

UCB - CS189
Introduction to Machine Learning
Fall 2015

Lecture 10: Model search

Isabelle Guyon
ChaLearn

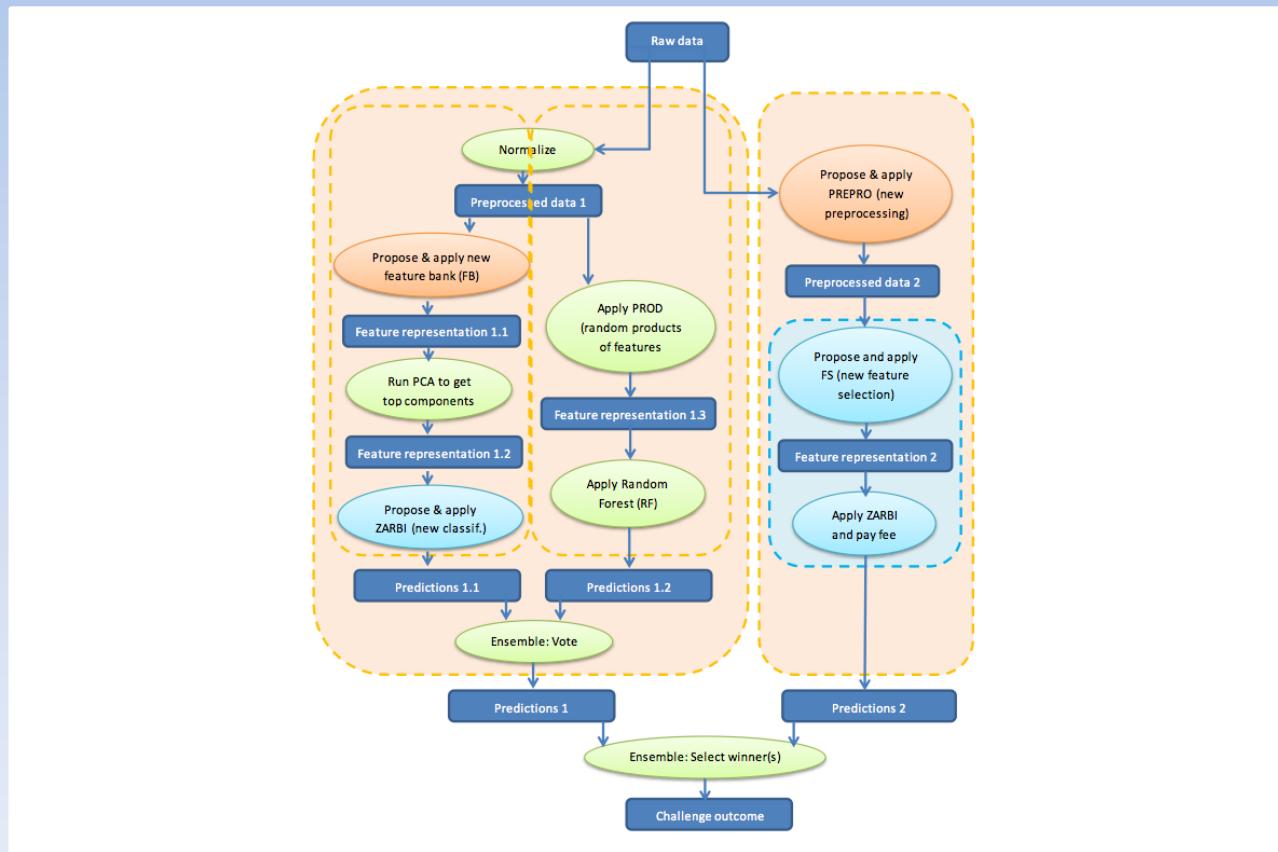
Come to my office hours...
Wed 2:30-4:30 Soda 329

