UCB - CS189
Introduction to Machine Learning
Fall 2015
Lecture 10: Model search

Isabelle Guyon

ChaLearn

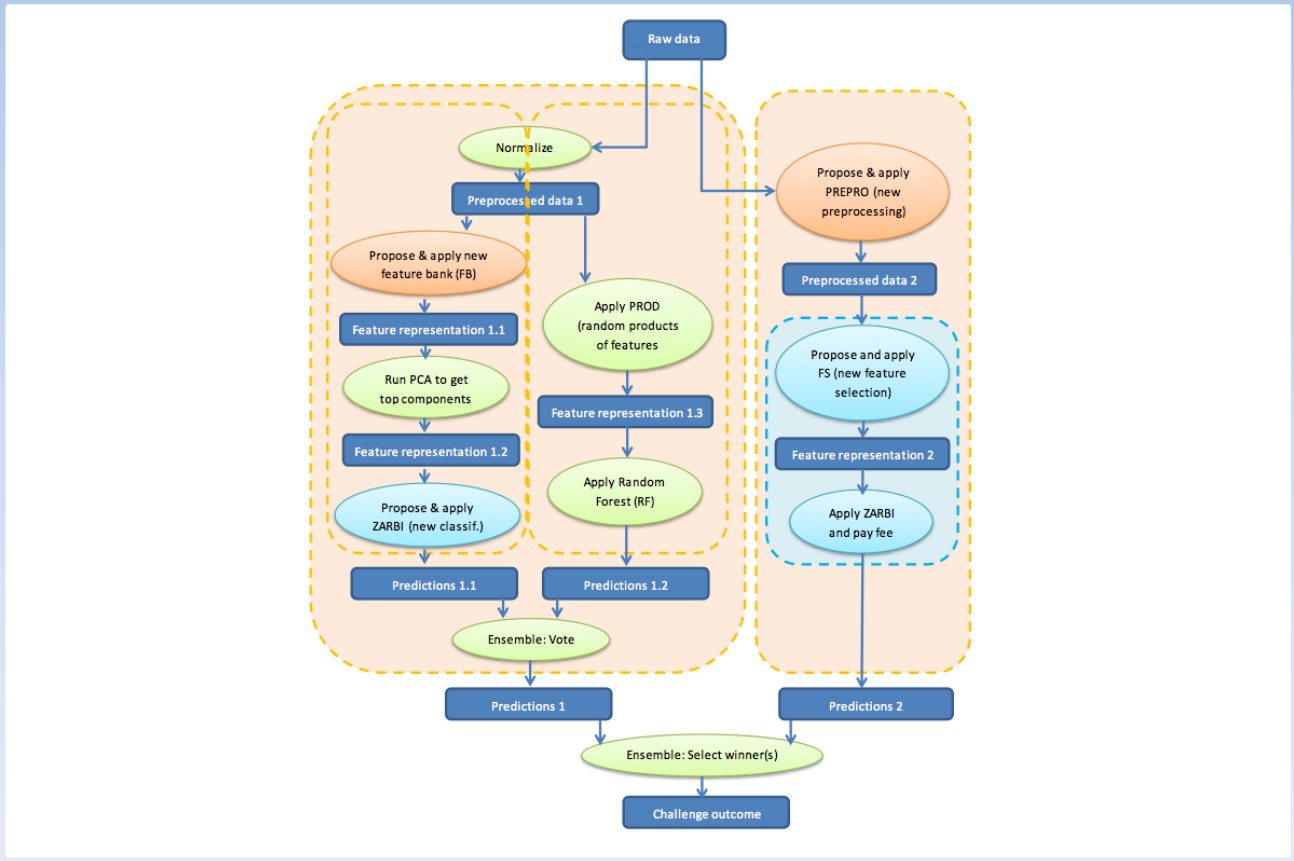
Come to my office hours…

Wed 2:30-4:30 Soda 329

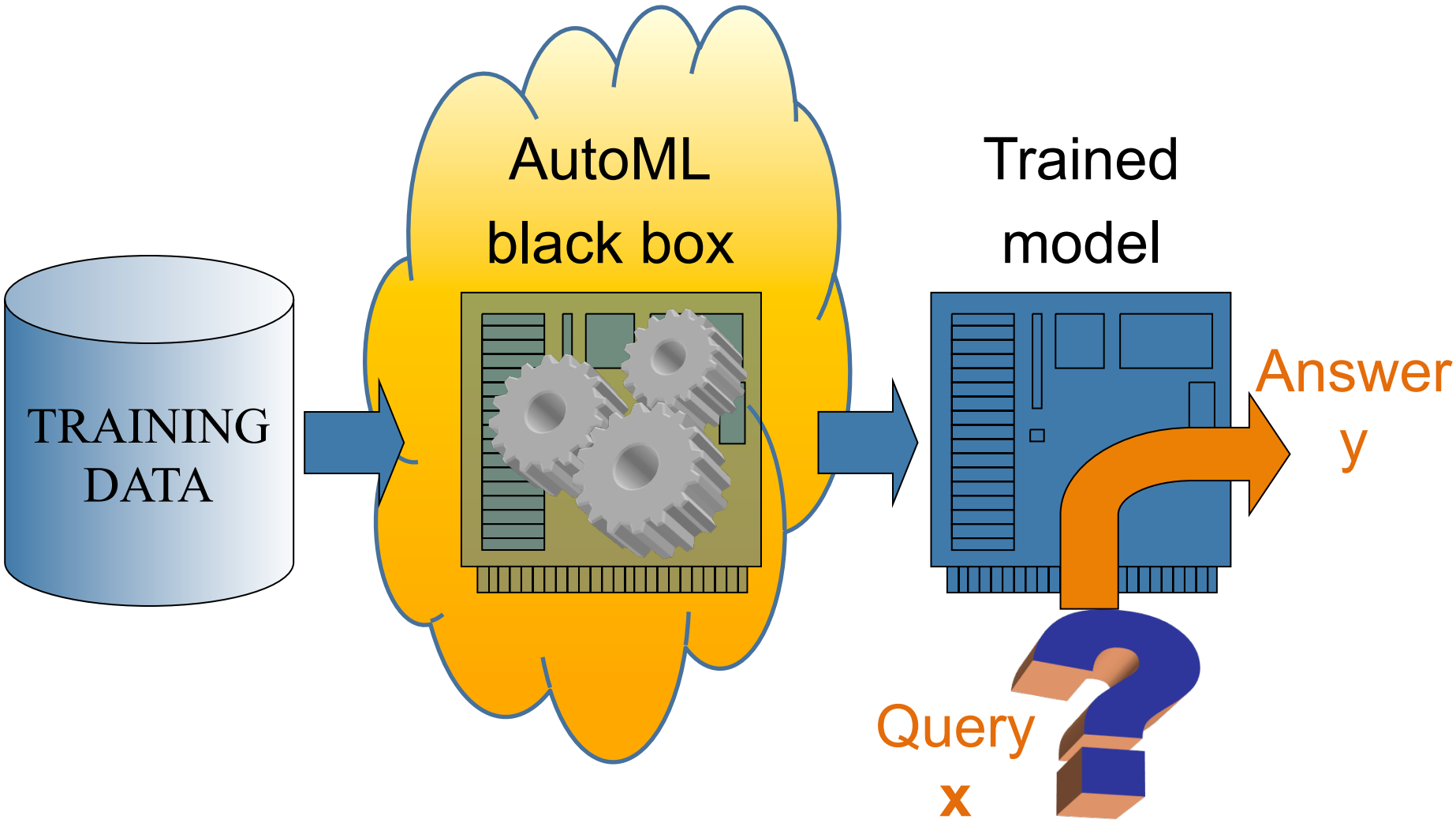
