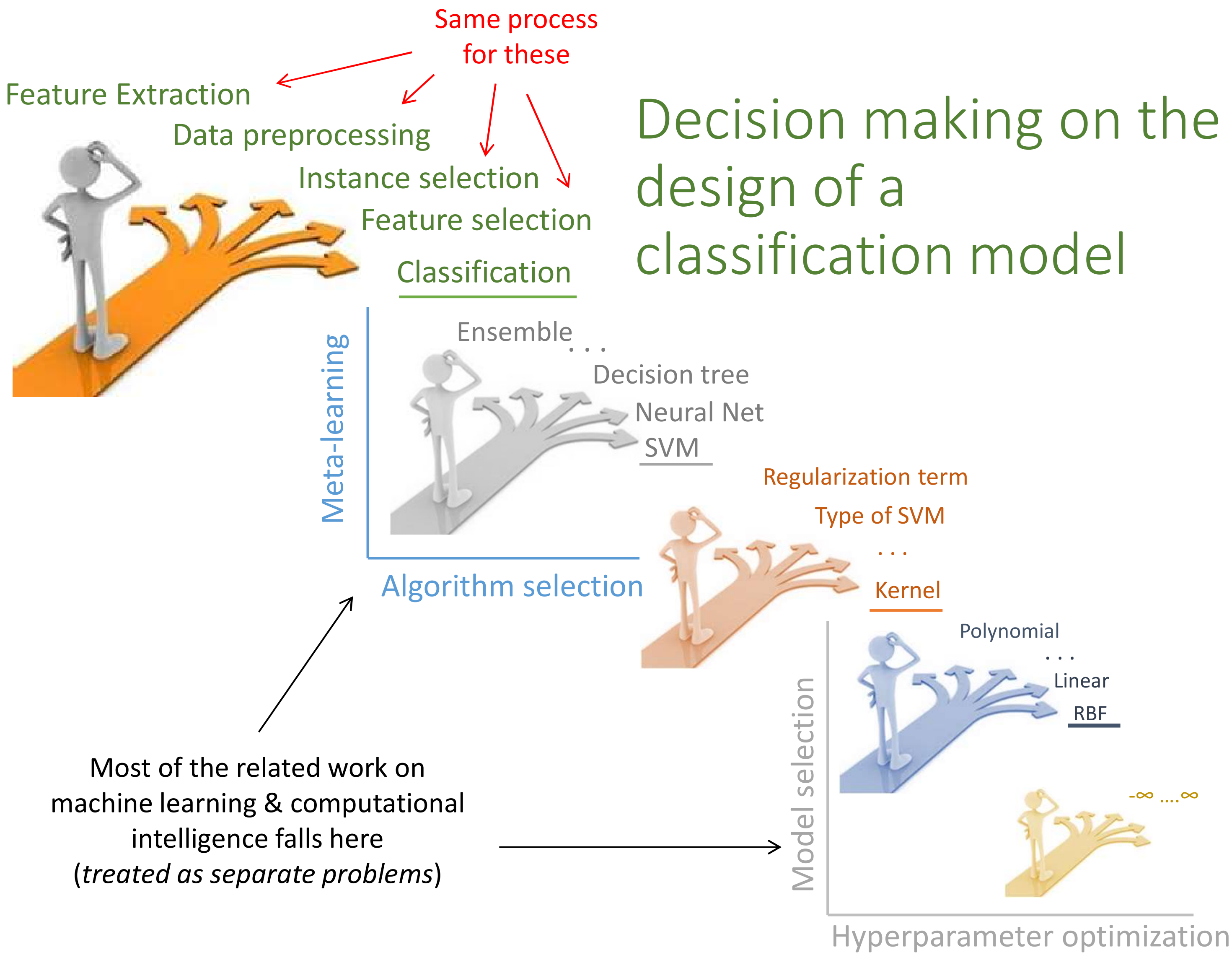
- What do you do when desiging a ML solution?
- How do you start designing/developong a ML model?
- How do you ensure the performance of your model is the best you can get?
- Do you have any biases on the way you design ML systems?

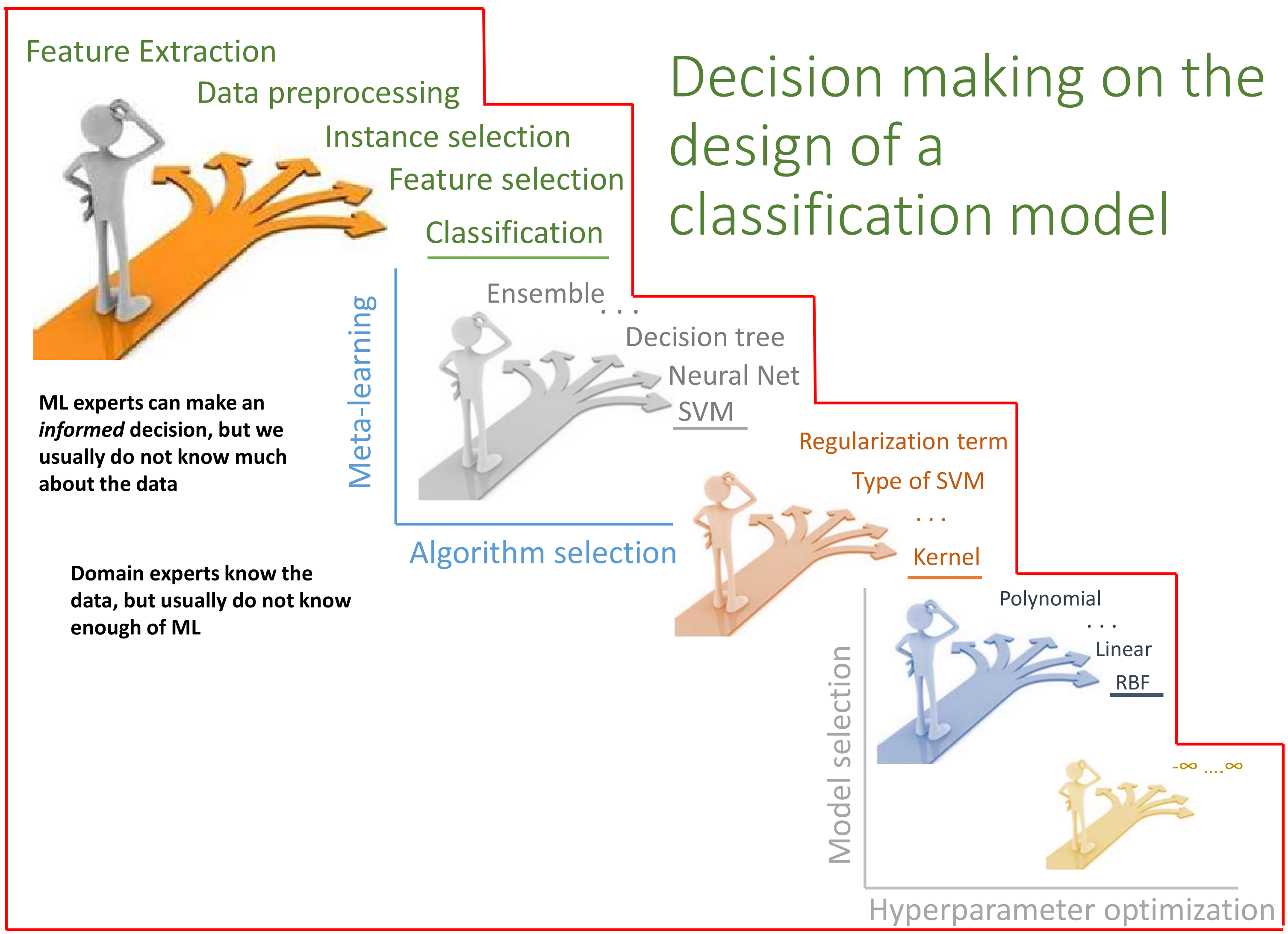
Typical PR design processs





Decision making on the design of a classification model




Decision making on the design of a classification model
Feature Extraction
Data preprocessing
Instance selection
Feature selection
Classification
ML experts can make an *informed* decision, but we usually do not know much about the data
Domain experts know the data, but usually do not know enough of ML
Meta-learning
Ensemble
. . .
Decision tree
Neural Net
SVM
Algorithm selection
Regularization term
Type of SVM
. . .
Kernel
Polynomial
. . .
Linear
RBF
-∞ ....∞
Model selection
Hyperparameter optimization

Some issues with the cycle of design

- Commonly, the above issues are fixed manually, relying on:
  - Domain expert’s knowledge
  - Machine learning specialists’ knowledge
  - Trial and error

- The design/development of a pattern classification system relies on the knowledge and biases of humans, which may be risky, expensive and time consuming

- Automated solutions are available but only for particular processes (e.g., either feature selection, or classifier selection but not both)

More issues …

- Models' performance is tied to an adequate tuning

- Training a single model and evaluating its performance is often computationally expensive

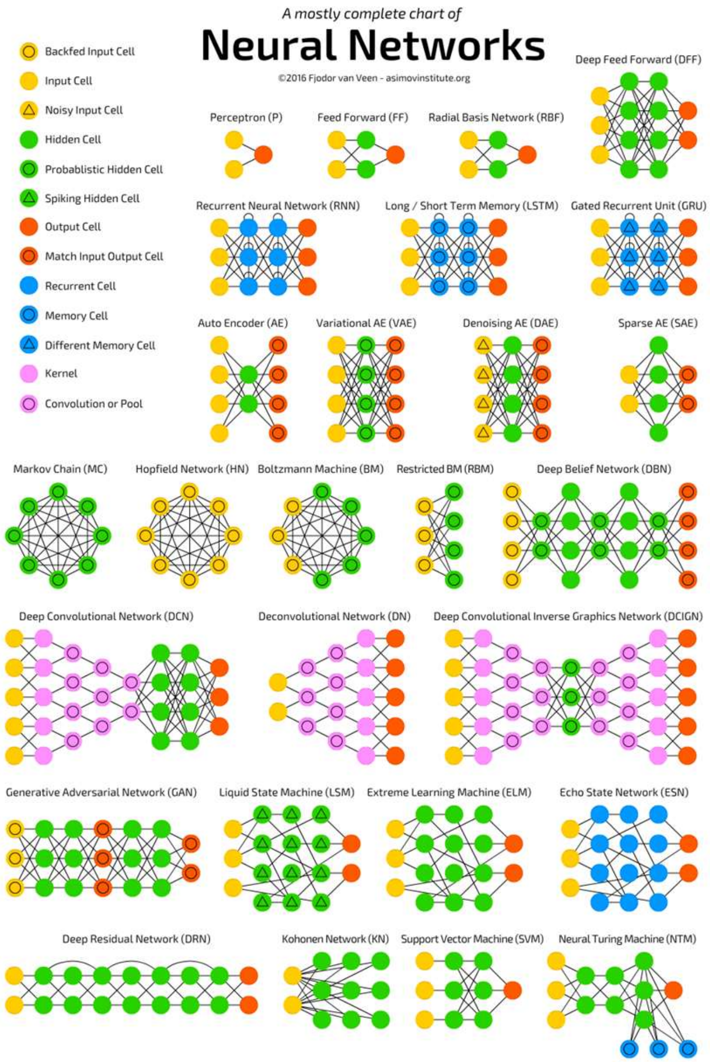
- The "search space" for the cycle of design is huge and highly complex

AutoML: Is it possible to automate the whole process?

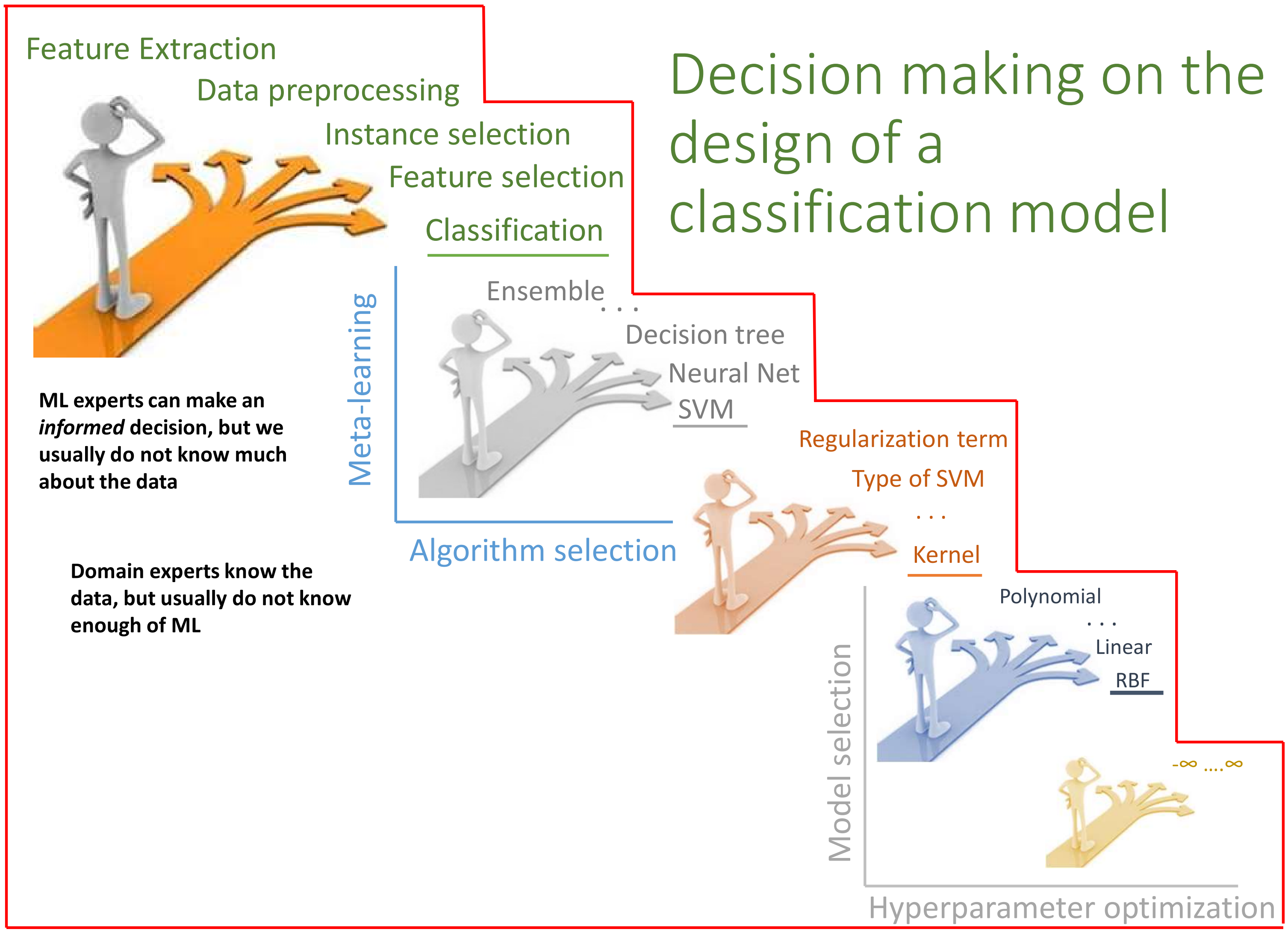
(What about deep learning?)