Last time

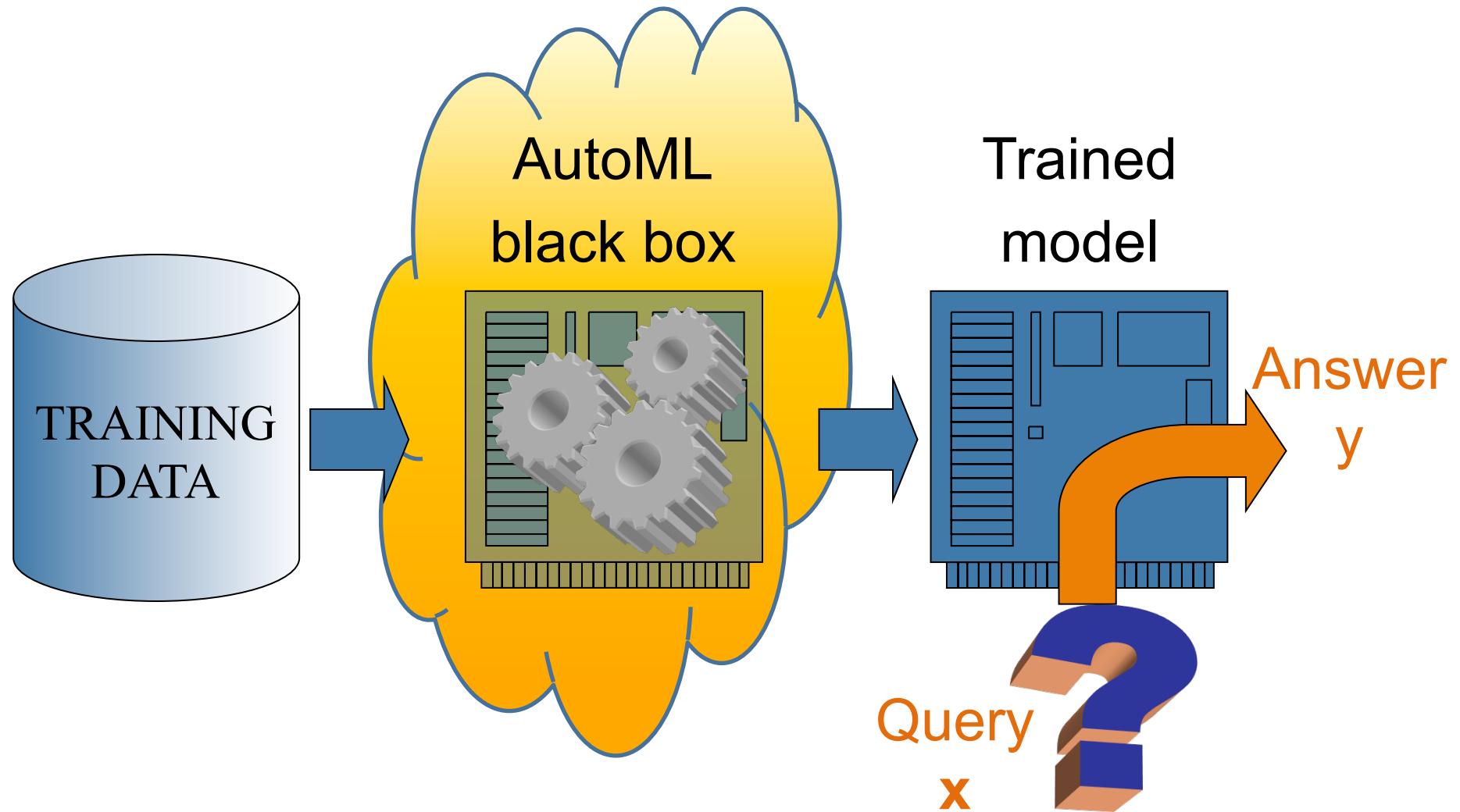


Come to my office hours...
Wed 2:30-4:30 Soda 329

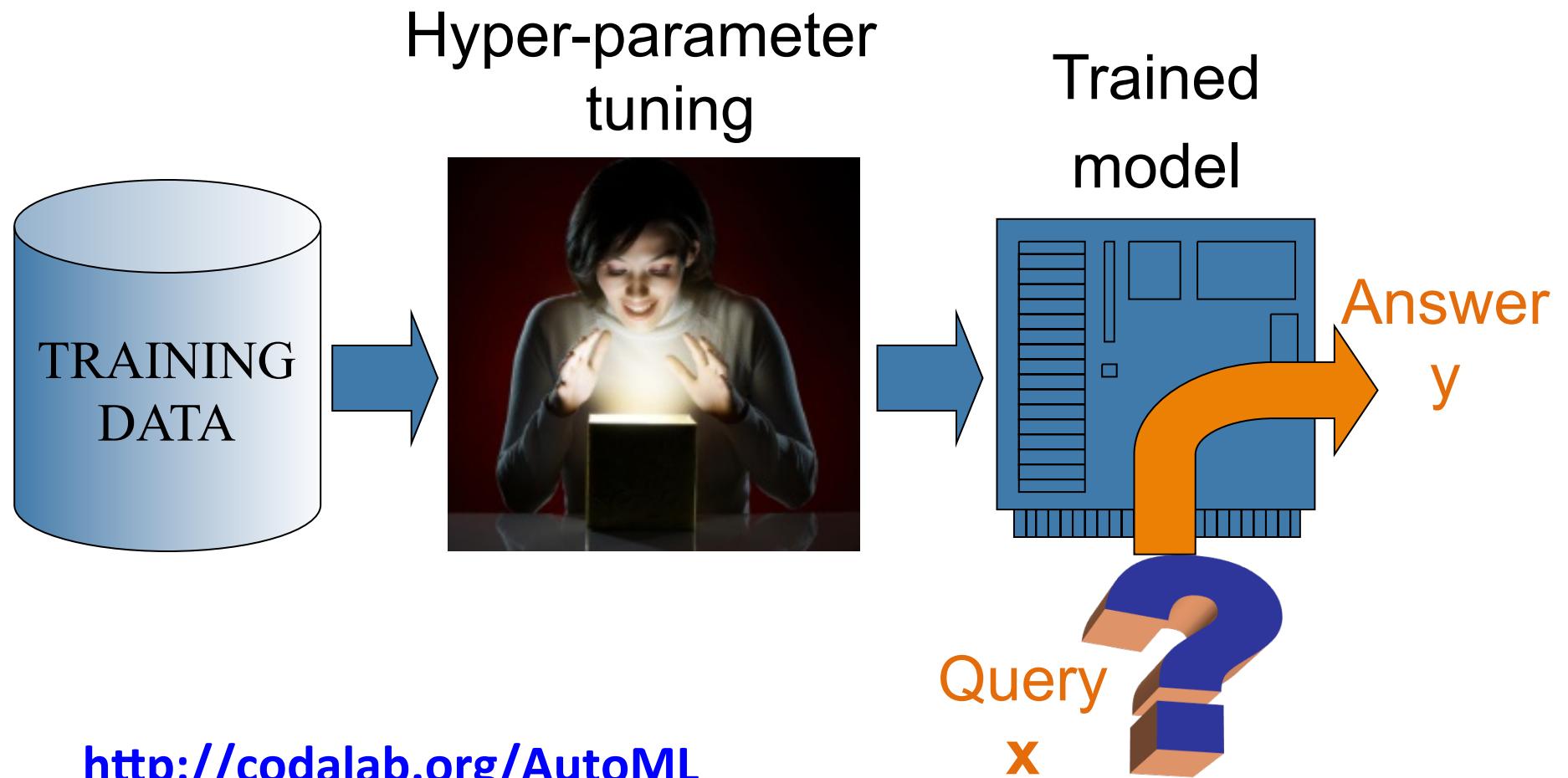
Today: model search



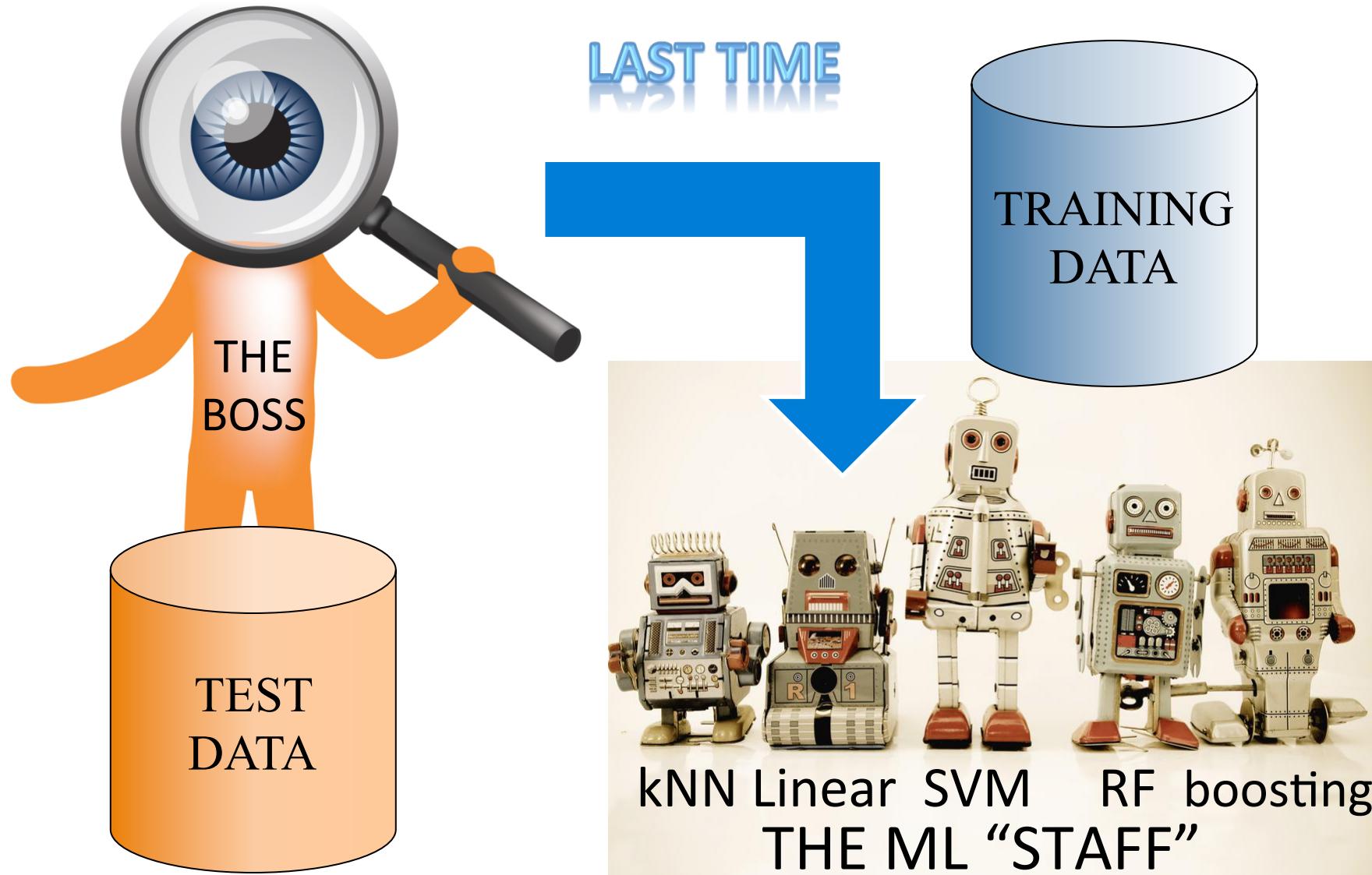
Remember: The DREAM



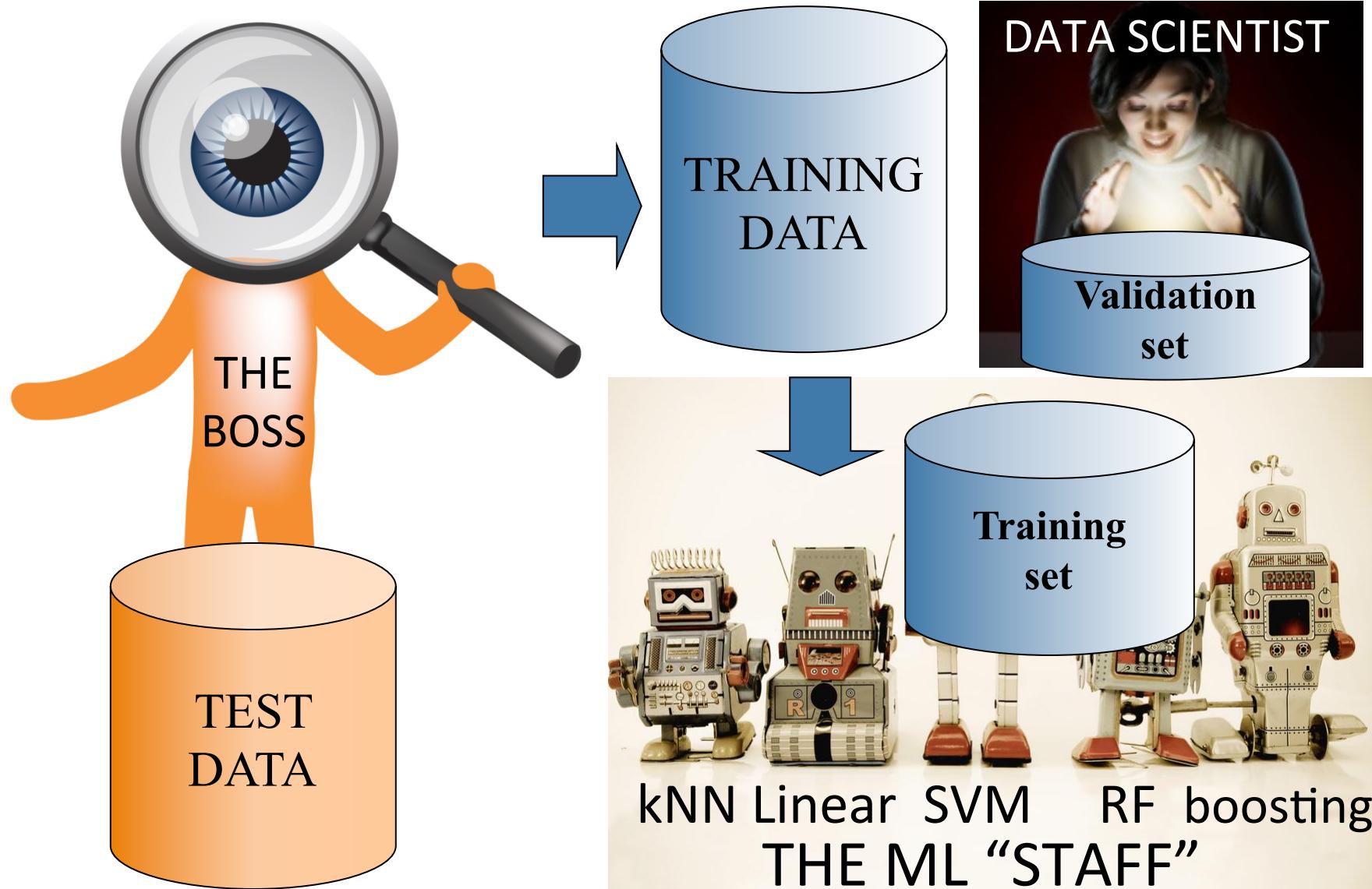
Remember: The REALITY



Changing roles again!