Last time

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

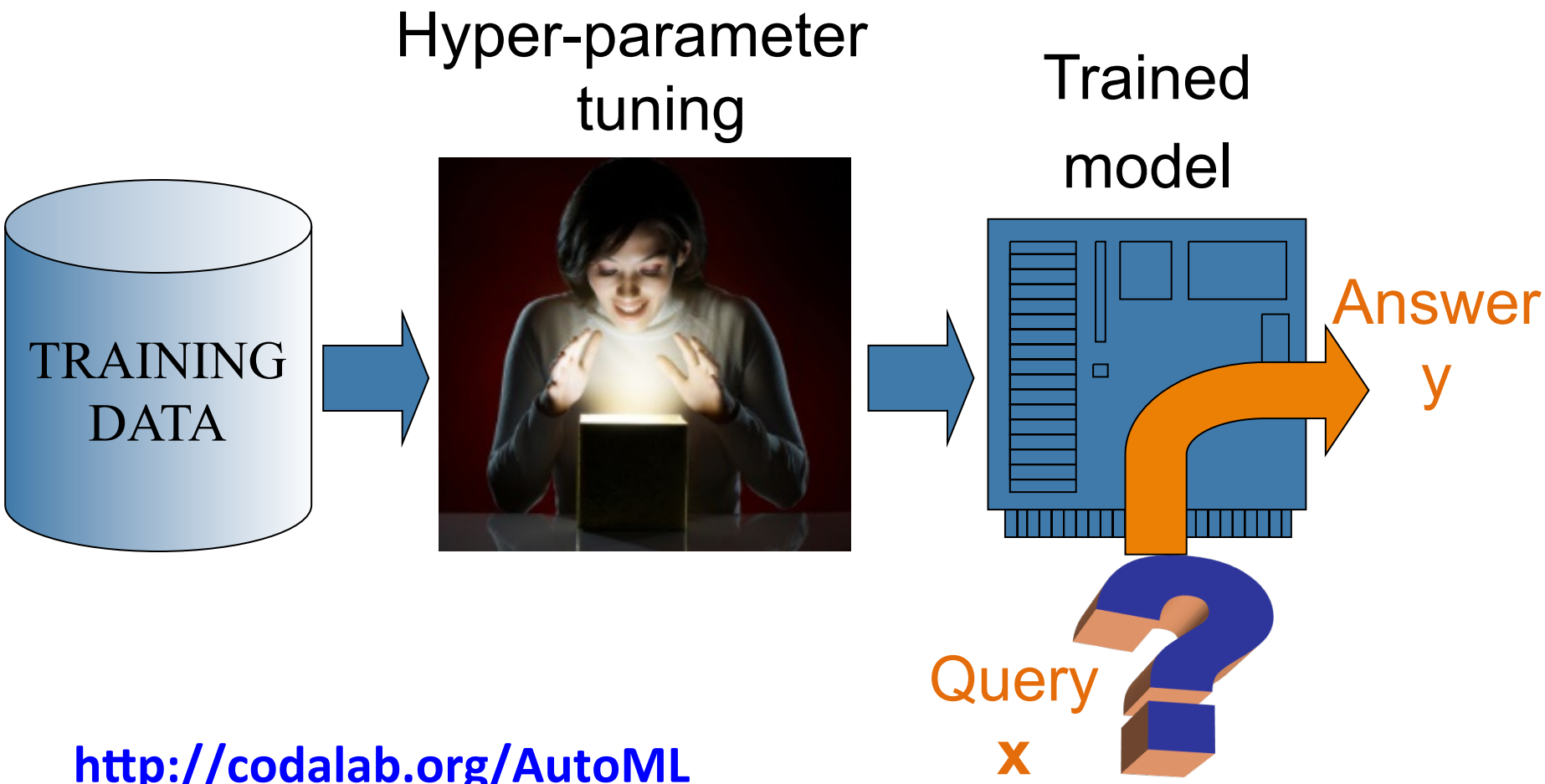