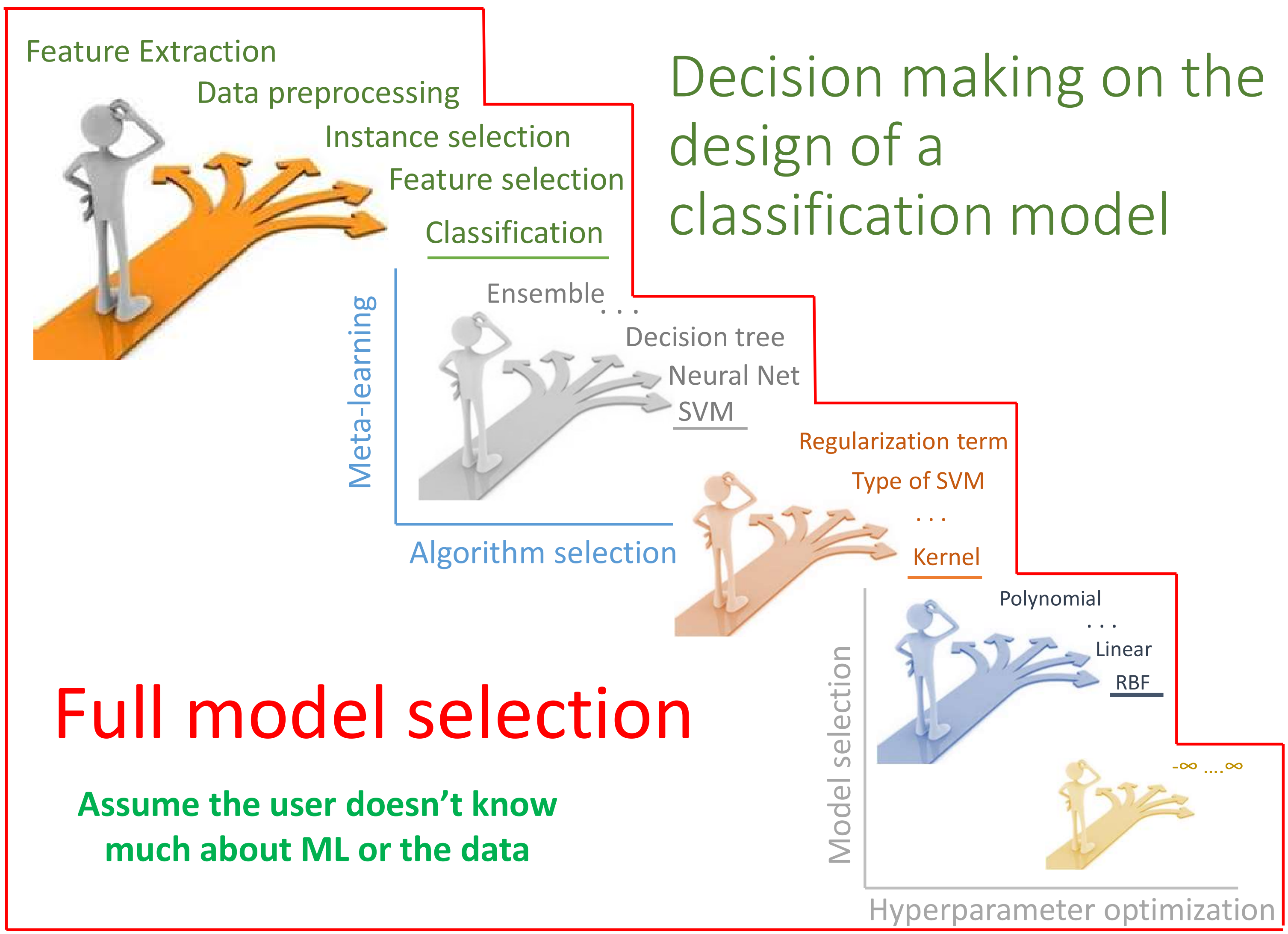
- The success of DL models largerly depends on the desing choices made by developers:
  - How many layers?
  - What type of units?
  - For CNNs, what kernel size? how many filters / feature maps per layer, pooling strategy? Etc.
  - Regularization strategy? Activation functions?
  - Optimizer, learning algorithm?
  - Etc.

Similar problem!



Decision making on the design of a classification model




Feature Extraction
Data preprocessing
Instance selection
Feature selection
Classification
Decision making on the design of a classification model
Meta-learning
Ensemble
. . .
Decision tree
Neural Net
SVM
Algorithm selection
Regularization term
Type of SVM
. . .
Kernel
Polynomial
. . .
Linear
RBF
-∞ ....∞
Model selection
Hyperparameter optimization
Full model selection
Assume the user doesn't know much about ML or the data

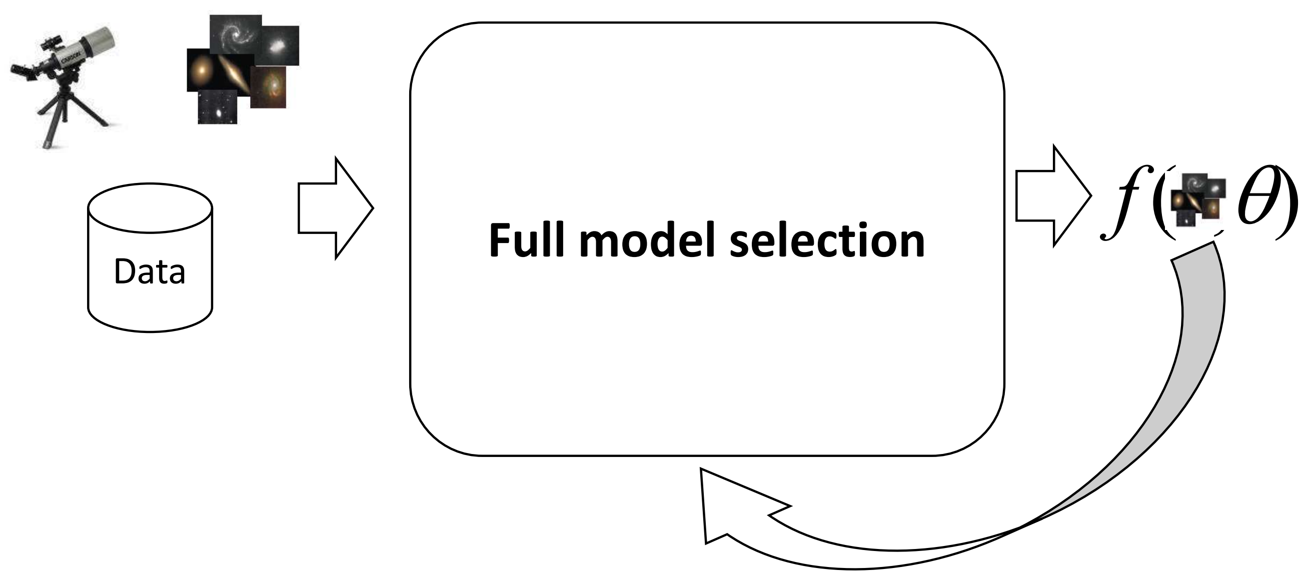
Full model selection

- Given a set of methods for data preprocessing, feature selection and classification select the combination of methods (together with their hyperparameters) that minimizes an estimate of classification performance

H. J. Escalante. **Towards a Particle Swarm Model Selection algorithm.** *Multi-level inference workshop and model selection game, NIPS, Whistler, VA, BC, Canada, 2006.*

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

Full model selection



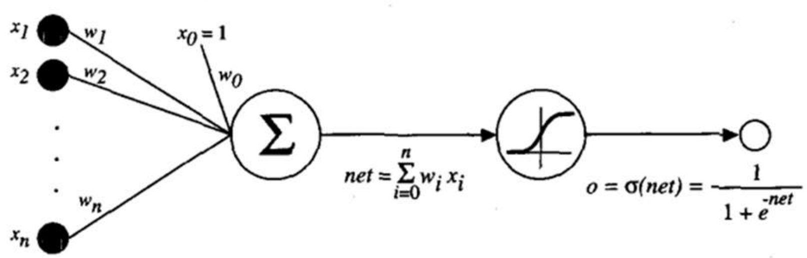
H. J. Escalante. **Towards a Particle Swarm Model Selection algorithm.** *Multi-level inference workshop and model selection game, NIPS, Whistler, VA, BC, Canada, 2006.*

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

(Parameters vs. hyperparameters)

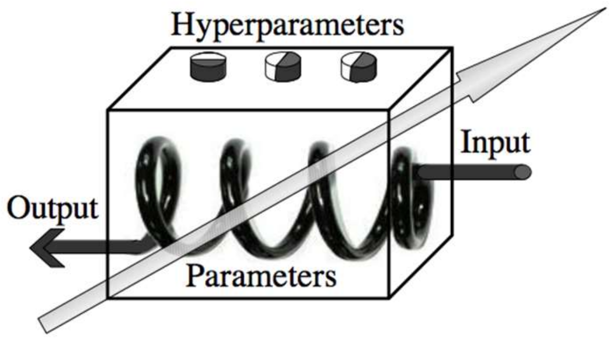
- Support vector machines
- Multilayer perceptron
- K-means
- K-NN
- Random forest
- CNN
- Linear regression

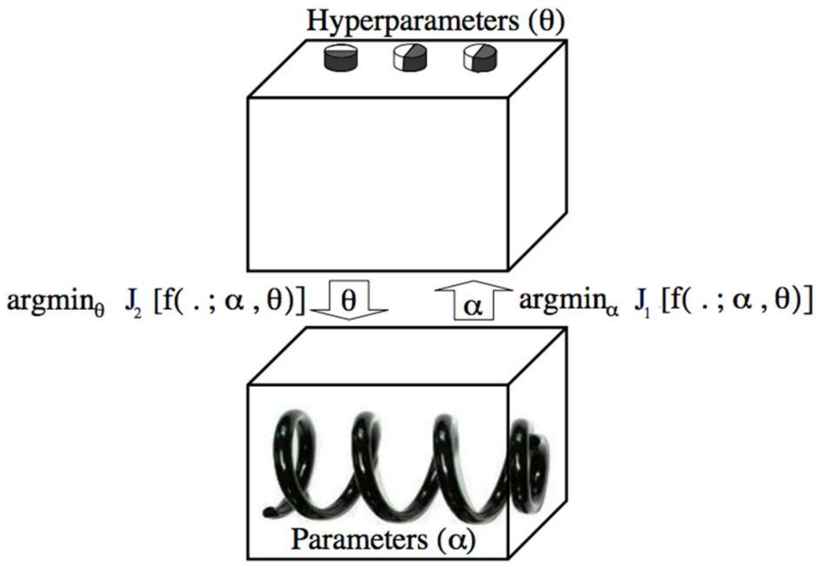
$$\text{mín}\ \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{N}\psi_i$$



Full model selection

- Two levels of inference:





FMS or AutoML?

- Thornton et al. formulation (2013): The *Combined Selection and Hyperparameter Optimization problem*

Definition 2 (CASH). *Let $\mathcal{A} = \{A^{(1)}, \ldots, A^{(R)}\}$ be a set of algorithms, and let the hyperparameters of each algorithm $A^{(j)}$ have domain $\Lambda^{(j)}$. Further, let $D_{train} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set which is split into K cross-validation folds $\{D_{valid}^{(1)}, \ldots, D_{valid}^{(K)}\}$ and $\{D_{train}^{(1)}, \ldots, D_{train}^{(K)}\}$ such that $D_{train}^{(i)} = D_{train} \backslash D_{valid}^{(i)}$ for $i = 1, \ldots, K$. Finally, let $\mathcal{L}(A_{\boldsymbol{\lambda}}^{(j)}, D_{train}^{(i)}, D_{valid}^{(i)})$ denote the loss that algorithm $A^{(j)}$ achieves on $D_{valid}^{(i)}$ when trained on $D_{train}^{(i)}$ with hyperparameters $\boldsymbol{\lambda}$. Then, the* Combined Algorithm Selection and Hyperparameter optimization (CASH) *problem is to find the joint algorithm and hyperparameter setting that minimizes this loss:*

$$A^{\star}, \boldsymbol{\lambda}_{\star} \in \operatorname*{argmin}_{A^{(j)} \in \mathcal{A}, \boldsymbol{\lambda} \in \Lambda^{(j)}} \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}(A_{\boldsymbol{\lambda}}^{(j)}, D_{train}^{(i)}, D_{valid}^{(i)}). \tag{1}$$

C. Thornton, F. Hutter, H. Hoos, and K. Leyton-Brown. **Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms.** In Proc. KDD'13, pages 847–855, 2013.

FMS or AutoML?

- … and a more recent one in 2015 by Feurer et al.:

Definition 1 (AutoML). *For* $i = 1, \ldots, n+m$, *let* $\boldsymbol{x}_i \in \mathbb{R}^d$ *denote a feature vector and* $y_i \in Y$ *the corresponding target value. Given a training dataset* $D_{train} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ *and the feature vectors* $\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+m}$ *of a test dataset* $D_{test} = \{(\boldsymbol{x}_{n+1}, y_{n+1}), \ldots, (\boldsymbol{x}_{n+m}, y_{n+m})\}$ *drawn from the same underlying data distribution, as well as a resource budget* b *and a loss metric* $\mathcal{L}(\cdot, \cdot)$, *the AutoML problem is to (automatically) produce test set predictions* $\hat{y}_{n+1}, \ldots, \hat{y}_{n+m}$. *The loss of a solution* $\hat{y}_{n+1}, \ldots, \hat{y}_{n+m}$ *to the AutoML problem is given by* $\frac{1}{m} \sum_{j=1}^{m} \mathcal{L}(\hat{y}_{n+j}, y_{n+j})$.

Matthias Feurer et al. **Efficient and Robust Automated Machine Learning**. NIPS 2015

Informal, but intuitive definition

- *AutoML is* the *task of finding the f that better generalizes in any possible dataset T with the less possible human intervention*
  - *f* can be the composition of multiple functions that may transform the input space, subsampling data, combining multiple predictors, etc.
  - where each of these models could be formed in turn by several other functions/models.

Hugo Jair Escalante. **Automated Machine Learning - a brief review at the end of the early years.** *Forthcoming, 2020*

A more inclusive notion of AutoML

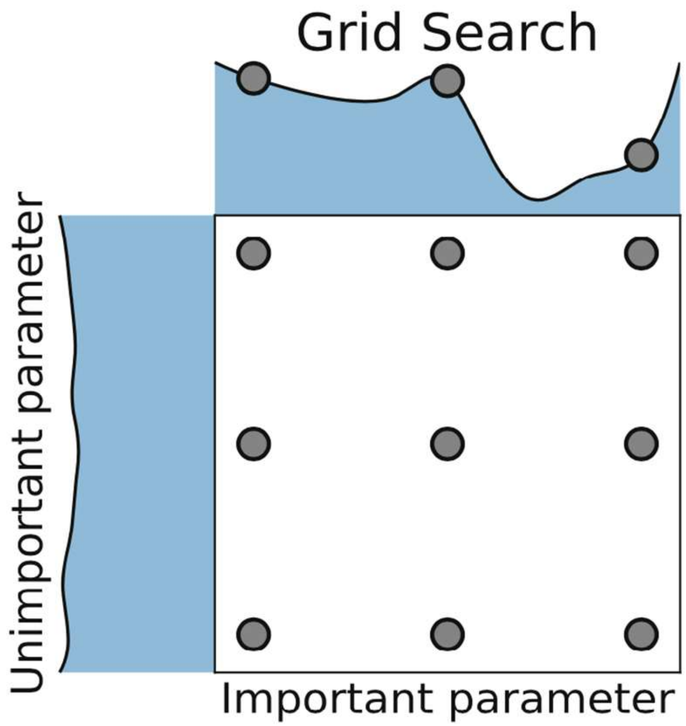
- Three level categorization

Level	Input	Output	Examples	Encoded by
α-level	sample/example (e.g. an image)	prediction of label (e.g. 'dog' or 'cat')	heuristically hard-coded classifier or already trained classifier	parameters, hyperparameters (if any) and meta-parameters (if any)
β-level	task/dataset (e.g. MNIST [20], CIFAR-10 [19])	α-level algorithm	learning algorithms (e.g. SVM [6], CNN [20]); HPO algorithms (e.g. grid search cross-validation, SMAC [1], NAS [34])	hyperparameters and meta-parameters (if any)
γ-level	meta-dataset (e.g. OpenML [29])	β-level algorithm	meta-learning algorithms (e.g. meta-learning part in Auto-sklearn [11])	meta-parameters

Liu Zhengying, Zhen Xu, Meysam Madadi, Julio Jacques Junior, Sergio Escalera, Shangeth Rajaa, and Isabelle Guyon. **Overview and unifying conceptualization of automated machine learning.** In Proc. of Automating Data Science Workshop @ECML-PKDD, 2019.