Changing roles again!



DATA SCIENTIST

THE SEXIEST JOB OF THE 21ST CENTURY



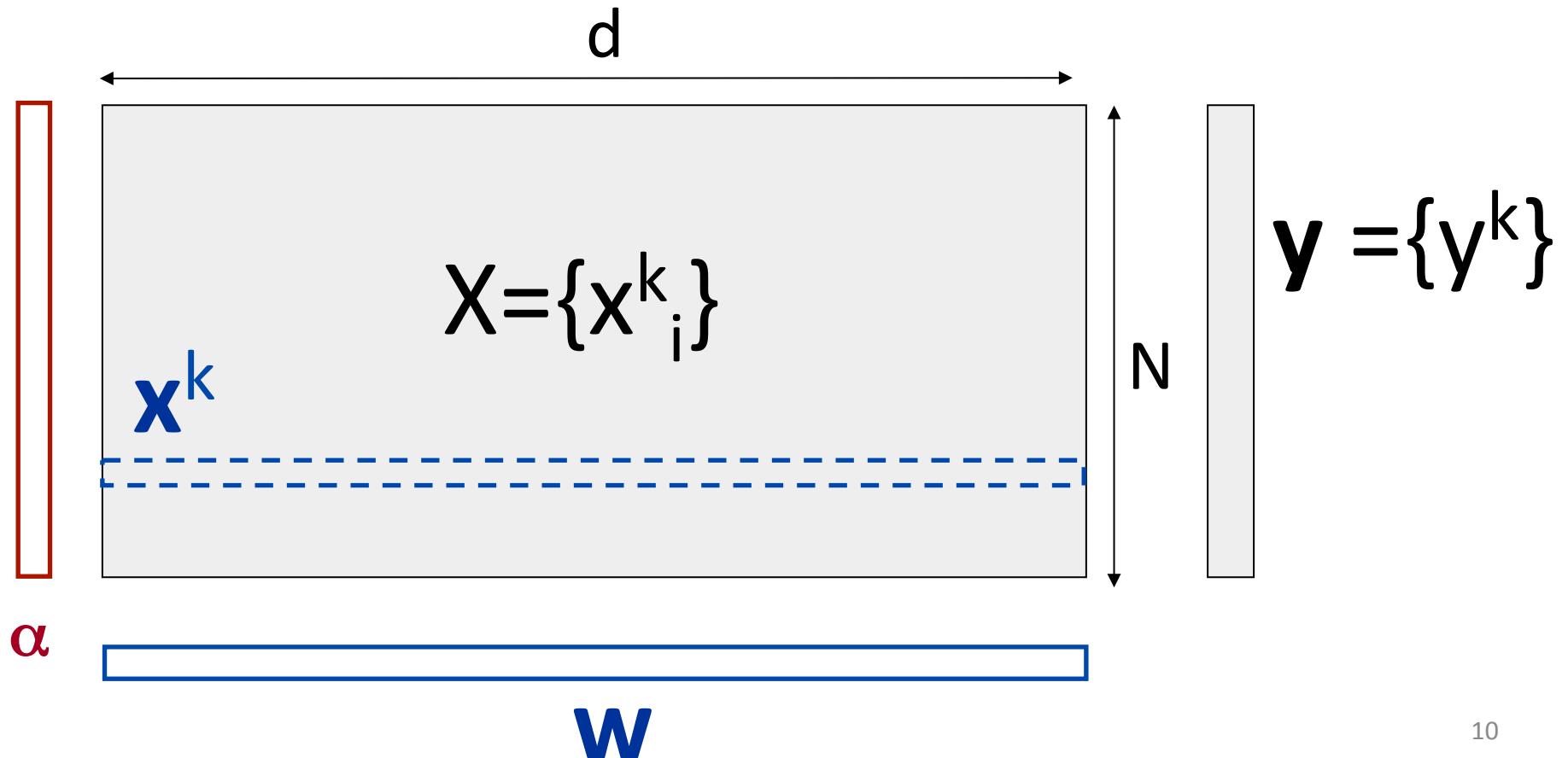
DATA SCIENTIST



- You are tuning the hyperparameters θ .
- Your “training data” is the validation set.

Produce ~~4~~ models:

- 1) "Hebb's rule" linear model $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \sum_k y_k \mathbf{x}^k \cdot \mathbf{x}$
- 2) Regularized linear model $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \sum_k \alpha_k \mathbf{x}^k \cdot \mathbf{x}$
- 3) Parzen windows $f(\mathbf{x}) = \sum_k y_k k(\mathbf{x}^k, \mathbf{x})$
- 4) Non-linear model $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) = \sum_k \alpha_k k(\mathbf{x}^k, \mathbf{x})$

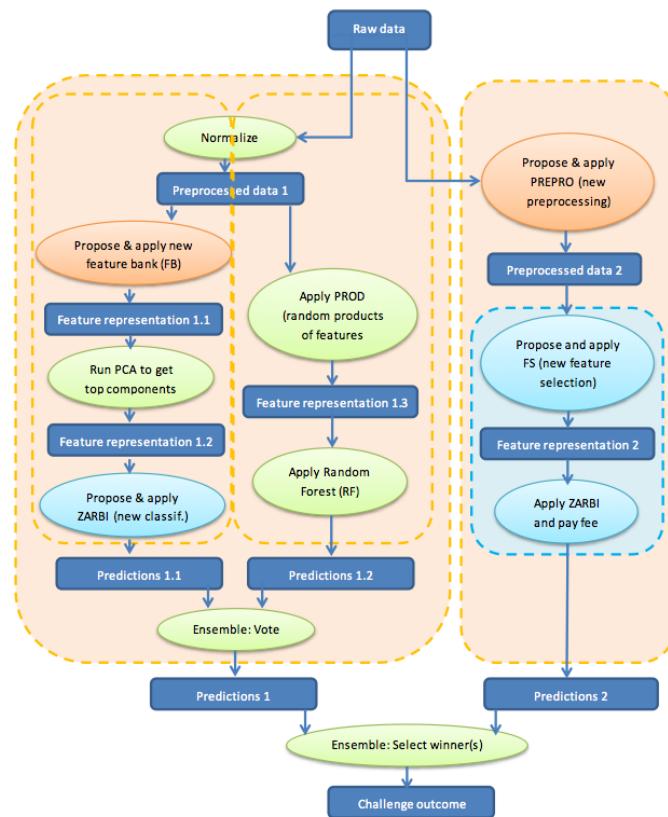


What are hyper-parameters

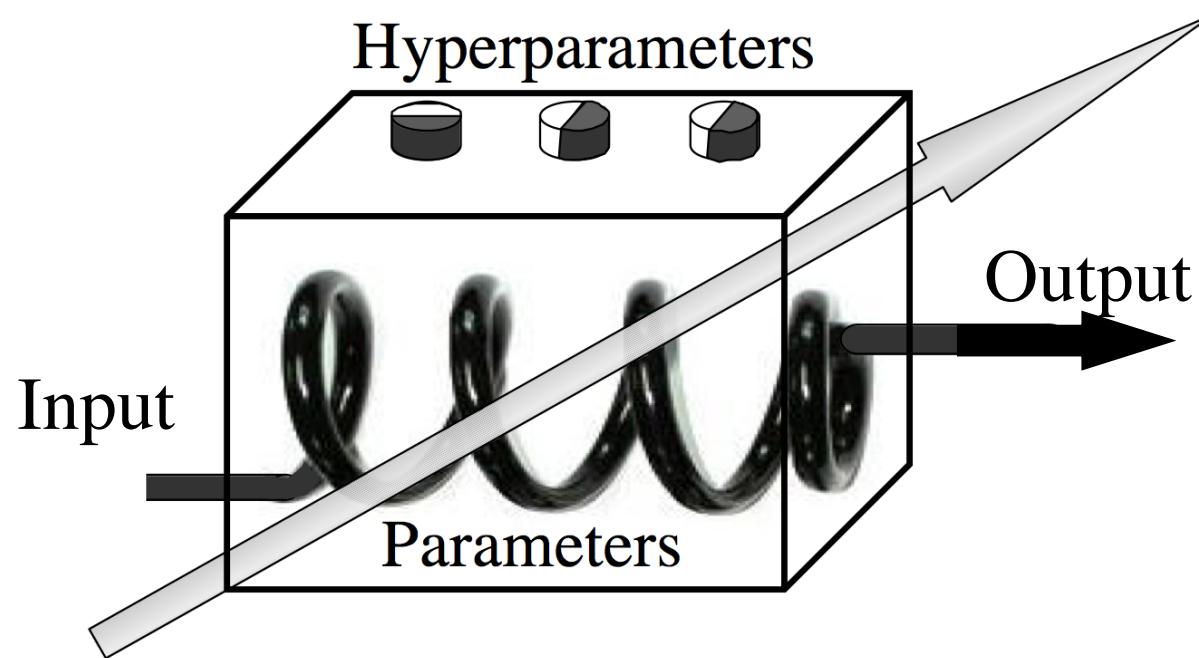
- **Preprocessing** (feature selection, feature construction, feature or pattern normalizations, etc.) [~ 3 HP]
- **Model** (linear, kernel, kNN, decision trees, etc.) [1 HP]
- Model **hyper-parameters** (number of neighbors in kNN, kernel choice polynomial or Gaussian, kernel parameters q , σ , number of layers and hidden units in neural nets, tree depth in decision trees, etc.) [~ 2 HP]
- **Loss function** (hinge, logistic, square-loss, etc.) [1 HP]
- **Regularizer** (2-norm, 1-norm, regularization parameter λ) [2 HP]
- **Learning rate** η for gradient descent. [1 HP]
- Etc. [Total ~ d=10 HP]

Hyper-models

You can create workflows (directed acyclic graphs) of all the modules available in your machine learning toolkit!