Today: model search



Remember: The DREAM

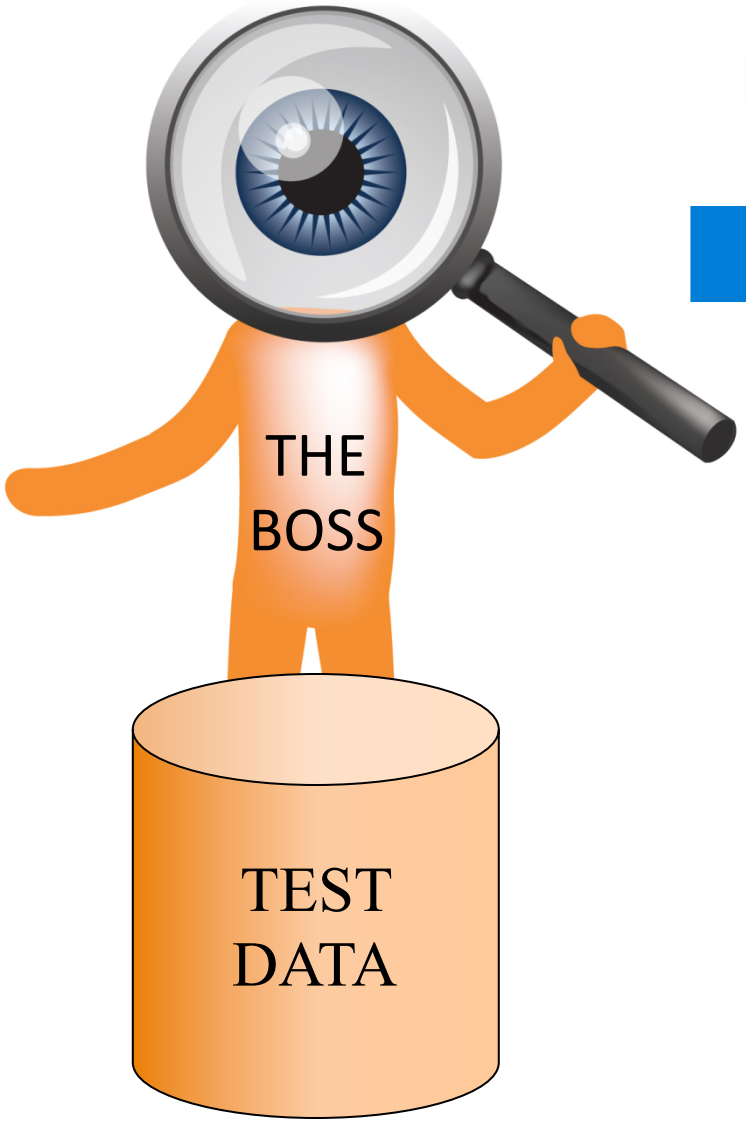


Remember: The REALITY