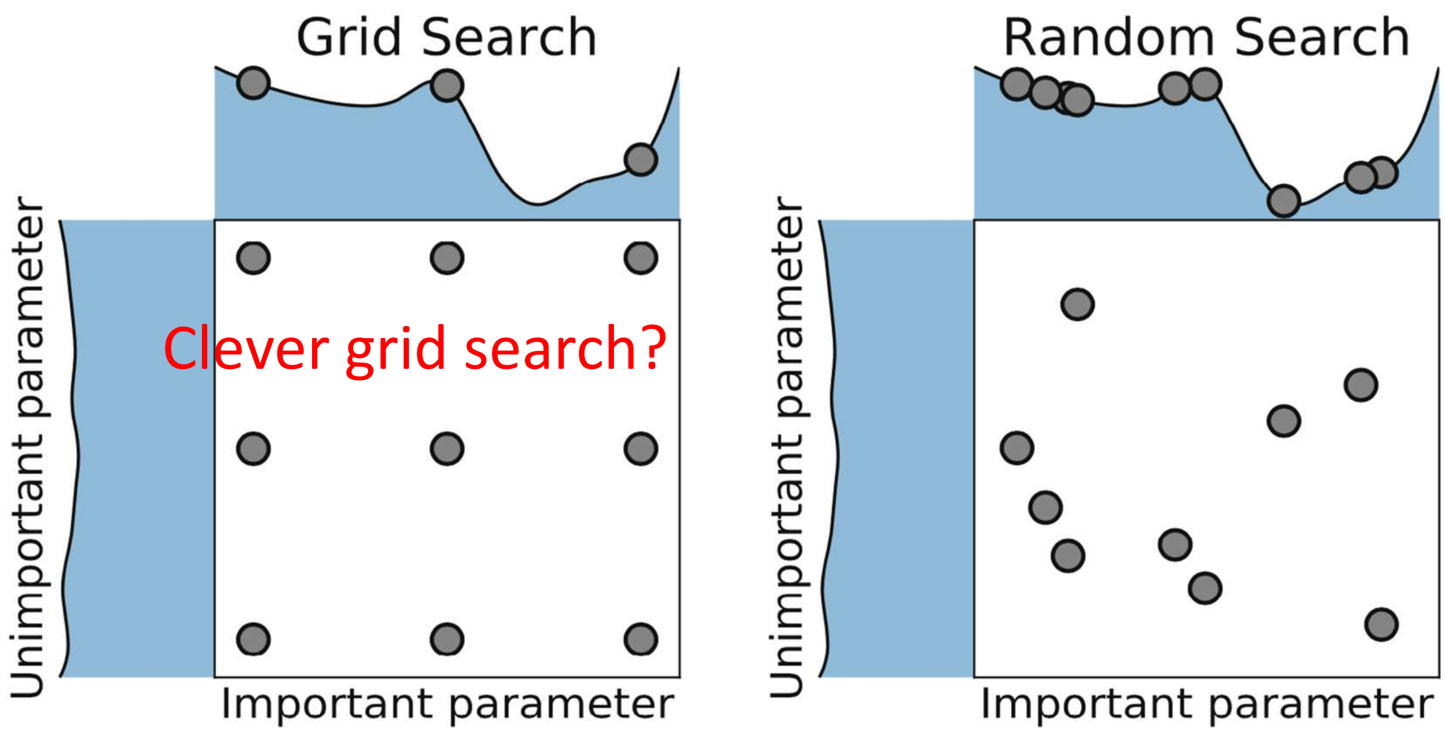
How would you solve the AutoML problem?

Simple/intuitive methods …



Feurer M., Hutter F. (2019) **Hyperparameter Optimization**. In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham

Simple/intuitive methods …



Feurer M., Hutter F. (2019) **Hyperparameter Optimization**. In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham

Overview of AutoML techniques

First wave

Second wave

Third wave

Year	Ref.	Method	Type	Description	Innovative aspects
2006	[6, 7]	PSMS	β	Vectorial representation of solutions, PSO used as optimizer, subsampling, CV	Formulation of the full model selection task
2007	[13, 12]	Heterogeneous surrogate evolution	β	Parallel co evolution of models, ensemble generation	Returned ensemble of solutions, large and heterogenous space of models
2010	[9]	Ensemble PSMS	β	Enhanced PSMS with ensemble of solutions	Returned an ensemble of solutions as output
2012	[44]	GPS: GA-PSO-FMS	β	GAs were used to search for a model template, PSO was used for hyperparameter optimization	Separation of template search and hyperparameter optimization
2013	[45]	Auto-WEKA	γ	SMBO with SMAC, approached the CASH problem	Definition of the combined algorithm selection and hyperparameter optimization problem
2014	[41]	Multi-objective surrogate-based FMS	γ	Multi objective (complexity/performance) evolutionary method, surrogates were used to approximate the fitness function	Among the first methods using a meta-learner for AutoML, multi-objective formulation
2015	[11]	AutoSKLearn	γ	SMBO, warm starting with a classifier, ensemble generation	AutoML definition, warm-starting with meta-learner, winner of AutoML challenge
2016	[37, 38]	TPOT	β	Genetic programming / NSGA-II selwection, cross validation, data sampling	Models naturally codified as GP trees
2019	[54, 28]	NAS /AutoKeras	γ	SMBO for Neural Architecture search	Kernel function for comparison of architectures

Overview of AutoML techniques

First wave

Second wave

Third wave

Year	Ref.	Method	Type	Description	Innovative aspects
2006	[6, 7]	PSMS	β	Vectorial representation of solutions, PSO used as optimizer, subsampling, CV	Formulation of the full model selection task
2007	[13, 12]	Heterogeneous surrogate evolution	β	Parallel co evolution of models, ensemble generation	Returned ensemble of solutions, large and heterogenous space of models
2010	[9]	Ensemble PSMS	β	Enhanced PSMS with ensemble of solutions	Returned an ensemble of solutions as output
2012	[44]	GPS: GA-PSO-FMS	β	GAs were used to search for a model template, PSO was used for hyperparameter optimization	Separation of template search and hyperparameter optimization
2013	[45]	Auto-WEKA	γ	SMBO with SMAC, approached the CASH problem	Definition of the combined algorithm selection and hyperparameter optimization problem
2014	[41]	Multi-objective surrogate-based FMS	γ	Multi objective (complexity/performance) evolutionary method, surrogates were used to approximate the fitness function	Among the first methods using a meta-learner for AutoML, multi-objective formulation
2015	[11]	AutoSKLearn	γ	SMBO, warm starting with a classifier, ensemble generation	AutoML definition, warm-starting with meta-learner, winner of AutoML challenge
2016	[37, 38]	TPOT	β	Genetic programming / NSGA-II selwection, cross validation, data sampling	Models naturally codified as GP trees
2019	[54, 28]	NAS /AutoKeras	γ	SMBO for Neural Architecture search	Kernel function for comparison of architectures

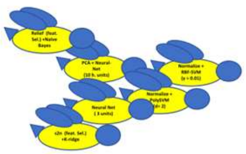
Please note we account for models that aim at optimizing classification pipelines. There are earlier efforts for related tasks like model selection, algorithm selection, meta-learning, hyper-parameter optimization etc.

Early efforts on FMS/AutoML

- Interest from diverse fronts (early):
  - Research agencies (e.g., IARPA, DARPA)
  - Industry (e.g., Google, IBM, ORACLE, Microsoft)
  - Researchers (machine learning, computational intelligence)

Early efforts on FMS/AutoML

- GEMS: AutoML for MGE Data (nested CV)
- SUMO: Genetic algorithms + ensemble models for surrogate modeling (regression)
- PSMS: Particle swarm optimization for FMS (with an ensemble version - EPSMS)



- Classical/related approaches:
  - Grid search, Meta-Learning, AS, MS, HO…

A. Statnikov, I. Tsamardinos, Y. Dosbayev, C.F. Aliferis. **GEMS: A System for Automated Cancer Diagnosis and Biomarker Discovery from Microarray Gene Expression Data**. International Journal of Medical Informatics, 2005 Aug;74(7-8):491-503.

H. J. Escalante. **Towards a Particle Swarm Model Selection algorithm.** *Multi-level inference workshop and model selection game, NIPS 2006.*

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

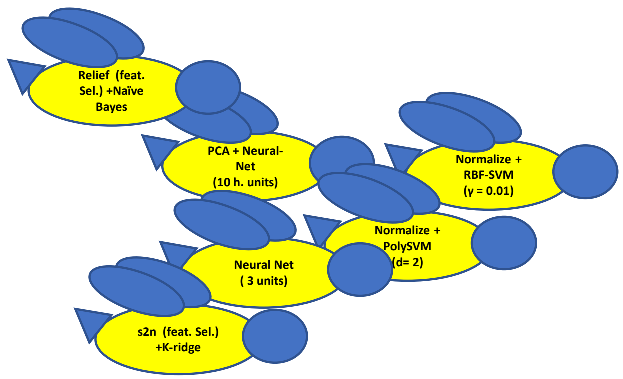
D. Gorissen, T. Dhaene, F. de Turck. **Evolutionary Model Type Selection for Global Surrogate Modeling.** *In Journal of Machine Learning Research,* 10(Jul):2039-2078, 2009

N. Fananananapazir, A. Statnikov, C.F. Aliferis. **The Fast-AIMS Clinical Mass Spectrometry Analysis System**. Advaces in Bioinformatics, 2009, Article ID 598241.

Particle swarm model selection

PSMS: our approach to FMS

- **Particle swarm model selection**: Use ***particle swarm optimization*** for exploring the search space of full models in a particular ML-toolbox



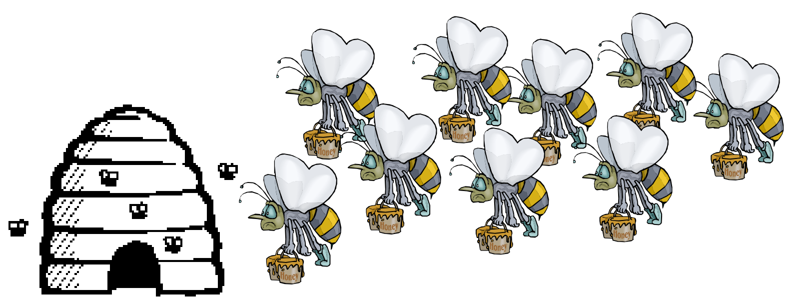
H. J. Escalante. **Towards a Particle Swarm Model Selection algorithm.** *Multi-level inference workshop and model selection game, NIPS, Whistler, VA, BC, Canada, 2006.*

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

Particle swarm optimization

- Population-based search heuristic

- Inspired on the behavior of biological communities that exhibit local and social behaviors



A. P. Engelbrecht. **Fundamentals of Computational Swarm Intelligence.** Wiley,2006

Particle swarm optimization

- Each individual (particle) *i* has:
  - A position in the search space ($\mathbf{X}_i^t$), which represents a solution to the problem at hand,
  - A velocity vector ($\mathbf{V}_i^t$), which determines how a particle explores the search space

- After random initialization, particles update their positions according to:

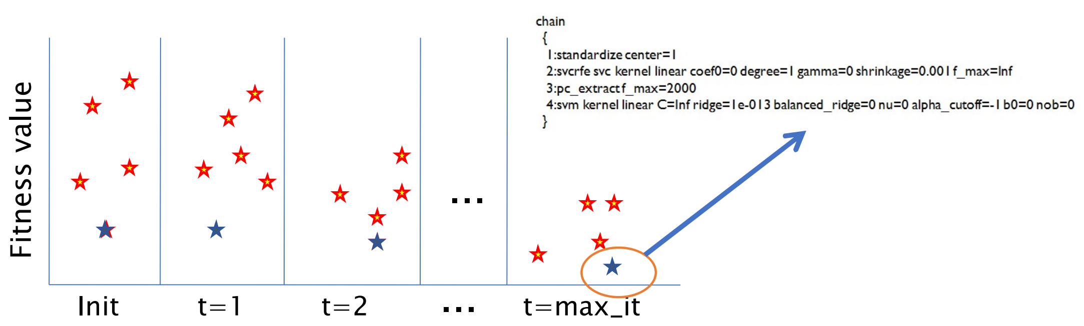
$$\mathbf{x}^{t+1}{}_i = \mathbf{v}^{t+1}{}_i + \mathbf{x}^t{}_i$$

$$\mathbf{v}^{t+1}{}_i = \phi_0 \times \mathbf{v}^t{}_i + \phi_1 \boxed{\times (\mathbf{p}_i - \mathbf{x}^t{}_i)} + \phi_2 \boxed{\times (\mathbf{p}_g - \mathbf{x}^t{}_i)}$$

A. P. Engelbrecht. **Fundamentals of Computational Swarm Intelligence.** Wiley,2006

Particle swarm optimization

1. Randomly initialize a population of particles (i.e., the swarm)
2. Repeat the following iterative process until stop criterion is meet:
   a) Evaluate the fitness of each particle
   b) Find personal best ($\mathbf{p}_i$) and global best ($\mathbf{p}_g$)
   c) Update particles
   d) Update best solution found (if needed)
3. Return the best particle (solution)



PSMS : PSO for full model selection

- Set of methods (not restricted to this set)