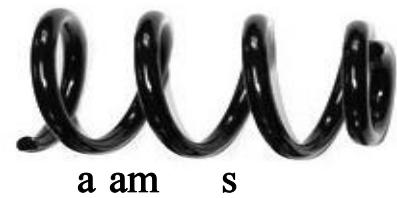
Model selection / hyper-parameter search



Model selection / hyper-parameter search

yp pa am s
—

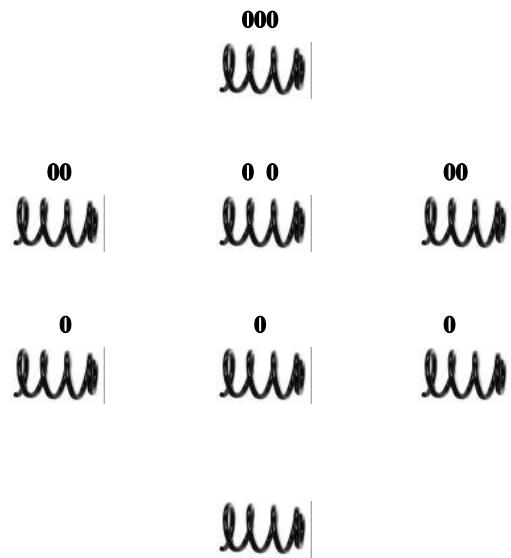
a m c . ;) a m . ;)



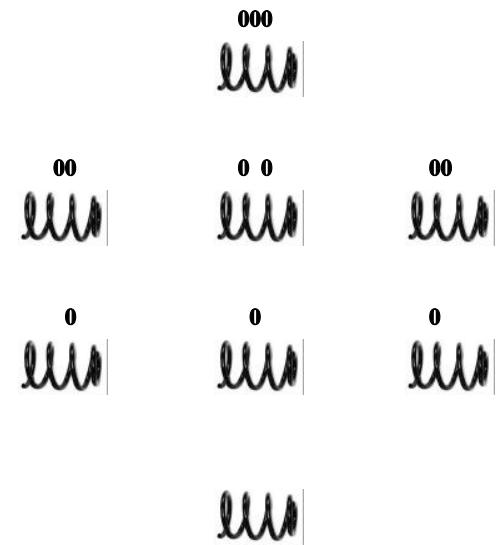
$$f^{**} = \operatorname{argmin}_{\theta} R_2[f^*, D], \text{ such that } f^* = \operatorname{argmin}_{\alpha} R_1[f, D]$$

Search strategies

Filters



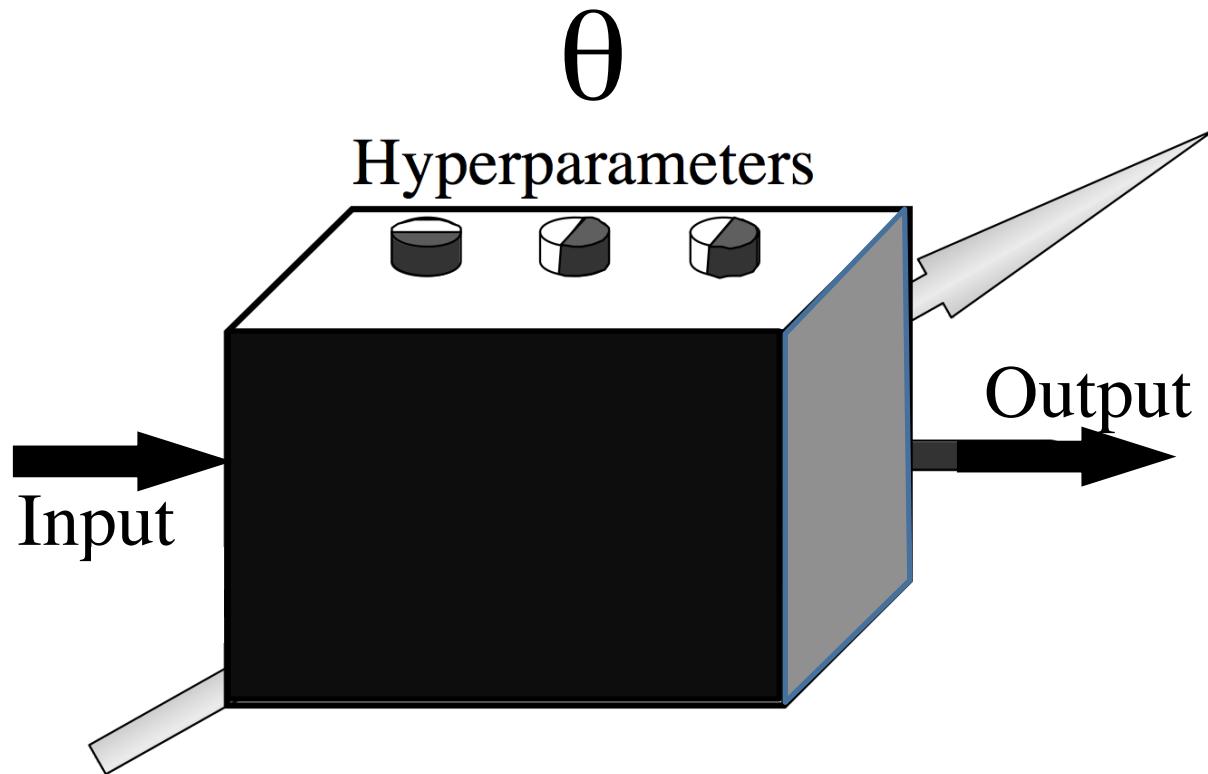
Wrappers



*Embedded
methods*

Wrappers:

your learning machine is a black box



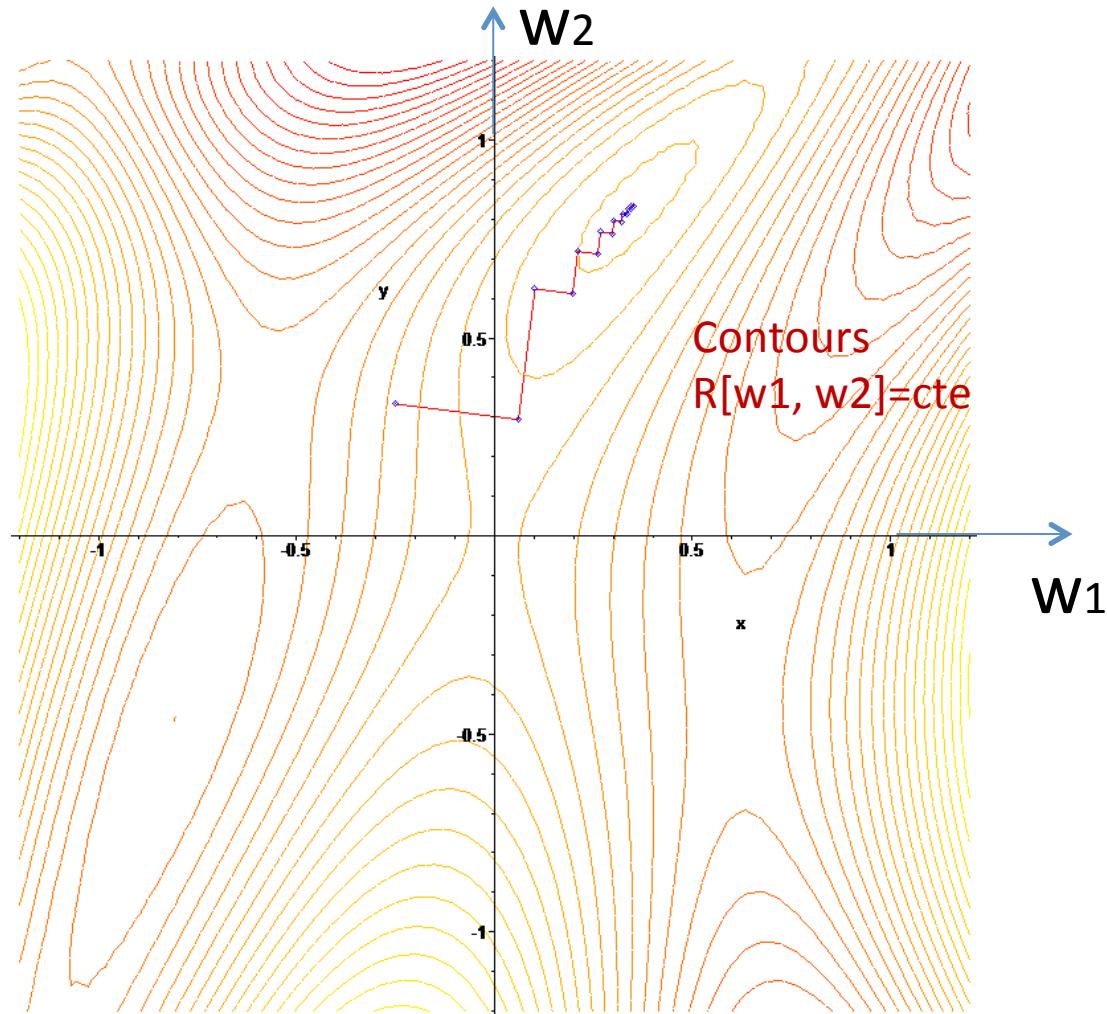
You cannot compute $\nabla_{\theta} R$ or $\partial R / \partial \theta_i$
No gradient descent!