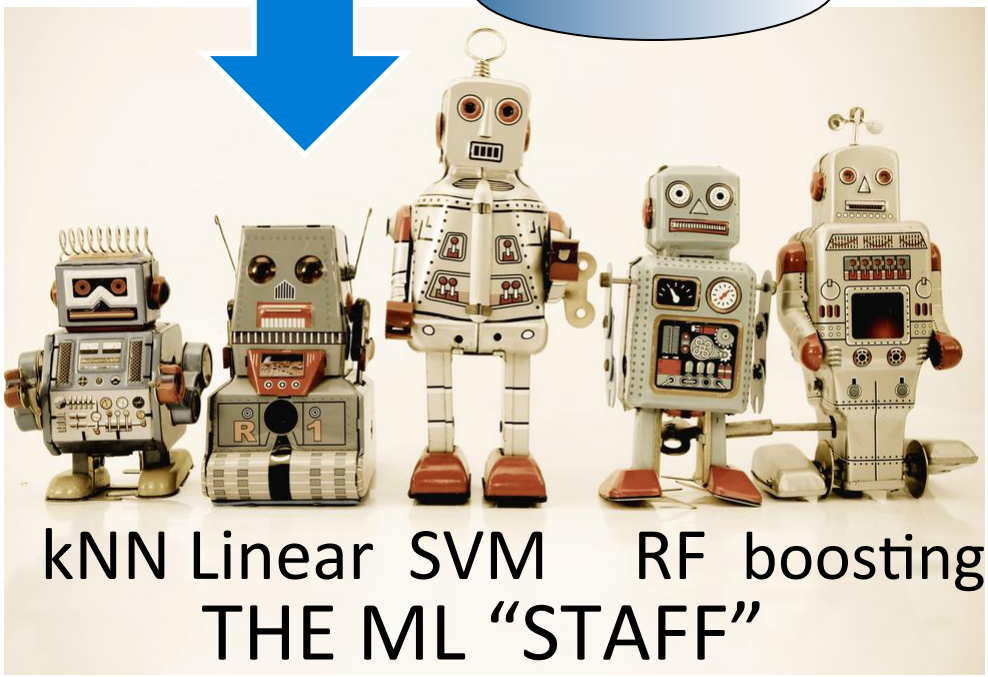


http://codalab.org/AutoML

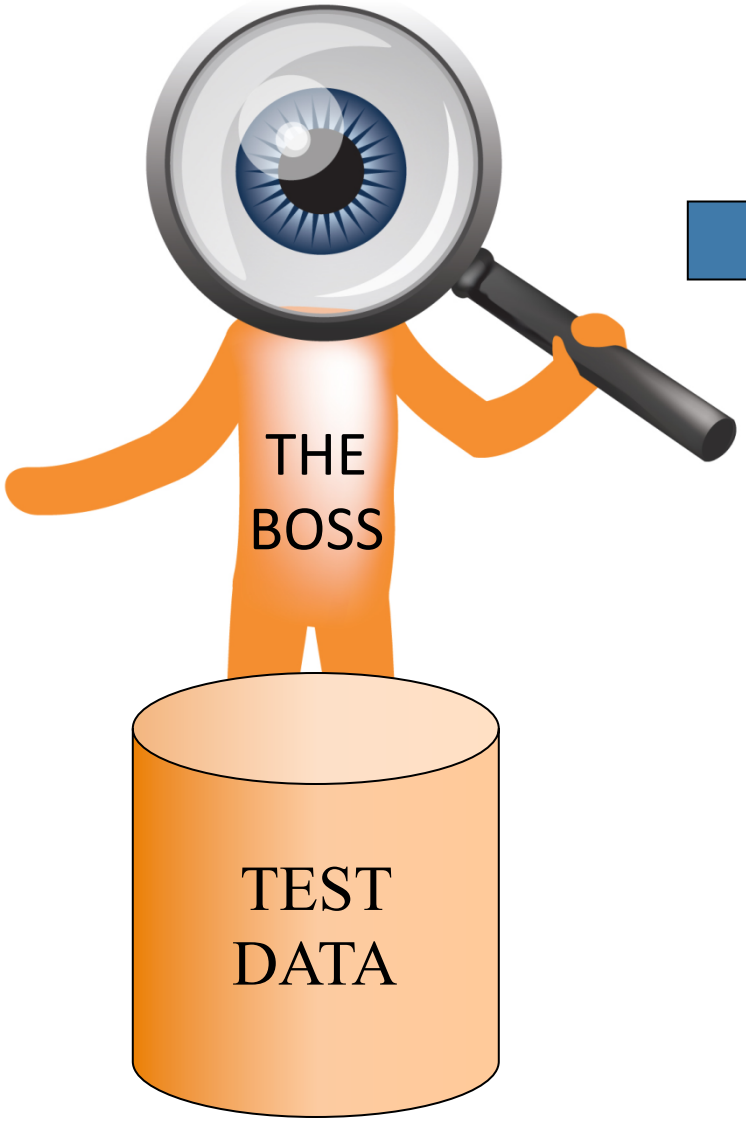
Changing roles again!