	Object name	Type	# pars.	Description
Classification	*zarbi*	C	0	Linear classifier
	naive	C	0	Naïve Bayes
	logitboost	C	3	Boosting with trees
	neural	C	4	Neural network
	svc	C	4	SVM classifier
	kridge	C	4	Kernel ridge regression
	rf	C	3	Random forest
	lssvm	C	5	Kernel ridge regression
Feature selection	*Ftest*	F	4	F-test criterion
	Ttest	F	4	T-test criterion
	aucfs	F	4	AUC criterion
	odds-ratio	F	4	Odds ratio criterion
	relief	F	3	Relief ranking criterion
	Pearson	F	4	Pearson correlation coefficient
	ZFilter	F	2	Statistical filter
	s2n	F	2	Signal-to-noise ratio
	$pc - extract$	F	1	Principal components analysis
	svcrfe	F	1	SVC- recursive feature elimination
Preprocessing	*normalize*	P	1	Data normalization
	standardize	P	1	Data standardization
	$shift - scale$	P	1	Data scaling

http://clopinet.com/CLOP

PSMS : PSO for full model selection

- Codification of solutions as real valued vectors

Hyperparameters for the selected methods

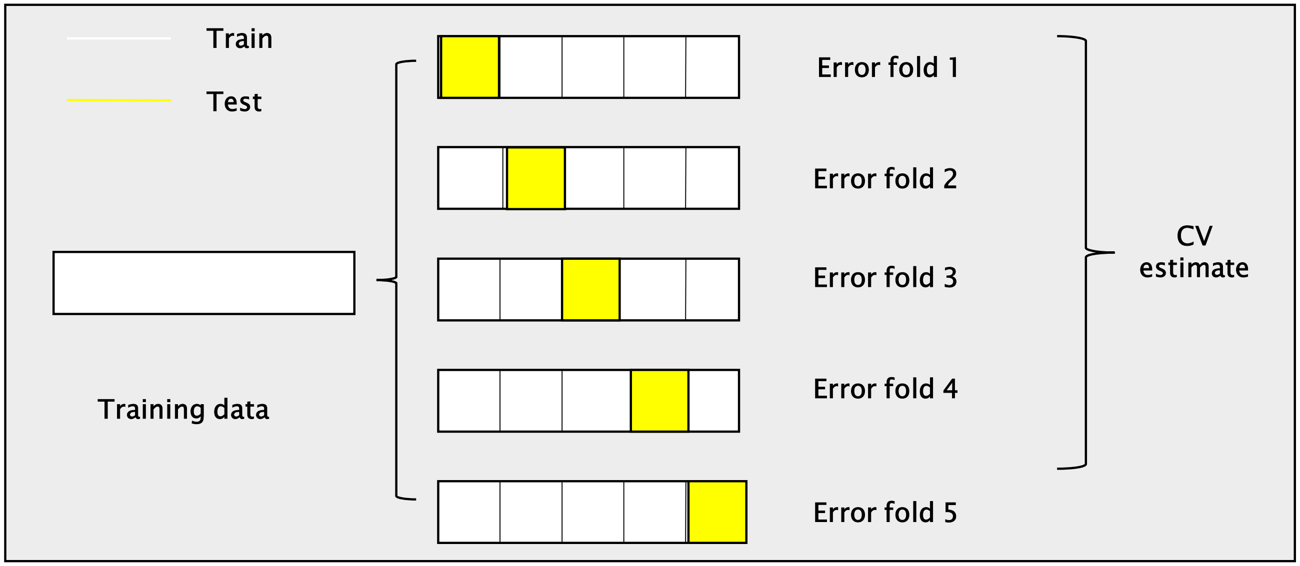
Preprocessing
before feature
selection?

$$\mathbf{x}_i = \left\langle \boxed{x_{\mathrm{i,pre}}}, \boxed{y_{\mathrm{i,1...Npre}}}, \boxed{x_{\mathrm{i,fs}}}, \boxed{y_{\mathrm{i,1...Nfs}}}, \boxed{x_{\mathrm{i,sel}}}, \boxed{x_{\mathrm{i,class}}}, \boxed{y_{\mathrm{i,1...Nclass}}} \right\rangle$$

Choice of methods for preprocessing, feature selection and classification

PSMS : PSO for full model selection

- Fitness function:
  - K-fold cross-validation balanced error rate
  - K-fold cross-validation area under the ROC curve



Some experimental results

PSMS in the ALvsPK challenge

- Five data sets for binary classification
- Goal: to obtain the best classification model for each data set
- Two tracks:
  - Prior knowledge
  - Agnostic learning

Dataset	Domain	Number of examples (training/validation /test)	Positive class (% num. ex.)	Number of features	
				Raw data (for PK)	Preprocessed (for AL)
ADA	Marketing	4147 / 415 / 41471	28.4	14	48
GINA	HWR	3153 / 315 / 31532	49.2	784	970
HIVA	Drug discovery	3845 / 384 / 38449	3.5	Molecules	1617
NOVA	Text classification	1754 / 175 / 17537	28.5	Text	16969
SYLVA	Ecology	13086 / 1309 / 130857	6.2	108	216

http://www.agnostic.inf.ethz.ch/

PSMS in the ALvsPK challenge

- Best configuration of PSMS:

Entry	Description	Ada	Gina	Hiva	Nova	Sylva	Overall	Rank
Interim-all-prior	Best PK	17.0	2.33	27.1	4.71	0.59	10.35	1st
psmsx_jmlr_run_I	PSMS	16.86	**2.41**	**28.01**	5.27	**0.62**	**10.63**	2nd
Logitboost-trees	Best AL	**16.6**	3.53	30.1	**4.69**	0.78	11.15	8th

Comparison of the performance of models selected with PSMS with that obtained by other techniques in the ALvsPK challenge

Data	SF	Model	Time (m)	Test-BER
Ada	1	chain({logitboost(units=469,shrinkage=0.4,depth=1),bias}	368.12	16.86
Gina	2	chain({sns(1),relief(fmax=487),gkridge,bias}	482.23	2.41
Hiva	3	chain({norm(1),rffs(fmax=1001),lssvm(gamma=0.096),bias}	124.54	28.01
Nova	1	chain({rffs(fmax=338),norm(1),std(1),sns(1),gkridge,bias}	82.12	5.27
Sylva	10	chain({sns(1),odds-ratio(fmax=60),gkridge,bias}	787.58	0.62

Models selected with PSMS for the different data sets

http://www.agnostic.inf.ethz.ch/results.php

PSMS in the ALvsPK challenge

- Official ranking:

Rank	Method	Balanced Error			Area Under Curve			Date	Name
		Train	Valid	Test	Train	Valid	Test		
1	interim all prior	0.0305	0.0934	0.1035	0.9893	0.9113	0.9332	2007-01-31 21:54:23	reference (gcc)
2	psmsx_jmlr_run_I	0.0482	0.0669	0.1063	0.9508	0.9355	0.8941	2008-10-18 01:29:25	H. Jair Escalante
3	psmsx_jmlr_run	0.0481	0.0691	0.1065	0.9513	0.9348	0.8938	2008-10-18 01:20:04	H. Jair Escalante
4	the bad	0.033	0.1002	0.1085	0.9876	0.9269	0.9332	2006-11-11 19:08:00	reference
5	the ugly	0.034	0.1016	0.1086	0.9873	0.9222	0.9328	2006-11-12 15:35:54	reference
6	vn3	0.0634	0.0744	0.1095	0.9464	0.9267	0.8949	2007-07-27 04:56:02	Vladimir Nikulin
7	cross-indexing-prior-1	0.0434	0.1284	0.1099	0.9835	0.9183	0.9308	2007-01-25 16:50:26	Juha Reunanen
8	cross-indexing-prior-1a	0.0497	0.0682	0.11	0.9782	0.9712	0.9312	2007-02-14 20:25:07	Juha Reunanen
9	Doubleboost	0.0411	0.0429	0.1114	0.9586	0.9605	0.8896	2007-02-20 14:30:27	Roman Lutz
10	LogitBoost with trees	0.0585	0.1056	0.1115	0.974	0.9323	0.9303	2006-10-10 18:05:48	Roman Lutz
11	LogitBoost with trees	0.0415	0.0432	0.1117	0.9583	0.9598	0.8892	2007-02-20 13:47:47	Roman Lutz
12	cross-indexing-prior-3	0.0415	0.049	0.1119	0.9846	0.9811	0.932	2007-06-10 01:14:52	Juha Reunanen
13	LogitBoost with trees	0.0585	0.1076	0.1121	0.974	0.9298	0.9301	2006-10-09 13:42:41	Roman Lutz
14	the good	0.0384	0.1088	0.1125	0.9827	0.9182	0.9299	2006-11-12 15:59:31	reference
15	SVM+GbO+trees	0.0382	0.0958	0.1139	0.9601	0.9086	0.882	2006-11-15 00:31:02	Vladimir Nikulin
16	cross-indexing-prior-2	0.0415	0.049	0.1141	0.9846	0.9811	0.9315	2007-02-25 21:42:16	Juha Reunanen
17	serate quadratic lssvm	0.0361	0.1219	0.1142	0.9841	0.9158	0.9281	2006-10-20 10:44:13	reference
18	out1-fs-nored-val (Intel final 1)	0.0698	0.0616	0.1142	0.9313	0.9406	0.8859	2007-07-31 02:14:16	IDEAL, Intel
19	vn1	0.0723	0.0721	0.1145	0.9517	0.9469	0.8987	2007-07-27 04:50:52	Vladimir Nikulin
20	cubic lssvm	0.0365	0.1057	0.1145	0.9842	0.9221	0.9301	2006-10-11 20:15:41	reference

http://www.agnostic.inf.ethz.ch/results.php

Some results in benchmark data

- Comparison of PSMS and pattern search

ID	Data set	PATSMS test-BER	PSMS test-BER	PATSMS CV-BER	PSMS CV-BER
1	Breast-cancer	36.98 ± 0.08	**33.59 ± 0.12**	**32.64 ± 0.06**	32.96 ± 0.01
2	Diabetes	26.07 ± 0.03	**25.37 ± 0.02**	**25.39 ± 0.02**	26.48 ± 0.05
3	Flare-solar	32.87 ± 0.02	**32.65 ± 0.01**	**32.69 ± 0.01**	33.13 ± 0.01
4	German	28.65 ± 0.02	**28.28 ± 0.02**	**31.00 ± 0.00**	31.02 ± 0.00
5	Heart	19.50 ± 0.19	**17.35 ± 0.06**	**16.96 ± 0.07**	19.93 ± 0.03
6	Image	3.58 ± 0.01	**2.50 ± 0.01**	**11.54 ± 0.10**	15.88 ± 0.04
7	Splice	13.94 ± 0.99	**9.46 ± 0.25**	**18.01 ± 0.05**	19.15 ± 0.07
8	Thyroid	10.84 ± 0.39	**5.98 ± 0.06**	**11.15 ± 0.20**	15.49 ± 0.12
9	Titanic	29.94 ± 0.00	**29.60 ± 0.00**	**27.19 ± 0.13**	27.32 ± 0.13

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

Some results in benchmark data

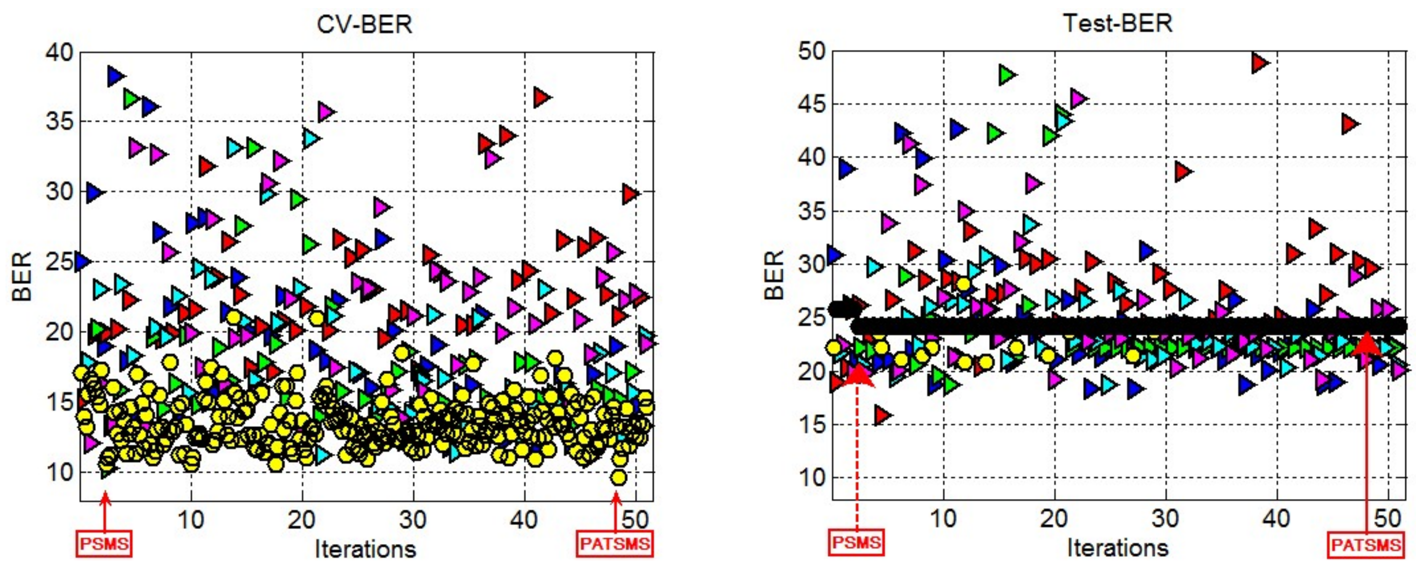
- Comparison of PSMS and pattern search

ID	Data set	PATSMS test-BER	PSMS test-BER	PATSMS CV-BER	PSMS CV-BER
1	Breast-cancer	36.98 ± 0.08	**33.59 ± 0.12**	**32.64 ± 0.06**	32.96 ± 0.01
2	Diabetes	26.07 ± 0.03	**25.37 ± 0.02**	**25.39 ± 0.02**	26.48 ± 0.05
3	Flare-solar	32.87 ± 0.02	**32.65 ± 0.01**	**32.69 ± 0.01**	33.13 ± 0.01
4	German	28.65 ± 0.02	**28.28 ± 0.02**	**31.00 ± 0.00**	31.02 ± 0.00
5	Heart	19.50 ± 0.19	**17.35 ± 0.06**	**16.96 ± 0.07**	19.93 ± 0.03
6	Image	3.58 ± 0.01	**2.50 ± 0.01**	**11.54 ± 0.10**	15.88 ± 0.04
7	Splice	13.94 ± 0.99	**9.46 ± 0.25**	**18.01 ± 0.05**	19.15 ± 0.07
8	Thyroid	10.84 ± 0.39	**5.98 ± 0.06**	**11.15 ± 0.20**	15.49 ± 0.12
9	Titanic	29.94 ± 0.00	**29.60 ± 0.00**	**27.19 ± 0.13**	27.32 ± 0.13