What the learning machines do:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} R$$

η too small: many steps needed to converge.

η too large: zigzags.



Picture from Wikipedia



Grid search is best
Prof. G. Cawley

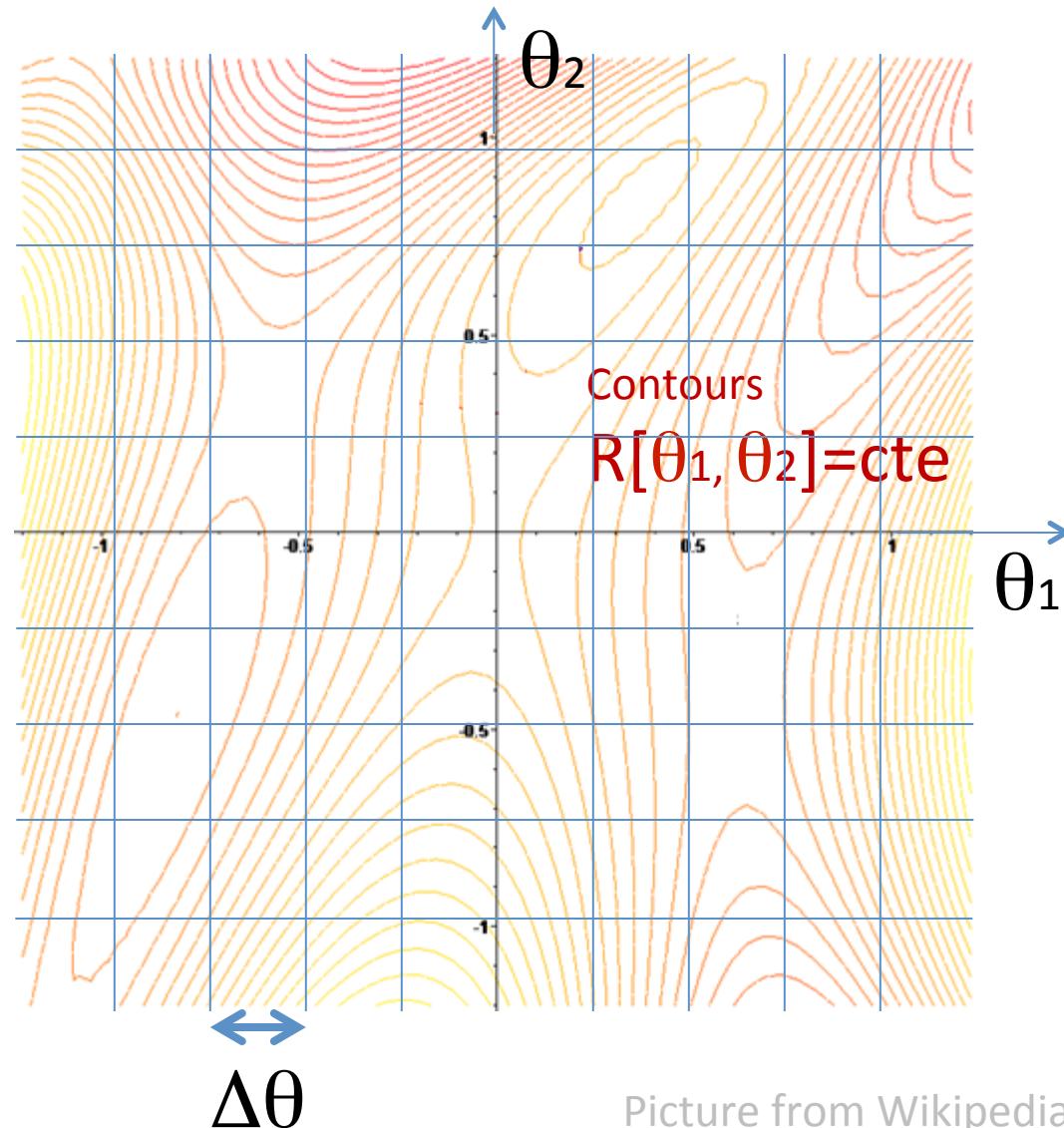
Grid search:

Evaluate $R[\theta]$
on grid nodes only to
find $\min_{\theta} R$.

$\Delta\theta$ small: too slow.

$\Delta\theta$ large: miss the
optimum.

What you have to do:



Picture from Wikipedia

Grid search

- **Advantages:**
 - Simple.
 - Does not fall into local minima.
- **Disadvantage:**
 - Scales poorly: for g grid points in each of the d dimensions, g^d evaluations of $R[\theta]$! Example: $d=10$, $g=10$. $10^{10} = 10,000,000$ evaluations. One evaluation means TRAINING ONE MODEL! (OR SEVERAL MODELS IN THE CASE OF K-FOLD!!)

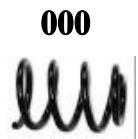
Your options

- 1. Reduce the number of hyper-parameters.**
- 2. Learn how to do “fancy” search.**