H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

Some results in benchmark data

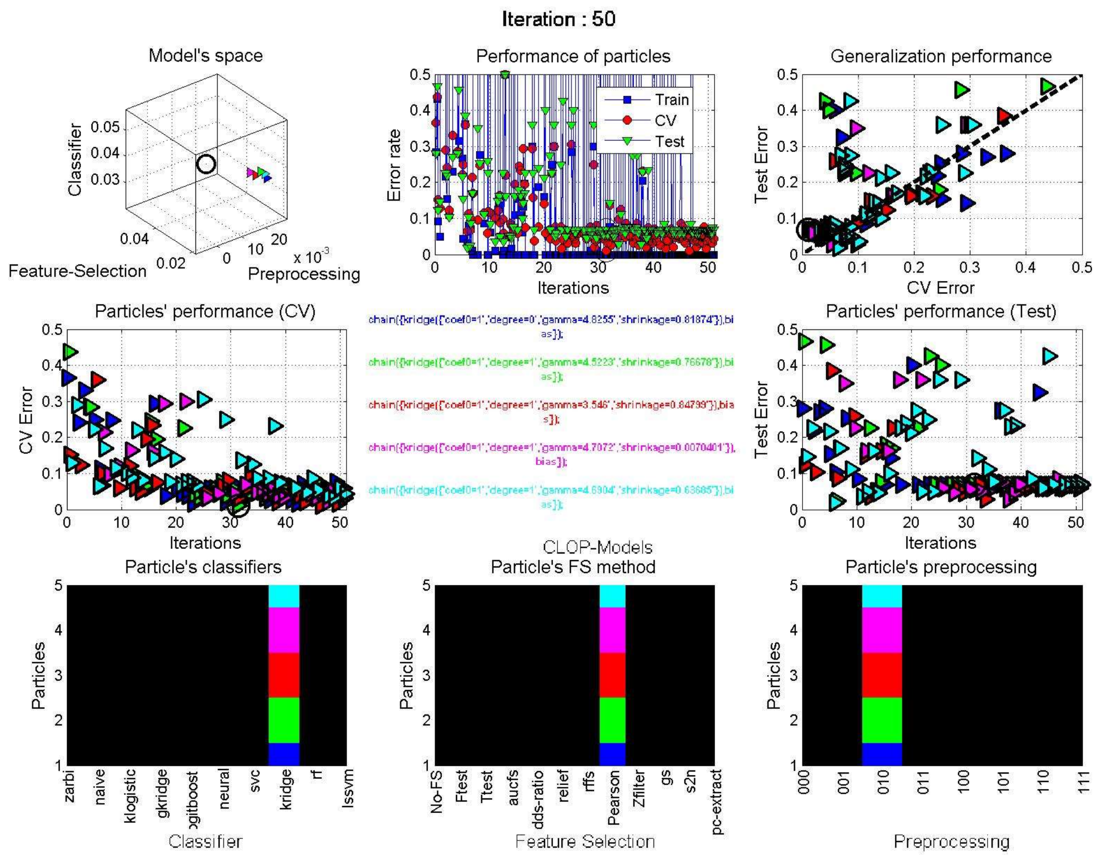
- Comparison of PSMS and pattern search



H. J. Escalante, E. Sucar, M. Montes. **Particle Swarm Model Selection,** *In Journal of Machine Learning Research, 10(Feb):405--440, 2009.*

PSMS: Interactive demo

http://clopinet.com/CLOP



Isabelle Guyon, Amir Saffari, Hugo Jair Escalante, Gokan Bakir, and Gavin Cawley, **CLOP: a Matlab Learning Object Package.** *NIPS 2007 Demonstrations, Vancouver, British Columbia, Canada 2007.*

Other applications of PSMS/EPSMS

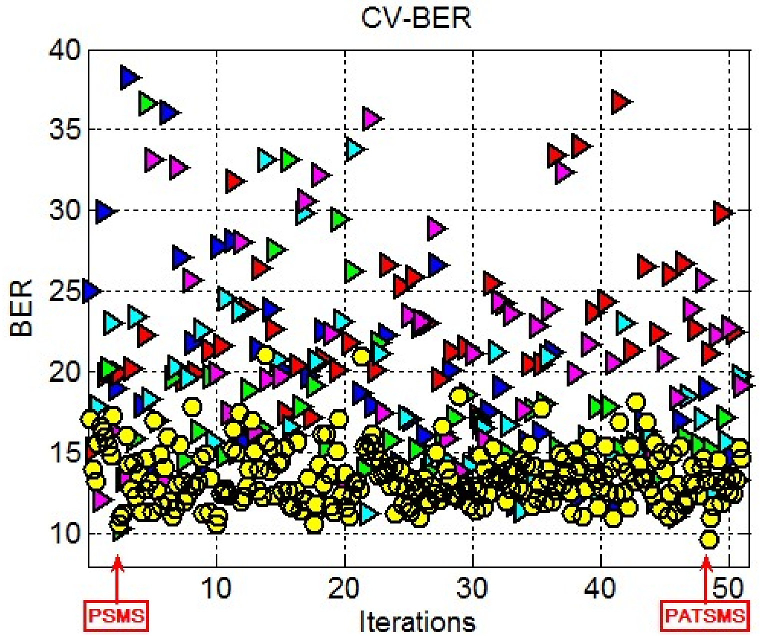
- Successful:
  - Acute leukemia classification
  - Authorship verification (Spanish/English)
  - Authorship attribution
  - Region labeling
  - ML Challenges

- Not successful:
  - Review recommendation (14 features)
  - Region labeling (~90 classes)
  - Sentiment analysis on speech signals (high p – small n)
  - Plagiarism detection (a few samples)
  - ML Challenges

Ensemble PSMS

Ensemble PSMS

- Many models are evaluated during the search process of PSMS; although a single model is selected



Ensemble PSMS

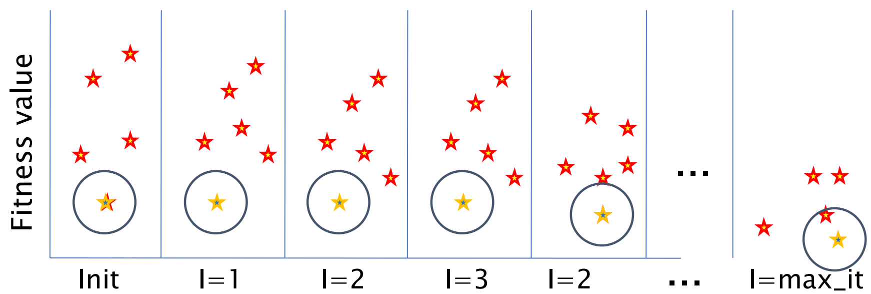
- **Idea:** taking advantage of the large number of models that are evaluated during the search for building ensemble classifiers
- **Problem:** How to select the partial solutions from PSMS so that they are accurate and diverse to each other
- **Motivation:** The success of ensemble classifiers depends mainly in two key aspects of individual models: ***Accuracy and diversity***

Ensemble PSMS

- How to select potential models for building ensembles?
  - **BS:** store the global best model in each iteration
  - **BI:** the best model in each iteration
  - **SE:** combine the outputs of the final swarm

- How to fuse the outputs of the selected models?
  - Simple (un-weighted) voting

$$g(E) = \frac{1}{L}\sum_{l=1}^{L} f_l(\mathbf{p})$$

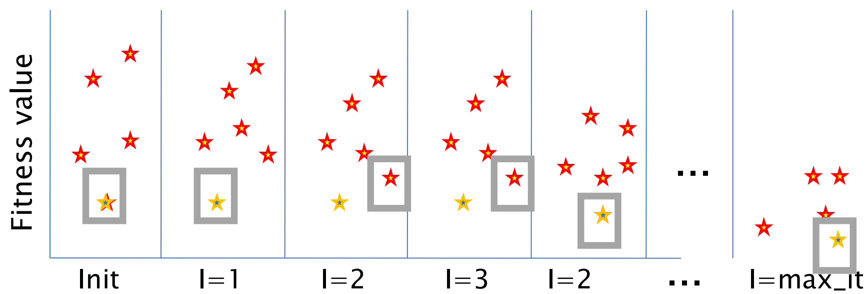


Ensemble PSMS

- How to select potential models for building ensembles?
  - **BS:** store the global best model in each iteration
  - **BI:** the best model in each iteration
  - **SE:** combine the outputs of the final swarm

- How to fuse the outputs of the selected models?
  - Simple (un-weighted) voting

$$g(E) = \frac{1}{L}\sum_{l=1}^{L} f_l(\mathbf{p})$$

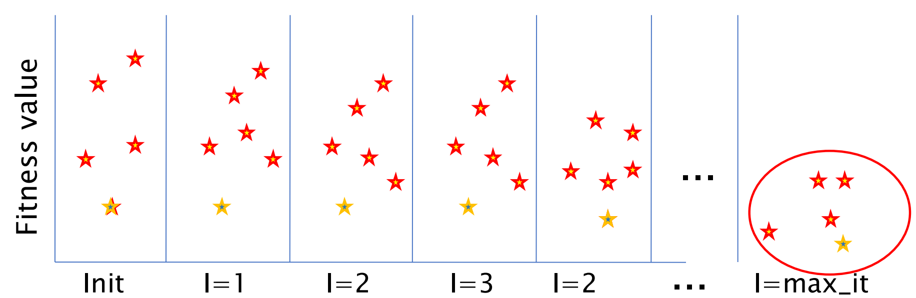


Ensemble PSMS

- How to select potential models for building ensembles?
  - **BS:** store the global best model in each iteration
  - **BI:** the best model in each iteration
  - **SE:** combine the outputs of the final swarm

- How to fuse the outputs of the selected models?
  - Simple (un-weighted) voting

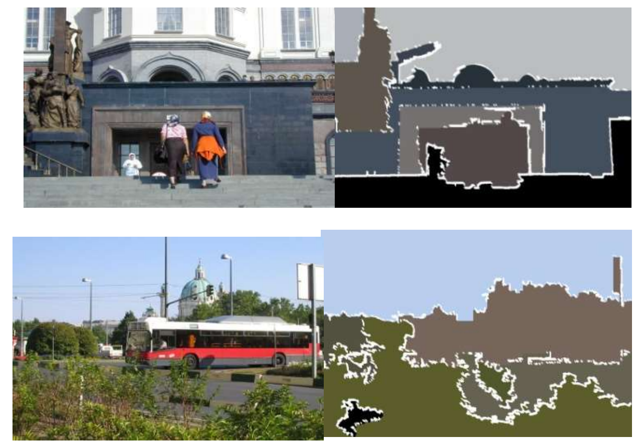
$$g(E) = \frac{1}{L}\sum_{l=1}^{L} f_l(\mathbf{p})$$



Experimental evaluation

- Data:
  - 9 Benchmark machine learning data sets (binary classification)
  - 1 Object recognition data set (multiclass, 10 classes)

ID	Data set	Training	Testing	Features
1	Breast-cancer	200	77	9
2	Diabetes	468	300	8
3	Flare solar	666	400	9
4	German	700	300	20
5	Heart	170	100	13
6	Image	1300	1010	20
7	Splice	1000	2175	60
8	Thyroid	140	75	5
9	Titanic	150	2051	3
OR	*SCEF*	*2378*	*3300*	*50*



H. J. Escalante, M. Montes, E. Sucar. **Ensemble Particle Swarm Model Selection.** Proceedings of the International Joint Conference on Neural Networks (IJCNN2010 – WCCI2010), pp. 1814--1821, IEEE,, 2010 [**Best Student Paper Award**].

Experimental results

- Evaluation:
  - Average of area under the ROC curve (performance)
  - Coincident failure diversity (ensemble diversity)

$$CFD = \begin{cases} \dfrac{1}{1-p_0}\displaystyle\sum_{r=1}^{L}\dfrac{L-r}{L-1}p_r & \text{If } p_0 < 1 \\ 0 & \text{If } p_0 = 1 \end{cases}$$

W. Wang. **Some fundamental issues in ensemble methods.** Proc. Of IJCNN07, pp. 2244–2251, IEEE, July 2007.

Experimental results: performance

- **Benchmark data sets:** better performance is obtained by ensemble methods

ID	PSMS	EPSMS-BS	EPSMS-SE	EPSMS-BI
1	72.03±2.24	73.40±0.78	74.05±0.91	**74.35±0.49**
2	82.11±1.29	82.60±1.52	74.07±13.7	**83.42±0.46**
3	68.81±4.31	69.38±4.53	70.13±7.48	**72.16±1.42**
4	73.92±1.23	73.84±1.53	74.70±0.72	**74.77±0.69**
5	85.55±5.48	87.40±2.01	87.07±0.75	**88.36±0.88**
6	97.21±3.15	98.85±1.45	95.27±3.04	**99.58±0.33**
7	97.26±0.55	98.02±0.64	96.99±1.21	**98.84±0.26**
8	96.00±4.75	98.18±0.94	97.29±1.54	**99.22±0.45**
9	73.24±1.16	73.50±0.95	**75.37±1.05**	74.40±0.91
Avg.	82.90±2.68	83.91±1.59	82.77±3.38	**85.01±0.65**

Average accuracy over 10-trials of PSMS and EPSMS in benchmark data

H. J. Escalante, M. Montes, E. Sucar. **Ensemble Particle Swarm Model Selection.** Proceedings of the International Joint Conference on Neural Networks (IJCNN2010 – WCCI2010), pp. 1814--1821, IEEE,, 2010 [**Best Student Paper Award**].

Experimental results: Diversity of ensemble

- Diversity results

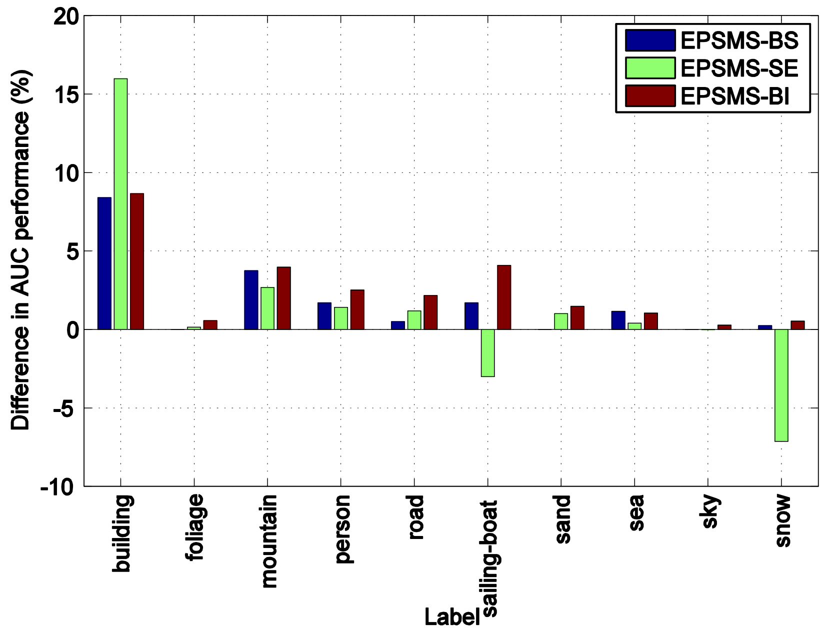
ID	EPSMS-BS	EPSMS-SE	EPSMS-BI
1	0.2055±0.1498	**0.5422±0.0550**	0.5017±0.1149
2	0.3547±0.1711	**0.6241±0.0619**	0.5081±0.0728
3	0.1295±0.1704	**0.4208±0.1357**	0.4012±0.1071
4	0.3019±0.1732	**0.5159±0.0596**	0.4296±0.0490
5	0.2733±0.1714	**0.5993±0.0925**	0.5647±0.0655
6	0.7801±0.0818	0.7555±0.0524	**0.8427±0.0408**
7	0.5427±0.3230	0.7807±0.0585	**0.8050±0.0294**
8	0.6933±0.1558	0.8173±0.0626	**0.8514±0.0403**
9	0.7473±0.0089	0.7473±0.0089	0.7473±0.0089
Avg.	0.4476±0.1562	**0.6448±0.0603**	0.6280±0.0588

EPSMS-SE models are more diverse

H. J. Escalante, M. Montes, E. Sucar. **Ensemble Particle Swarm Model Selection.** Proceedings of the International Joint Conference on Neural Networks (IJCNN2010 – WCCI2010), pp. 1814--1821, IEEE,, 2010 [**Best Student Paper Award**].