You chose:
Reduce HP Space

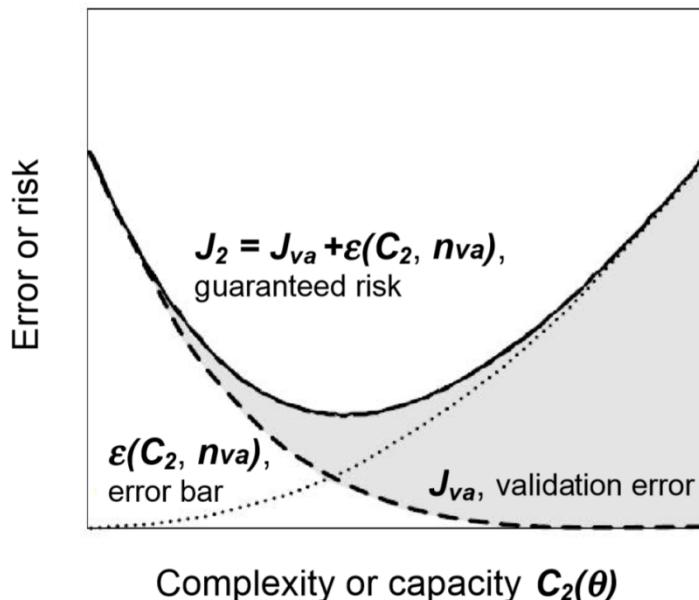
Filter methods



You chose:
Reduce HP Space

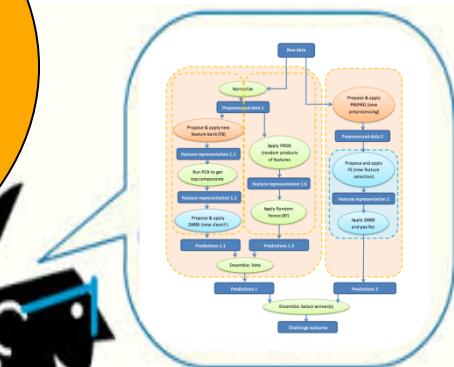
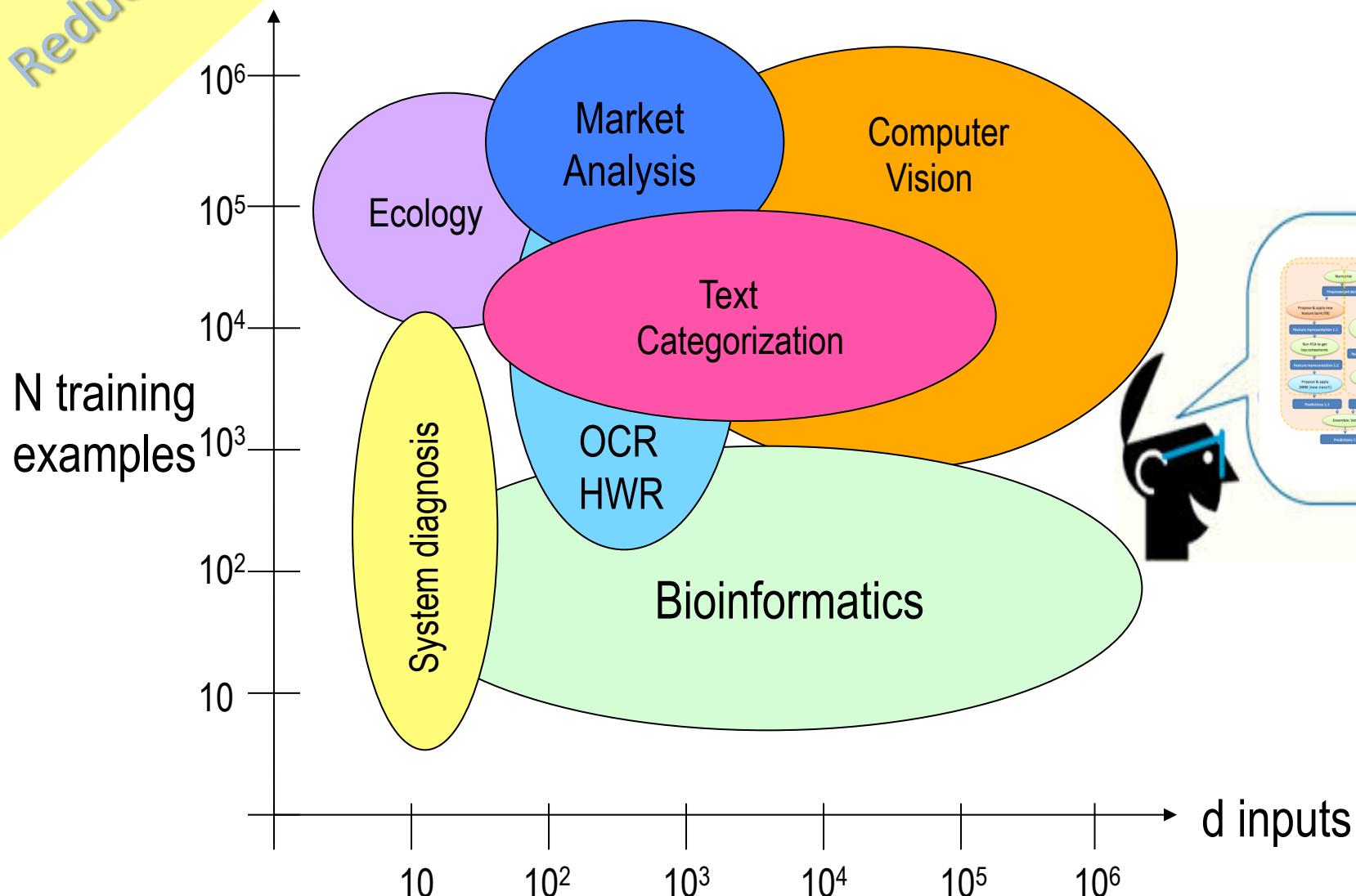
Good choice!

- Reducing HP space ALSO reduces the risk of overfitting!



You chose:
Reduce HP Space

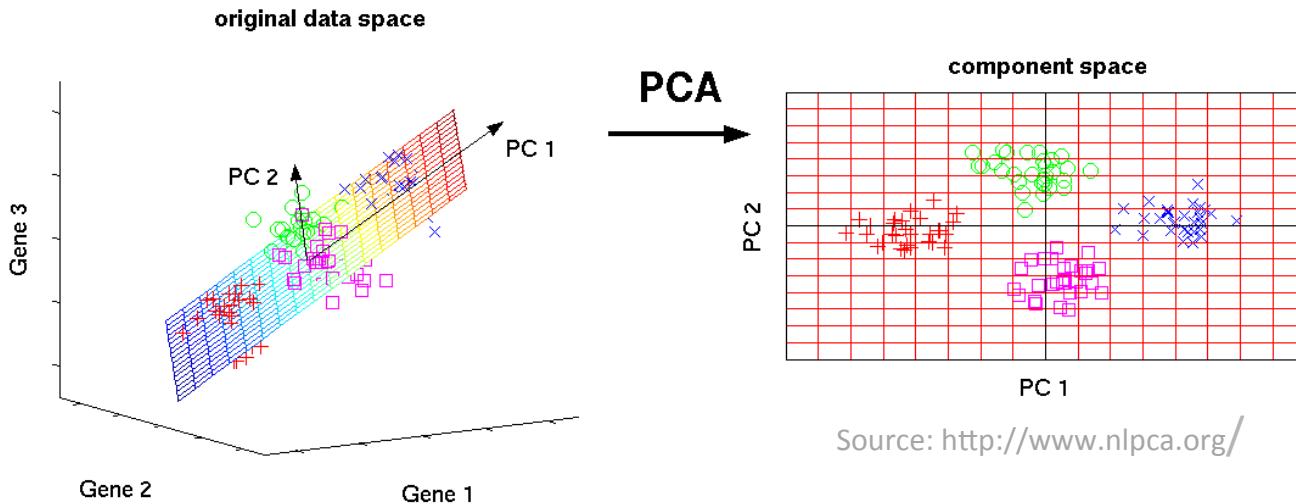
Prior knowledge



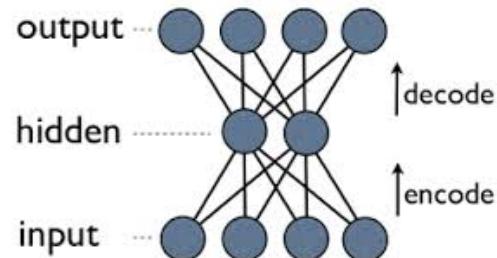
You choose:
Reduce HP Space

Unsupervised learning

1. Space dimensionality reduction with PCA (Principal Component Analysis).

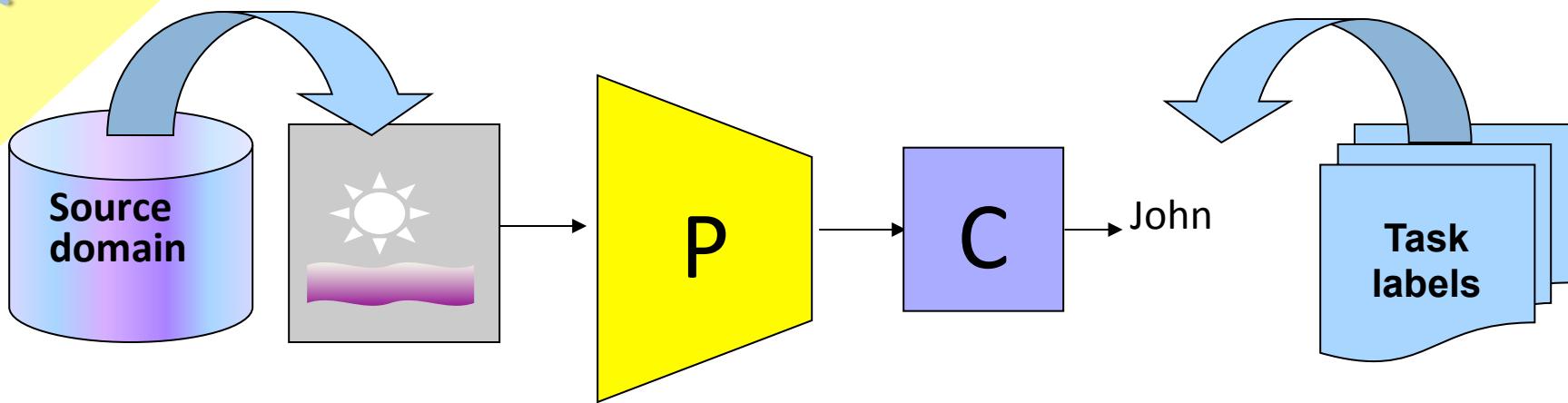


2. Unsupervised “pre-training” with (staked) auto-encoders.



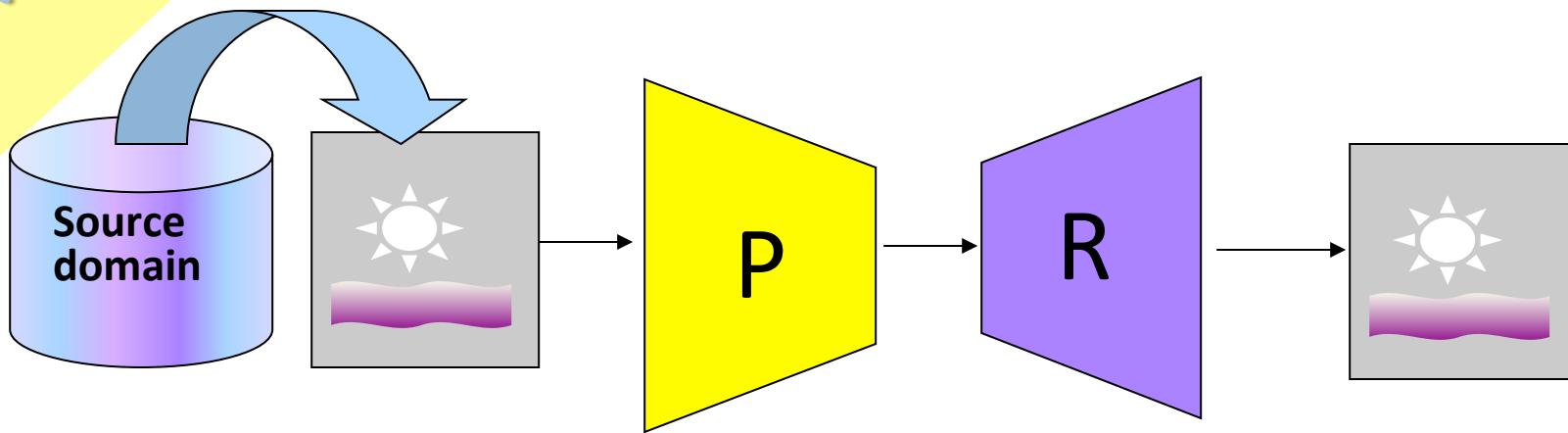
You chose:
Reduce HP Space

Transfer learning



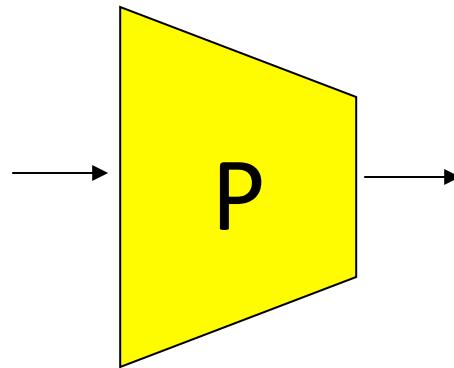
You chose:
Reduce HP Space

Transfer learning



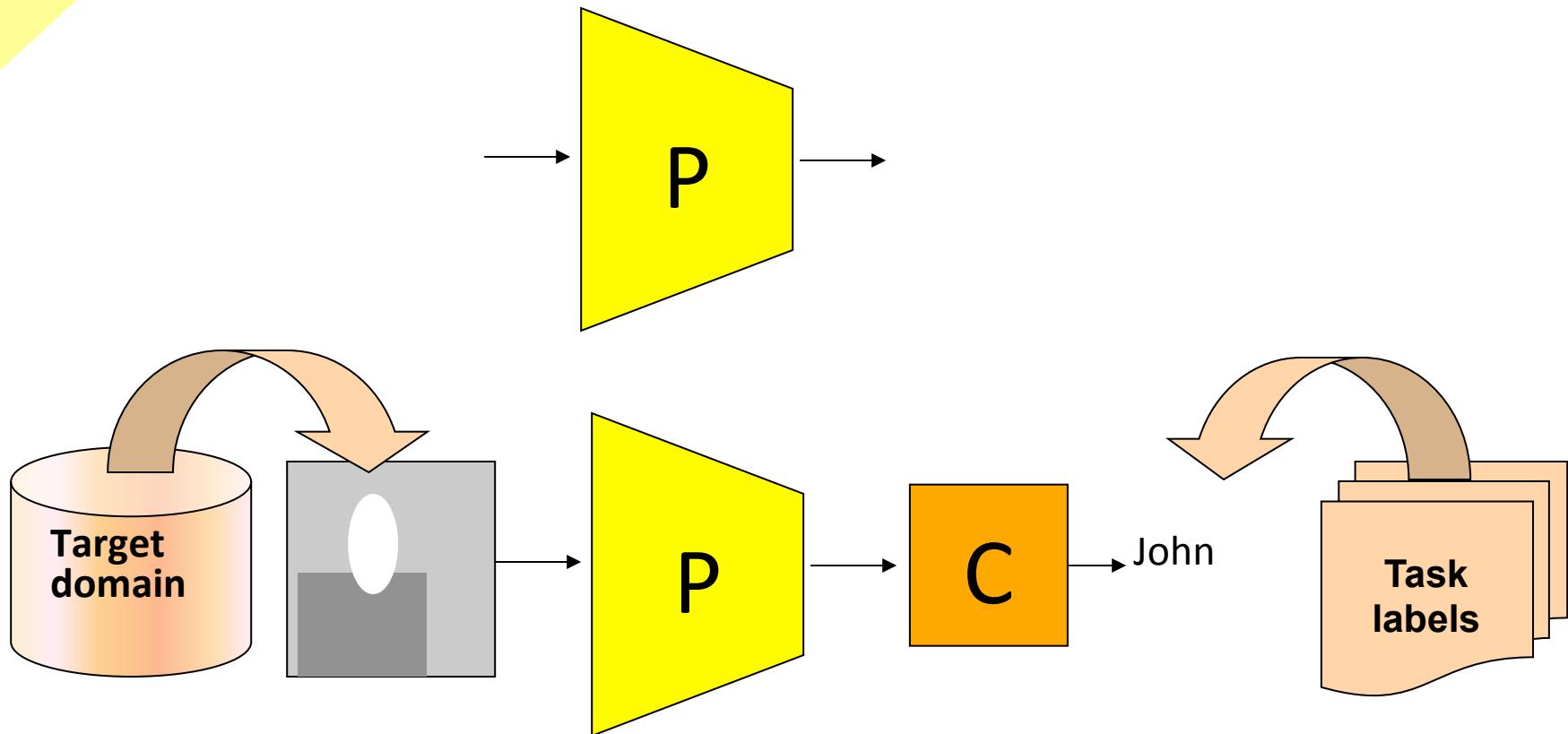
You chose:
Reduce HP Space

Transfer learning



You chose:
Reduce HP Space

Transfer learning



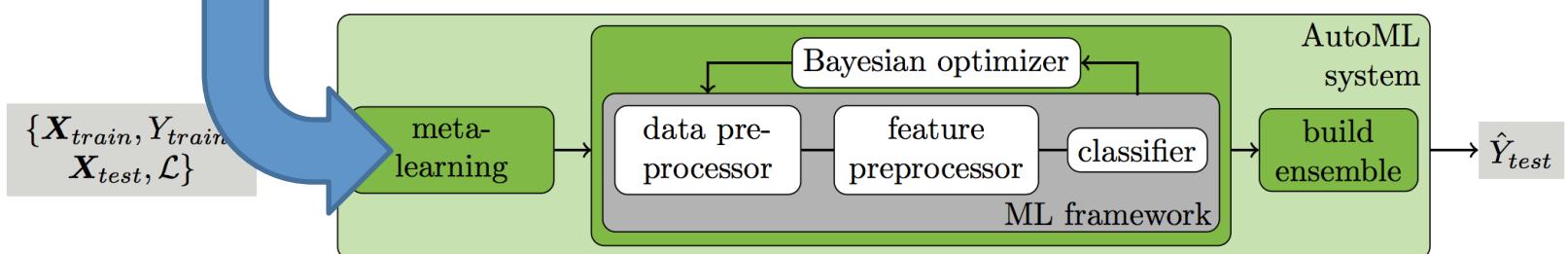
<http://www.causality.inf.ethz.ch/unsupervised-learning.php>

You chose:
Reduce HP Space

Meta learning

The screenshot shows the UCI Machine Learning Repository homepage. At the top, there's a logo of a hand-drawn antelope and the text "Machine Learning Repository" and "Center for Machine Learning and Intelligent Systems". Below this, a search bar and navigation links for "About", "Citation Policy", "Donate a Data Set", and "Contact". A large blue button labeled "View ALL Data Sets" is prominent. On the left, a sidebar titled "Browse Through: 333 Data Sets" provides filters for various categories: Default Task (Classification 241, Regression 55, Clustering 46, Other 50), Attribute Type (Categorical 37, Numerical 189, Mixed 56), Data Type (Multivariate 256, Univariate 16, Sequential 31, Time-Series 34, Text 30, Domain-Theory 21, Other 21), Area (Life Sciences 82, Physical Sciences 42, CS / Engineering 93, Social Sciences 22, Business 17, Game 9, Other 65), # Attributes (Less than 10 79, 10 to 100 150, Greater than 100 53), # Instances (Less than 100 15, 100 to 1000 124, Greater than 1000 163), and Format Type (Matrix 231, Non-Matrix 102). The main area displays a table of 10 datasets with columns: Name, Data Types, Default Task, Attribute Types, # Instances, # Attributes, and Year. The first dataset listed is Abalone.

Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995
Adult	Multivariate	Classification	Categorical, Integer	48842	14	1996
UCI Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38	
UCI Anonymous Microsoft Web Data		Recommender-Systems	Categorical	37711	294	1998
Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998
Artificial Characters	Multivariate	Classification	Categorical, Integer, Real	6000	7	1992
Audiology (Original)	Multivariate	Classification	Categorical	226		1987
Audiology (Standardized)	Multivariate	Classification	Categorical	226	69	1992
Auto MPG	Multivariate	Regression	Categorical, Real	398	8	1993
Automobile	Multivariate	Regression	Categorical, Integer, Real	205	26	1987



Methods for Improving Bayesian Optimization for AutoML
 Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias, Manuel Blum, Frank Hutter, ICML workshop on AutoML, 2015.

You choose:
Reduce HP Space

Pre-selection with a surrogate learning machine

- Take a simplistic learning machine (fast to train).
- Select a bunch of HP with this LM.
- Example: univariate filter methods of feature selection.

“Fancy” search (a.k.a. heuristic search)

- Intensive search:
 - Simulated annealing of MCMC.
 - Genetic algorithms.
 - Particle swarm optimization.
- Greedy search:
 - Nelder-Mead/simplex.
 - Pattern search.
 - Nested subsets.