Experimental results: region labeling

ID	PSMS	EPSMS-BS	EPSMS-SE	EPSMS-BI
AUC	91.53±6.8	93.27±5.6	92.79±7.4	**94.05±5.3**
MCC	69.58%	76.59%	79.13%	**81.49%**

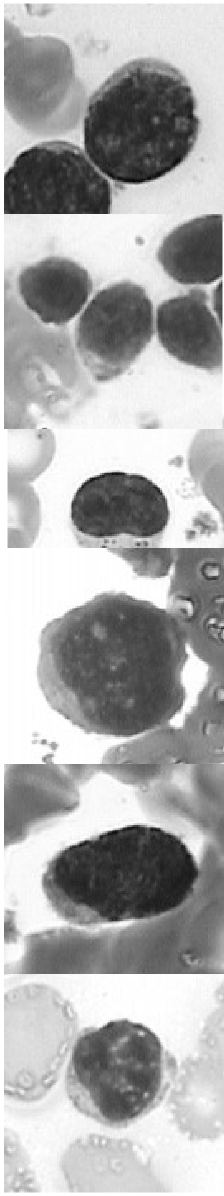


Per-concept improvement of EPSMS variants over straight PSMS

EPSMS for acute leukemia classification

- **Acute leukemia:** a malignant disease that affects a considerable portion of the world population

- There are different types and subtypes of acute leukemia, requiring different treatments.



Lessons learned

- Ensembles generated with EPSMS outperformed individual classifiers; including those selected with PSMS

- Models evaluated by PSMS are diverse to each other and accurate

- More stable predictions are obtained with the ensemble version of PSMS

Main findings from the early years

- Overfitting avoidance mechanisms
- Heterogeneous codification of solutions
- “Straight” global optimization approaches
- Data subsampling for efficient estimation of the objective function
- Assembling solutions
- Template-based initialization

Overview of AutoML techniques

First wave

Second wave

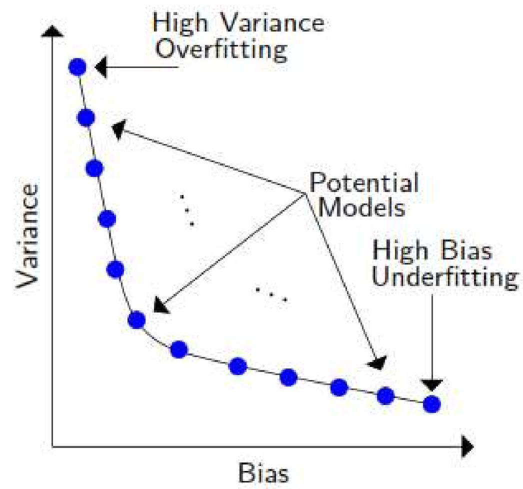
Third wave

Year	Ref.	Method	Type	Description	Innovative aspects
2006	[6, 7]	PSMS	β	Vectorial representation of solutions, PSO used as optimizer, subsampling, CV	Formulation of the full model selection task
2007	[13, 12]	Heterogeneous surrogate evolution	β	Parallel co evolution of models, ensemble generation	Returned ensemble of solutions, large and heterogenous space of models
2010	[9]	Ensemble PSMS	β	Enhanced PSMS with ensemble of solutions	Returned an ensemble of solutions as output
2012	[44]	GPS: GA-PSO-FMS	β	GAs were used to search for a model template, PSO was used for hyperparameter optimization	Separation of template search and hyperparameter optimization
2013	[45]	Auto-WEKA	γ	SMBO with SMAC, approached the CASH problem	Definition of the combined algorithm selection and hyperparameter optimization problem
2014	[41]	Multi-objective surrogate-based FMS	γ	Multi objective (complexity/performance) evolutionary method, surrogates were used to approximate the fitness function	Among the first methods using a meta-learner for AutoML, multi-objective formulation
2015	[11]	AutoSKLearn	γ	SMBO, warm starting with a classifier, ensemble generation	AutoML definition, warm-starting with meta-learner, winner of AutoML challenge
2016	[37, 38]	TPOT	β	Genetic programming / NSGA-II selwection, cross validation, data sampling	Models naturally codified as GP trees
2019	[54, 28]	NAS /AutoKeras	γ	SMBO for Neural Architecture search	Kernel function for comparison of architectures

Multi objective FMS

- Selecting models that optimize more than a single criterion combinations include:
  - Bias-variance
  - Performance-time
  - Performance-complexity

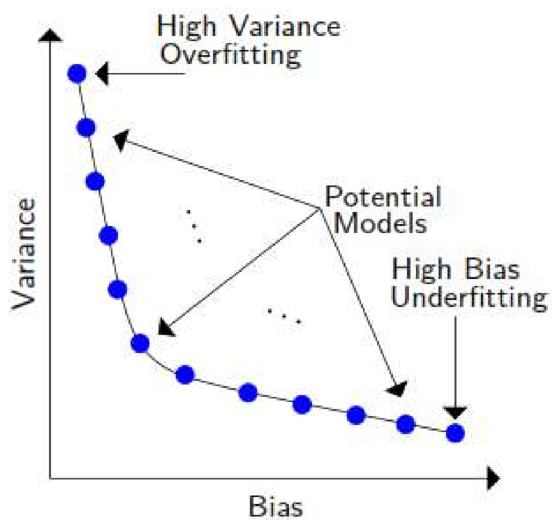
$$\begin{array}{ll} \text{minimize} & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_l(\mathbf{x})]^T \\ \text{subject to} & \mathbf{x} \in \mathcal{X} \end{array}$$

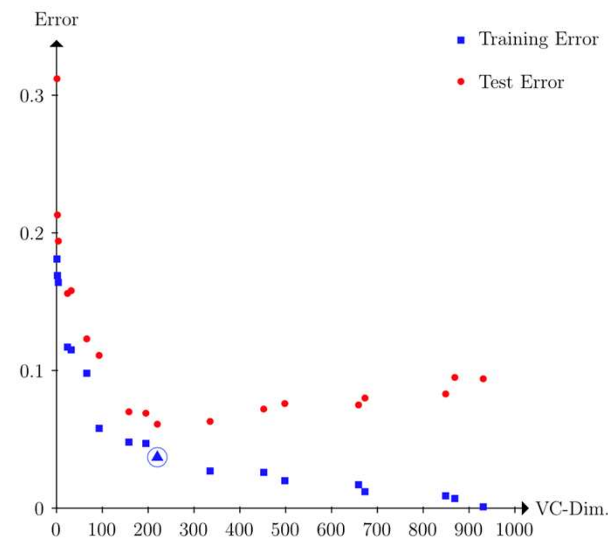


Alejandro Rosales-Pérez, Hugo Jair Escalante, Jesús A. González, Carlos A. Reyes, Carlos A. Coello. **Bias-variance multi-objective optimization for svms parameter selection .** IbPRIA 2013: 6th Iberian Conference on Pattern Recognition and Image Analysis, Madeira, Portugal. June 5-7, 2013

Multi objective FMS

- Selecting models that optimize more than a single criterion





Surrogate models were used here!

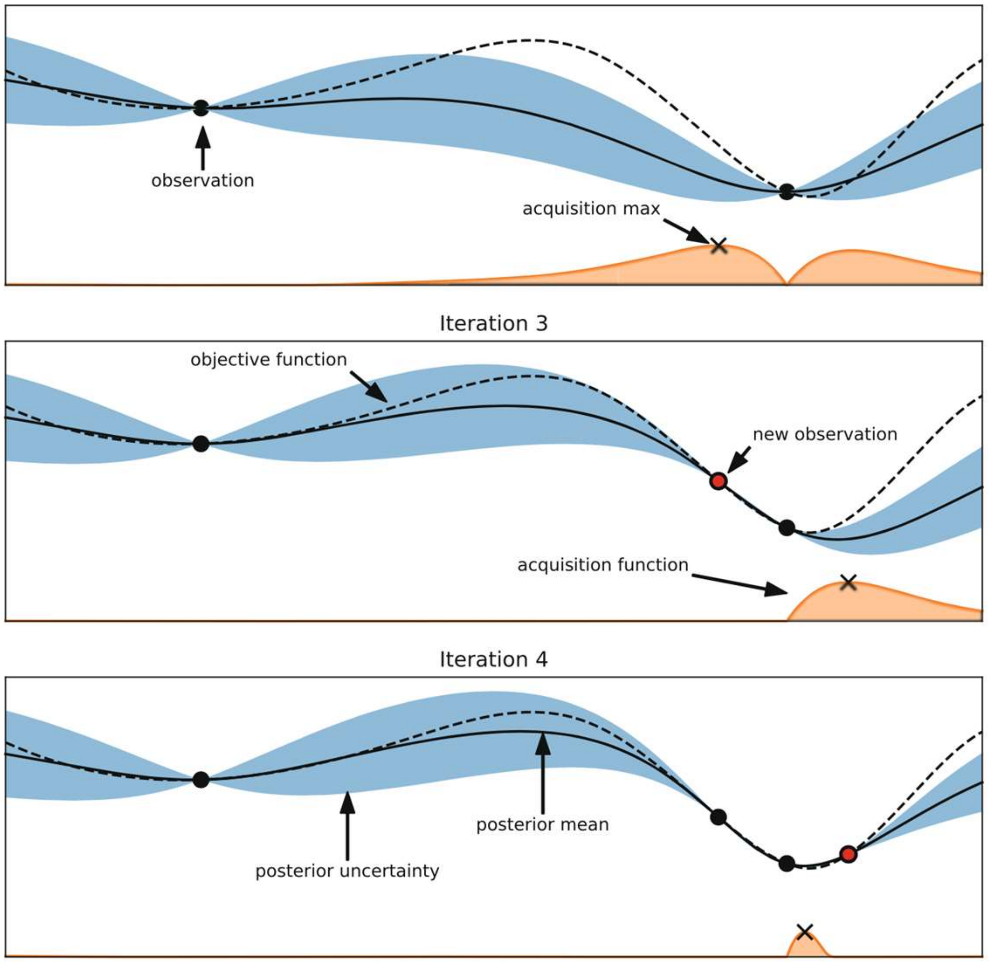
Alejandro Rosales-Pérez, Jesus A. Gonzalez, Carlos A. Coello Coello, Hugo Jair Escalante, Carlos A. Reyes García: **Multi-objective model type selection.** Neurocomputing 146: 83-94 (2014)

Alejandro Rosales-Pérez a,n, Jesus A. Gonzalez a, Carlos A. Coello Coello b, Hugo Jair Escalante a, Carlos A. Reyes-Garcia. **Surrogate-assisted multi-objective model selection for support vector machines.** Neurocomputing 150 (2015) 163–172, 2015

Bayesian optimization

- A global optimization procedure, based on two functions:
  - Surrogate,
    - Gaussian processes, random forest, etc
  - Acquisition functions

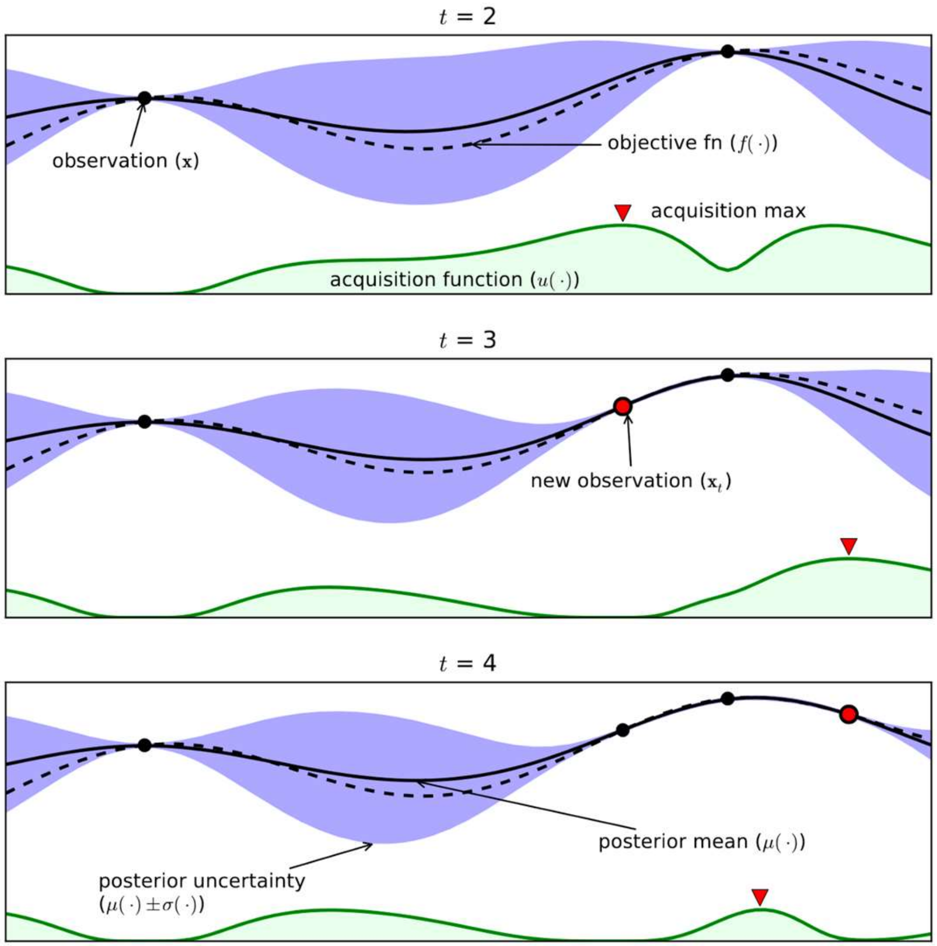
$$\mathbb{E}[\mathbb{I}(\boldsymbol{\lambda})] = \mathbb{E}[\max(f_{min} - y, 0)]$$



Feurer M., Hutter F. (2019) **Hyperparameter Optimization**. In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham

AutoWeka

- Introduced the definition of CASH, and approached the problem with Bayesian Optimization / Sequential Model-Based Optimization (SMBO)
  - BO/SMBO: sequential design strategy for global optimization of black-box functions that doesn't require derivatives.



https://www.cs.ubc.ca/labs/beta/Projects/autoweka/

C. Thornton, F. Hutter, H. Hoos, and K. Leyton-Brown. **Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms.** In Proc. KDD'13, pages 847–855, 2013.

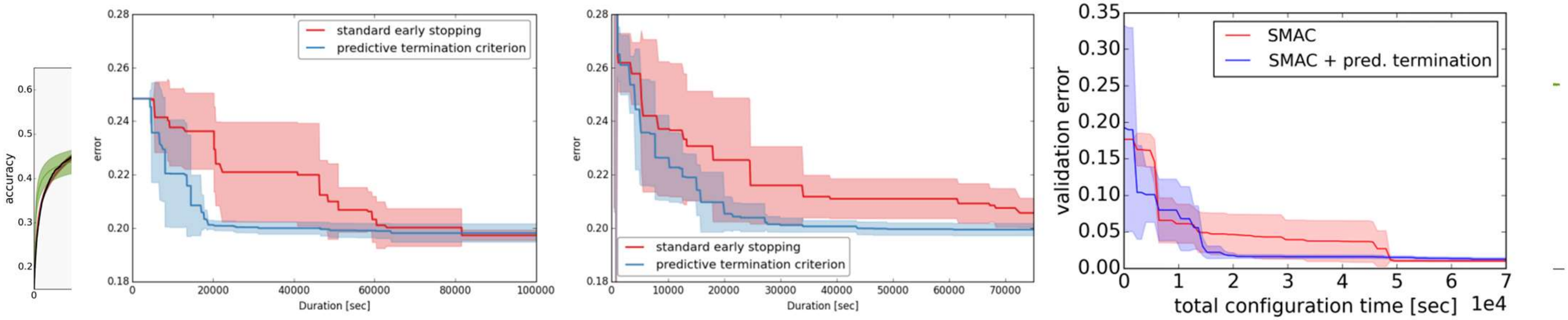
Multi fidelity approaches

- For AutoML solutions based on black box optimization, the evaluation of a candidate solution (model) is computationally expensive, and usually the more models are evaluated the better the performance of the AutoML methodology

- Methods aiming to make budget-constraint approximations of the real performance have been proposed

Multi fidelity approaches

- Learning curve-based performance:
  - Build and try to predict a learning curve (e.g., model performance vs. dataset size or model complexity)
  - Discard the model whenever the predictive curve is not promising



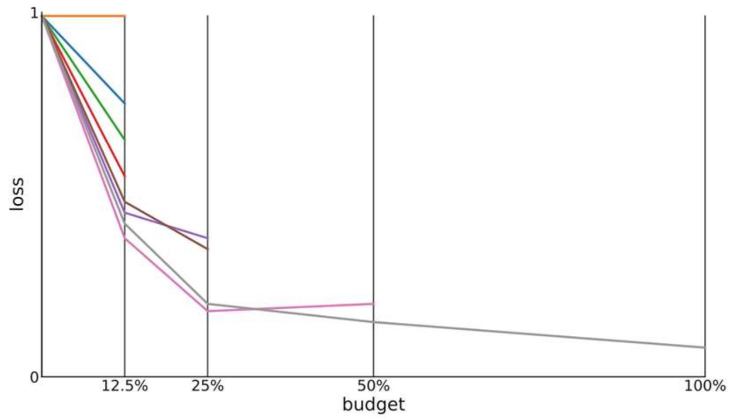
(a) SMAC on k-means CIFAR-10 (b) TPE on k-means CIFAR-10 (c) SMAC on MNIST

Tobias Domhan, Jost Tobias Springenberg, Frank Hutter **Speeding up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves.** Proc. IJCAI 2015

Multi fidelity approaches

- **Successive halving (SH):** Given a budget (time, other resources), evaluate all algorithms for that budget; then remove the half worst models from consideration; duplicate the budget and repeat until a single model remains.

- **Hyperband:** Generate random configurations of different budgets. Then using SH as subroutine



Algorithm 1: HYPERBAND algorithm for hyperparameter optimization.

input : R, η (default $\eta = 3$)
initialization : $s_{\max} = \lfloor \log_\eta(R) \rfloor$, $B = (s_{\max} + 1)R$

```
for s ∈ {s_max, s_max − 1, ..., 0} do
    n = ⌈(B/R) · η^s/(s+1)⌉,    r = R η^(−s)
    // begin SuccessiveHalving with (n, r) inner loop
    T = get_hyperparameter_configuration(n)
    for i ∈ {0, ..., s} do
        n_i = ⌊n η^(−i)⌋
        r_i = r η^i
        L = {run_then_return_val_loss(t, r_i) : t ∈ T}
        T = top_k(T, L, ⌊n_i/η⌋)
    end
end
return Configuration with the smallest intermediate loss seen so far.
```

K. Jamieson and Talwalkar. A. Non-stochastic Best Arm Identification and Hyperparameter Optimization. 2016
L. Li et al. Hyperband: **A Novel Bandit-Based Approach to Hyperparameter Optimization.** JMLR 2018

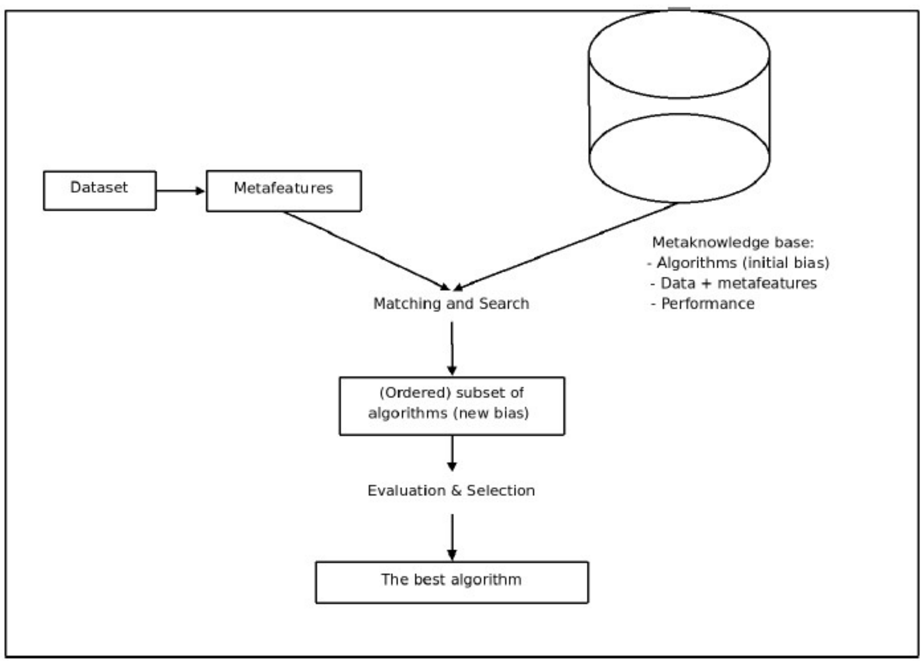
Meta-learning: learning to learn

- Evaluates and compares the application of learning algorithms on (many) previous tasks/domains to suggest learning algorithms (combinations, rankings) for new tasks
- Focuses on the relation between tasks/domains and learning algorithms
- Accumulating experience on the performance of multiple applications of learning methods

Brazdil P., Giraud-Carrier C., Soares C., Vilalta R. **Metalearning: Applications to Data Mining.** Springer Verlag. ISBN: 978-3-540-73262-4, 2008.

Brazdil P., Vilalta R, Giraud-Carrier C., Soares C.. **Metalearning.** Encyclopedia of Machine learning. Springer, 2010.

Meta-learning: learning to learn



Brazdil P., Giraud-Carrier C., Soares C., Vilalta R. **Metalearning: Applications to Data Mining.** Springer Verlag..

J. Vanschoren. Meta learning. Chapter in AutoML, CiML 2019

Brazdil P., Vilalta R, Giraud-Carrier C., Soares C.. **Metalearning.** Encyclopedia of Machine learning. Springer, 2010.

Table 2.1 Overview of commonly used meta-features. Groups from top to bottom: simple, statistical, information-theoretic, complexity, model-based, and landmarkers. Continuous features X and target Y have mean μ_X, stdev σ_X, variance σ_X^2. Categorical features $\mathtt{X}$ and class $\mathtt{C}$ have categorical values π_i, conditional probabilities $\pi_{i|j}$, joint probabilities $\pi_{i,j}$, marginal probabilities $\pi_{i+} = \sum_j \pi_{ij}$, entropy $H(\mathtt{X}) = -\sum_i \pi_{i+} log_2(\pi_{i+})$

Name	Formula	Rationale	Variants
Nr instances	n	Speed, Scalability [99]	p/n, $log(n)$, $log(n/p)$
Nr features	p	Curse of dimensionality [99]	$log(p)$, % categorical
Nr classes	c	Complexity, imbalance [99]	ratio min/maj class
Nr missing values	m	Imputation effects [70]	% missing
Nr outliers	o	Data noisiness [141]	o/n
Skewness	$\frac{E(X-\mu_X)^3}{\sigma_X^3}$	Feature normality [99]	min,max,μ,σ,q_1, q_3
Kurtosis	$\frac{E(X-\mu_X)^4}{\sigma_X^4}$	Feature normality [99]	min,max,μ,σ,q_1, q_3
Correlation	$\rho_{X_1 X_2}$	Feature interdependence [99]	min,max,μ,σ,ρ_{XY} [158]
Covariance	$cov_{X_1 X_2}$	Feature interdependence [99]	min,max,μ,σ,cov_{XY}
Concentration	$\tau_{X_1 X_2}$	Feature interdependence [72]	min,max,μ,σ,τ_{XY}
Sparsity	sparsity(X)	Degree of discreteness [143]	min,max,μ,σ
Gravity	gravity(X)	Inter-class dispersion [5]	
ANOVA p-value	$p_{val_{\mathtt{X}_1 \mathtt{X}_2}}$	Feature redundancy [70]	$p_{val_{XY}}$ [158]
Coeff. of variation	$\frac{\sigma_Y}{\mu_Y}$	Variation in target [158]	
PCA ρ_{λ_1}	$\sqrt{\frac{\lambda_1}{1+\lambda_1}}$	Variance in first PC [99]	$\frac{\lambda_1}{\sum_i \lambda_i}$ [99]
PCA skewness		Skewness of first PC [48]	PCA kurtosis [48]
PCA 95%	$\frac{dim_{95\%var}}{p}$	Intrinsic dimensionality [9]	
Class probability	$P(\mathtt{C})$	Class distribution [99]	min,max,μ,σ
Class entropy	$H(\mathtt{C})$	Class imbalance [99]	
Norm. entropy	$\frac{H(\mathtt{X})}{log_2 n}$	Feature informativeness [26]	min,max,μ,σ
Mutual inform.	$MI(\mathtt{C}, \mathtt{X})$	Feature importance [99]	min,max,μ,σ
Uncertainty coeff.	$\frac{MI(\mathtt{C},\mathtt{X})}{H(\mathtt{C})}$	Feature importance [3]	min,max,μ,σ
Equiv. nr. feats	$\frac{\overline{H(C)}}{\overline{MI(C,X)}}$	Intrinsic dimensionality [99]	
Noise-signal ratio	$\frac{\overline{H(X)}-\overline{MI(C,X)}}{\overline{MI(C,X)}}$	Noisiness of data [99]	
Fisher's discrimin.	$\frac{(\mu_{c1}-\mu_{c2})^2}{\sigma_{c1}^2-\sigma_{c2}^2}$	Separability classes c_1, c_2 [64]	See [64]
Volume of overlap		Class distribution overlap [64]	See [64]
Concept variation		Task complexity [180]	See [179, 180]
Data consistency		Data quality [76]	See [76]
Nr nodes, leaves	$\|\eta\|$, $\|\psi\|$	Concept complexity [113]	Tree depth
Branch length		Concept complexity [113]	min,max,μ,σ
Nodes per feature	$\|\eta_X\|$	Feature importance [113]	min,max,μ,σ
Leaves per class	$\frac{\|\psi_c\|}{\|\psi\|}$	Class complexity [49]	min,max,μ,σ
Leaves agreement	$\frac{n_{\psi_i}}{n}$	Class separability [16]	min,max,μ,σ
Information gain		Feature importance [16]	min,max,μ,σ, gini

(continued)

Meta learning

- Given:
  - prior tasks $t_j \in T$,
  - learning algorithms $\theta_i \in \Theta$
  - **P** the set of all prior evaluations $P_{i,j} = P(\theta_i, t_j)$ of configuration θ_i on task t_j, according to a predefined evaluation measure.
  - $\mathbf{P}_{new}$ the set of known evaluations $P_{i,new}$ on a new task t_{new}.

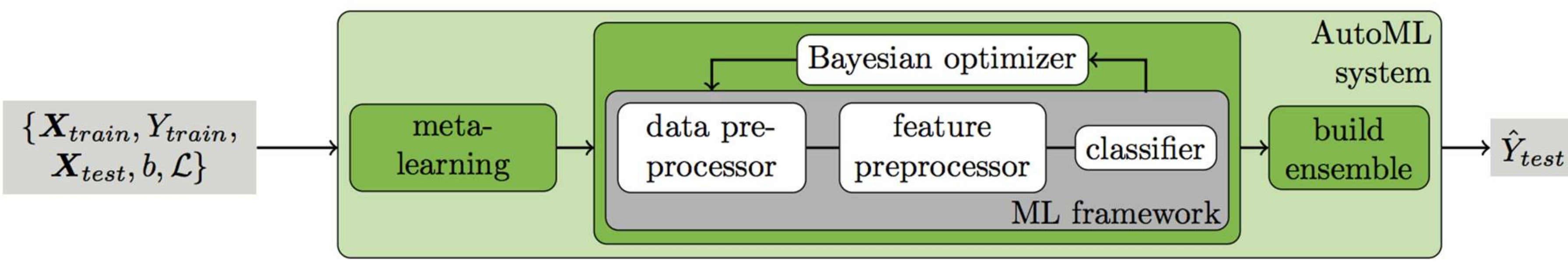
- We want to train a *meta- learner L* that predicts recommended configurations Θ_{new} for a new task t_{new}

- What for?
  - Few shot learning
  - Transfer learning
  - AutoML

- How?
  - ?

AutoSKlearn

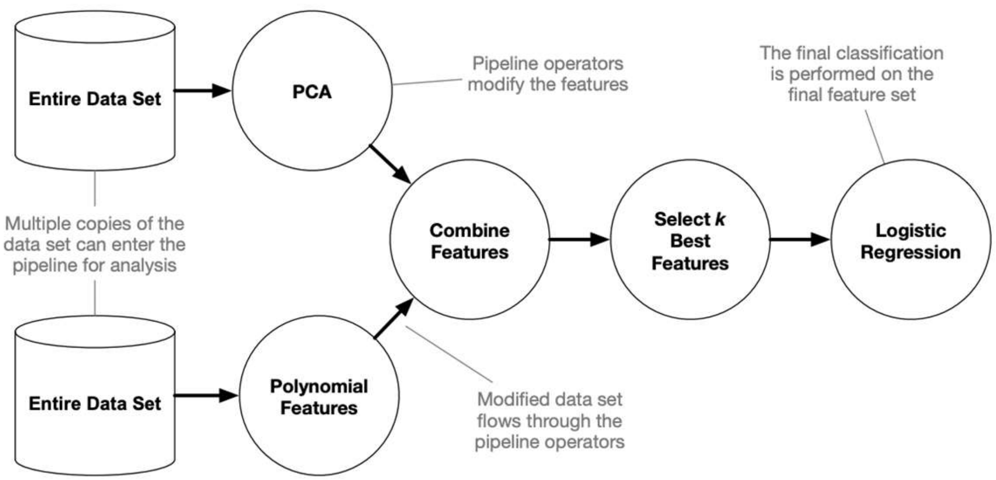
- Meta-learning + SMBO
- Trained a meta-learner on OpenML, uses it to warmstart the optimization process
- Ensembles

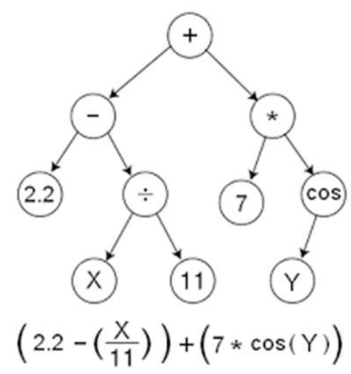


Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Auto-sklearn: Efficient and Robust Automated Machine Learning, pages 113–134. Springer International Publishing, Cham, 2019.

TPOT: Tree-based Pipeline Optimization Tool

- Uses genetic programming to explore the space of programs (classifiers) that can be build starting from a set of primitives





Randal S. Olson, Jason H. Moore **TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning;** Proceedings of the Workshop on Automatic Machine Learning, PMLR 64:66-74, 2016.