You chose:
Fancy search

Intensive search

Example: simulated annealing

Random walk on a search graph:

$$\theta = \theta_0, T = T_0$$

for $k = 0$ to k_{\max} :

$T \leftarrow \text{temperature}(k)$ annealing schedule

$\theta_{\text{new}} \leftarrow \text{random_neighbor}(\theta)$ transition probability

if $P(R[\theta], R[\theta_{\text{new}}], T) \geq \text{random}(0, 1)$:

$\theta \leftarrow \theta_{\text{new}}$

return θ

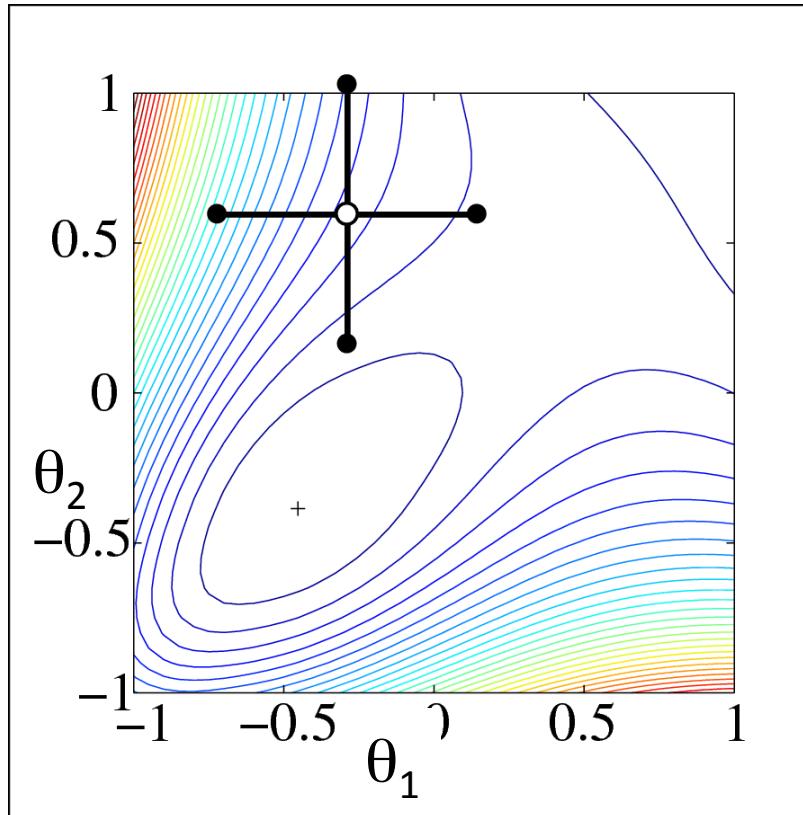
acceptance probability:

$P(E, E', T) = 1$ if $E' < E$ and $\exp(-(E' - E)/T)$ otherwise

You chose:
Fancy search

Greedy search

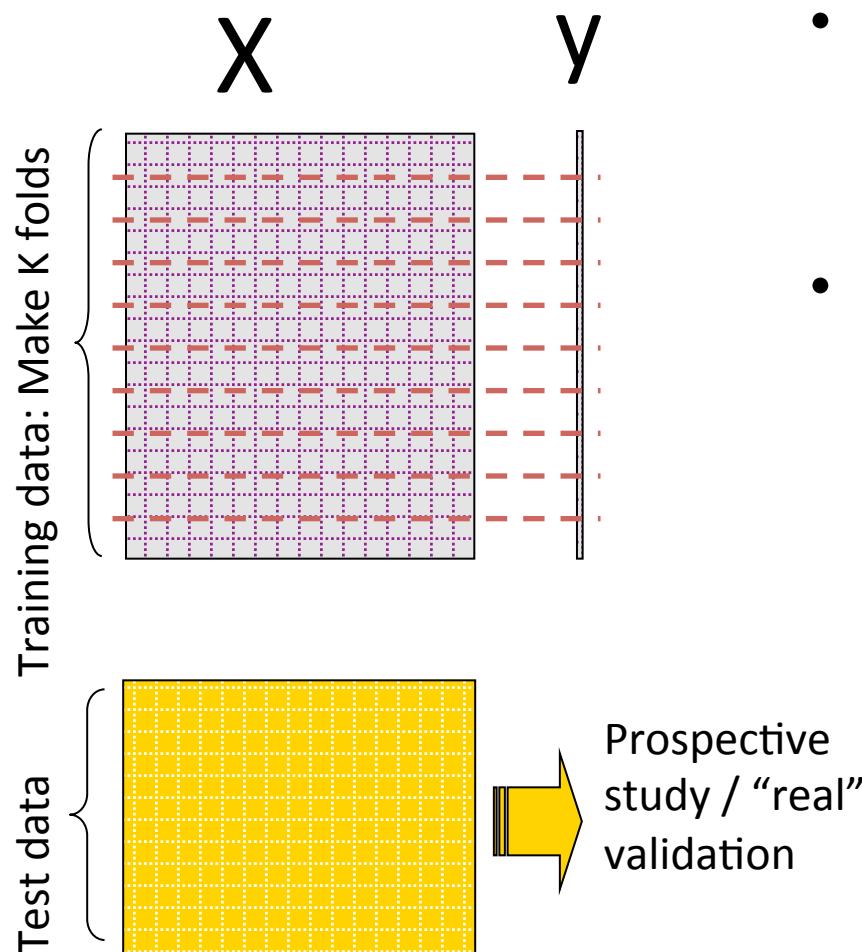
Example: pattern search



- Choose a direction i
- Vary θ_i by steps of the same magnitude (positive or negative) until a minimum of $R[\theta]$ is reached
- Half the step size and choose another θ_i

[https://en.wikipedia.org/wiki/Pattern_search_\(optimization\)](https://en.wikipedia.org/wiki/Pattern_search_(optimization))

Cross-validation

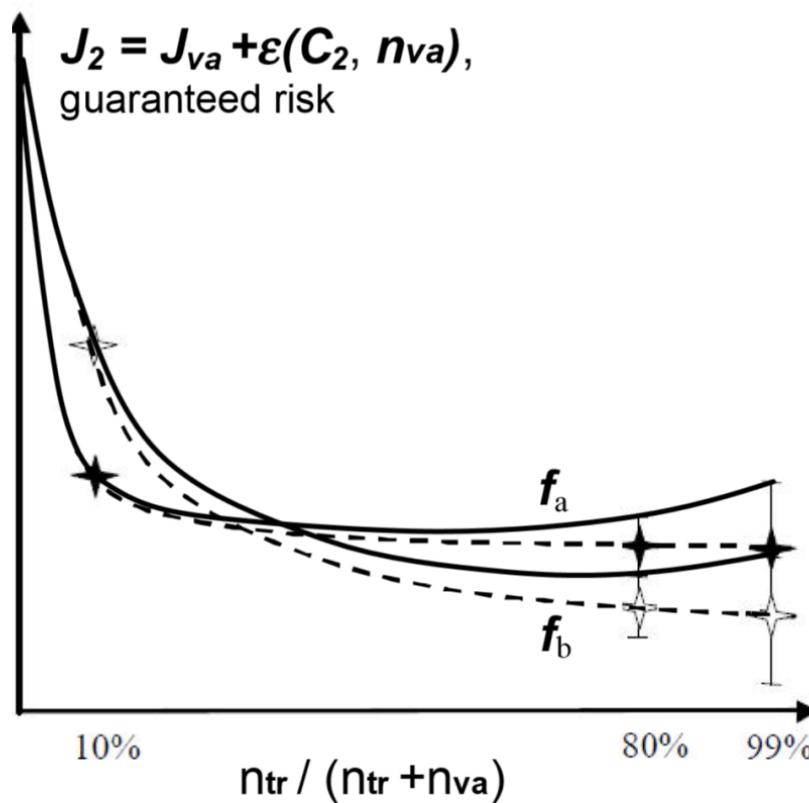


- **Learning = adjusting:**
parameters (w vector).
hyper-parameters (γ, ν, q, σ).
- ***Cross-validation with K-folds:***

For various values of γ, ν, q, σ :

- Adjust w on a fraction $(K-1)/K$ of training examples e.g. $9/10^{\text{th}}$.
- Test on $1/K$ remaining examples e.g. $1/10^{\text{th}}$.
- Rotate examples and average test results (CV error).
- Select γ, ν, q, σ to minimize CV error.
- Re-compute w on **all** training examples using optimal γ, ν, q, σ .

What should be K in K-fold?



K=10 is commonly used. Repeat k times if possible.

Why choose?

- Ensembles of models can be made by voting over all the models trained during the search.
- You may want to use the CV error as voting weight, similarly to a “Bayesian” method:
$$E(Y|X; D) \sim \sum_{\text{model}} E(Y|X; \text{model}) P(\text{model}|D)$$

$$F(x) = \sum_f f(x) \exp(-R_{cv}[f])$$

Summary

- Hyper-parameter/model selection has 2 ingredients:
 - Searching for the best solution (an optimization problem)
 - Evaluating the solution (a statistics problem)
- Searching must be done with heuristic search:
 - Exhaustive search or grid search.
 - Intensive search (simulated annealing).
 - Greedy search (pattern search).
- Evaluation of the solution can be done by:
 - k times 10-fold cross-validation (e.g. 10 times 10-fold).
 - Bootstrap.
- Robustness is gained by reducing HP space with filter methods.

Come to my office hours...
Wed 2:30-4:30 Soda 329

Next time