Main developments from the second wave

- Surrogate models
- Bayesian optimization is now a “standard”
- Multi fidelity approaches
- Resurgence of meta-learning
- …

Overview of AutoML techniques

First wave

Second wave

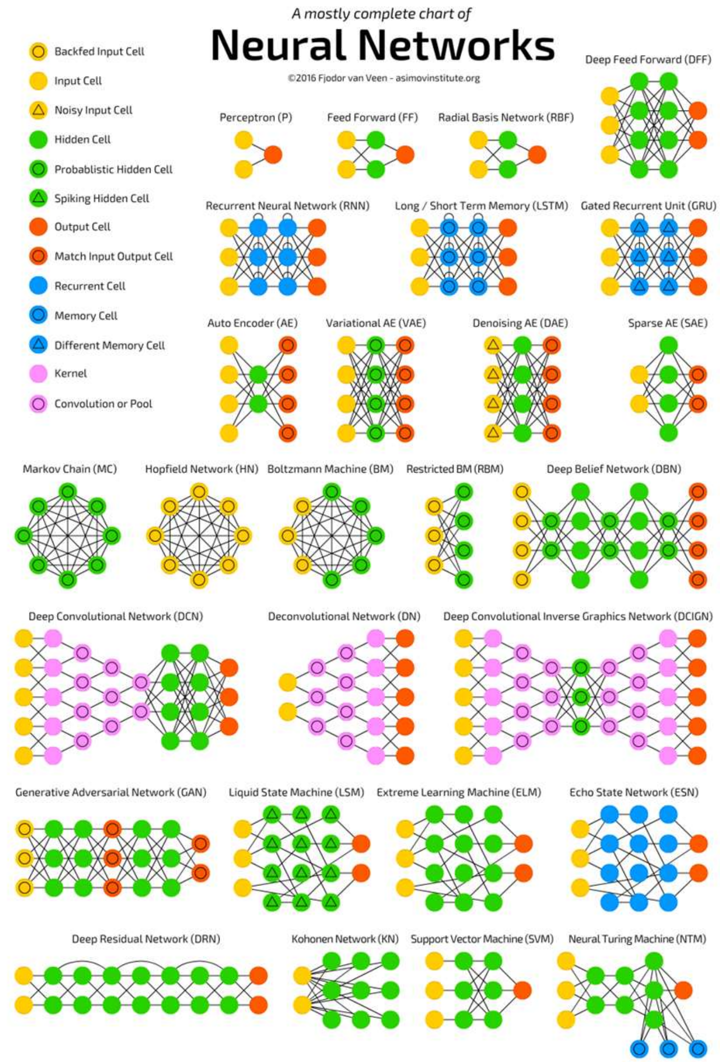
Third wave

Year	Ref.	Method	Type	Description	Innovative aspects
2006	[6, 7]	PSMS	β	Vectorial representation of solutions, PSO used as optimizer, subsampling, CV	Formulation of the full model selection task
2007	[13, 12]	Heterogeneous surrogate evolution	β	Parallel co evolution of models, ensemble generation	Returned ensemble of solutions, large and heterogenous space of models
2010	[9]	Ensemble PSMS	β	Enhanced PSMS with ensemble of solutions	Returned an ensemble of solutions as output
2012	[44]	GPS: GA-PSO-FMS	β	GAs were used to search for a model template, PSO was used for hyperparameter optimization	Separation of template search and hyperparameter optimization
2013	[45]	Auto-WEKA	γ	SMBO with SMAC, approached the CASH problem	Definition of the combined algorithm selection and hyperparameter optimization problem
2014	[41]	Multi-objective surrogate-based FMS	γ	Multi objective (complexity/performance) evolutionary method, surrogates were used to approximate the fitness function	Among the first methods using a meta-learner for AutoML, multi-objective formulation
2015	[11]	AutoSKLearn	γ	SMBO, warm starting with a classifier, ensemble generation	AutoML definition, warm-starting with meta-learner, winner of AutoML challenge
2016	[37, 38]	TPOT	β	Genetic programming / NSGA-II selwection, cross validation, data sampling	Models naturally codified as GP trees
2019	[54, 28]	NAS /AutoKeras	γ	SMBO for Neural Architecture search	Kernel function for comparison of architectures

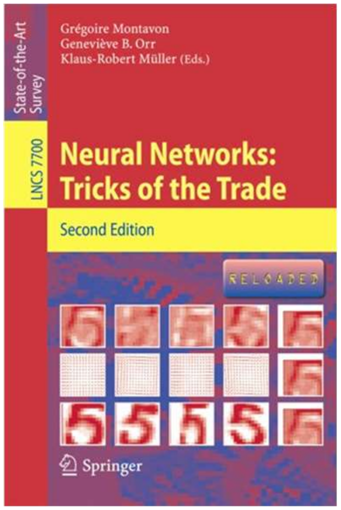
(What about deep learning?)

- The success of DL models largerly depends on the desing choices made by developers:
  - How many layers?
  - What type of units?
  - For CNNs, what kernel size? how many filters / feature maps per layer, pooling strategy? Etc.
  - Regularization strategy? Activation functions?
  - Optimizer, learning algorithm?
  - Etc.

Similar problem!



Neural architecture search



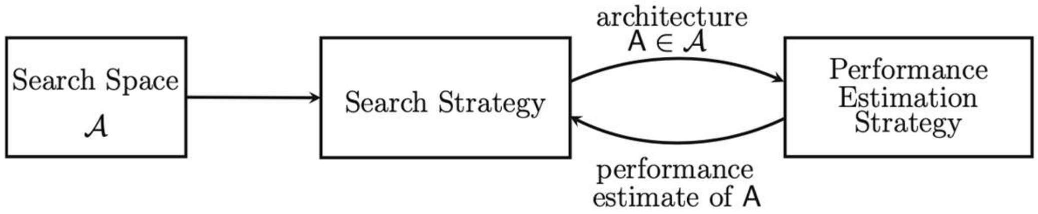
- It is well known that the success of deep learning (DL) solutions relies heavily on the design choices and hyperparameters

- DL is everywhere, hence, AutoML methods targeting DL can have a huge impact

Neural architecture search

- NAS is the process of automating architecture design in the context of DL models (share same complications as AutoML in supervised learning)

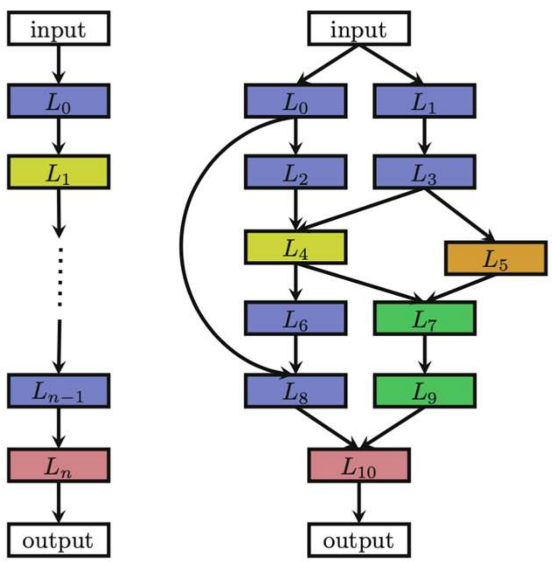
- Components of a NAS method:
  - Search space
  - Search strategy
  - Performance estimation strategy

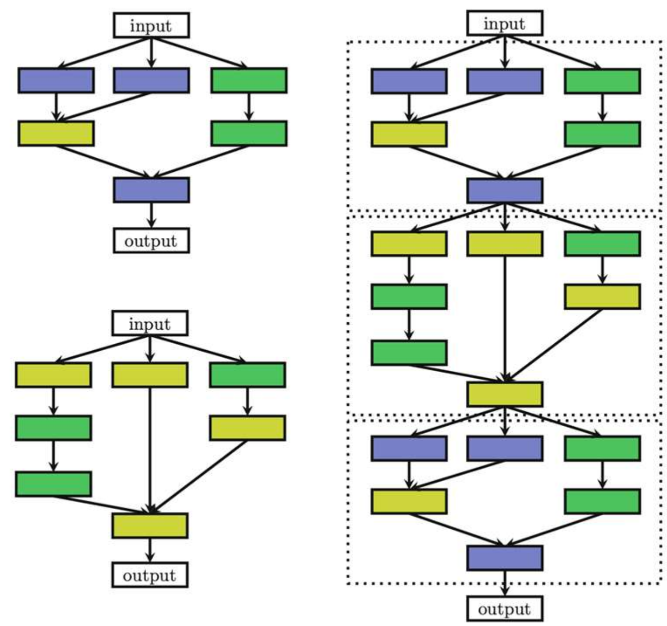


T. Elsken et al.. (2019) **Neural architecture search**. In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham

Search space of NAS

- The search space determines the architectures that can be discovered with NAS





Neural architecture search

- Search strategy:
  - Evolutionary algorithms
  - Bayesian optimization
  - Reinforcement learning
  - Gradient based

- Performance estimation:
  - Multi fidelity approaches
  - One shot architecture search

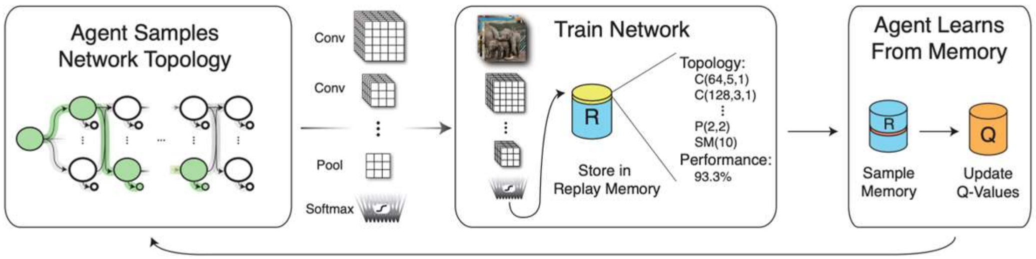
Neural architecture search

- Search strategy:
  - Evolutionary algorithms
  - Bayesian optimization
  - Reinforcement learning
  - Gradient based

- Performance estimation:
  - Multi fidelity approaches
  - One shot architecture search

Neural architecture search

- Search strategy:
  - Evolutionary algorithms
  - Bayesian optimization
  - Reinforcement learning
  - Gradient based

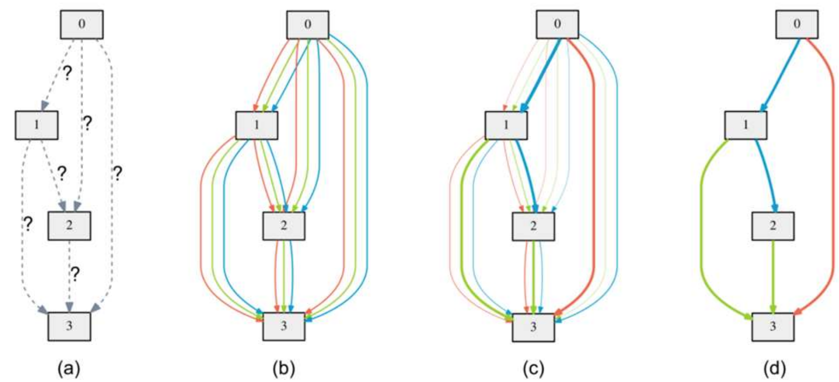


- Performance estimation:
  - Multi fidelity approaches
  - One shot architecture search

Bowen Baker, Otkrist Gupta, Nikhil Naik, Ramesh Raskar, **Designing Neural Network Architectures using Reinforcement Learning**
https://arxiv.org/pdf/1611.02167.pdf

Neural architecture search

- Search strategy:
  - Evolutionary algorithms
  - Bayesian optimization
  - Reinforcement learning
  - Gradient based



DARTS

- Performance estimation:
  - Multi fidelity approaches
  - One shot architecture search

Hanxiao Liu et al. DARTS: DIFFERENTIABLE ARCHITECTURE SEARCH. ICLR 2019

Latest developments

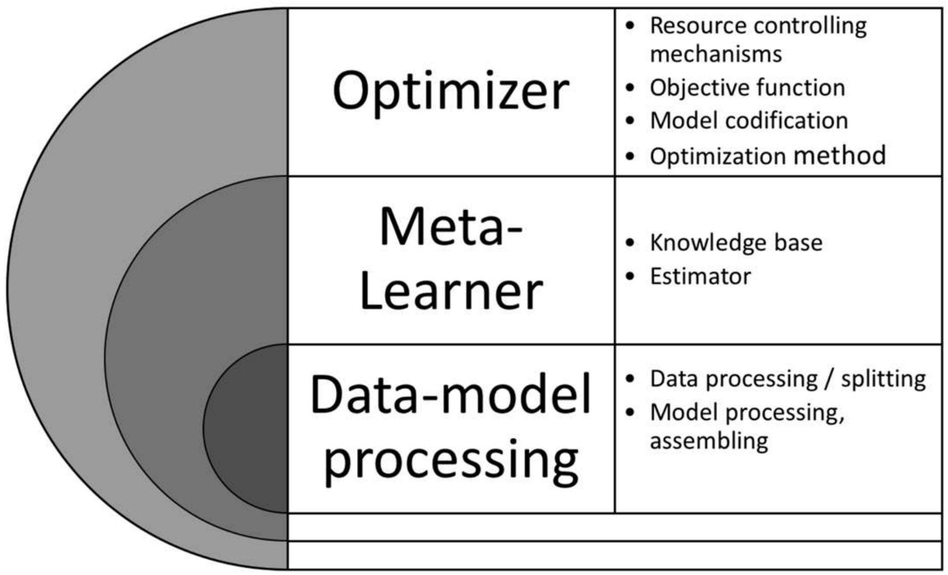
- Neural architecture search is driving new ways for efficient AutoML
- RL-based AutoML
- Meta learning resurgence
- Complex search spaces that require of new optimization methodologies

Recent work on AutoML

- From the commercial side:
  - https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml
  - http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

Elements of AutoML methodologies

- IMHO: one can distinguish three main components, namely: Optimizer, Meta-learner and data-model processing methods.



Hugo Jair Escalante. **Automated Machine Learning - a brief review at the end of the early years.** *Forthcoming, 2020*

Elements of AutoML methodologies

- IMHO: one can distinguish three main components, namely: Optimizer, Meta-learner and data-model processing methods.
  - **Optimizer:** score of the AutoML method, comprises the optimization algorithm itself, together with the objective function
  - **Meta-learner:** refers to any estimator that is used during the AutoML optimization process
  - **Data processing mechanisms** modify, organize data according to the need of AutoML methods.

Hugo Jair Escalante. **Automated Machine Learning - a brief review at the end of the early years.** *Forthcoming, 2020*

AutoML / Full model selection

- Pros
  - The job of the data analyst is considerably reduced
  - Neither knowledge on the application domain nor on machine learning is required
  - Different methods for preprocessing, feature selection and classification are considered
  - It can be used in any classification problem

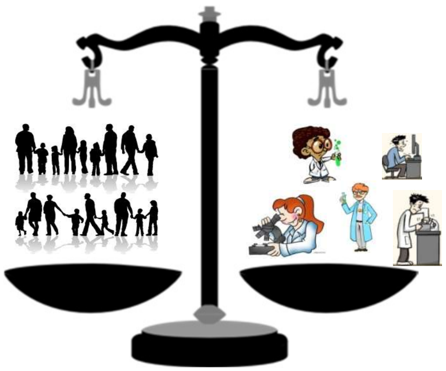
- Cons
  - It is real function + combinatory optimization problem
  - Computationally expensive
  - Risk of overfitting

NLF theorem!

… and full models for all

- **Short-term goal:** to provide data analysts with tool that allows them to build effective classification systems without much effort

- **Long-term goal:** An APP that allows anyone to build a classification model from their data (photographs, smart phone data, tweets, etc.)



Challenges and research opportunities

- Explainable AutoML models.
- AutoML in feature engineering.
- AutoML for non tabular data. Re
- Large scale AutoML.
- Transfer learning in AutoML.
- Benchmarking and reproducibility in AutoML.
- Interactive AutoML methods.

Questions?