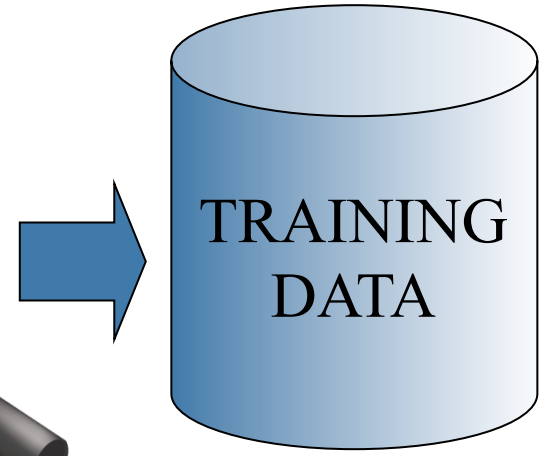


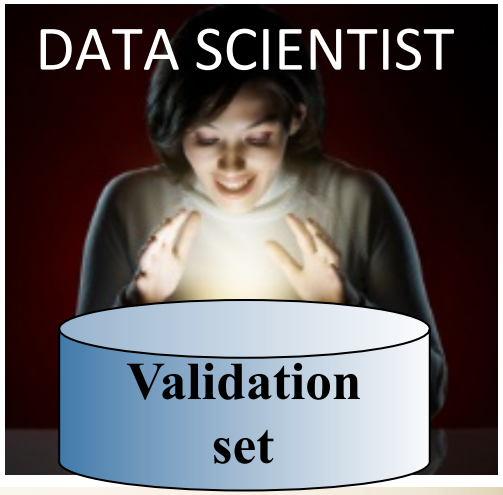
LAST TIME

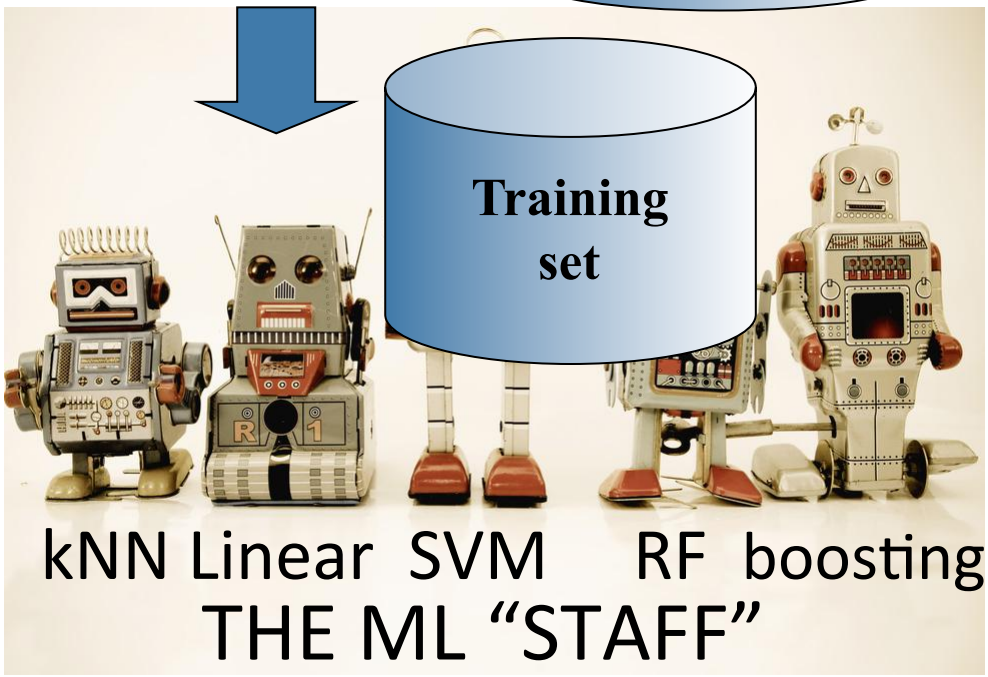


Changing roles again!









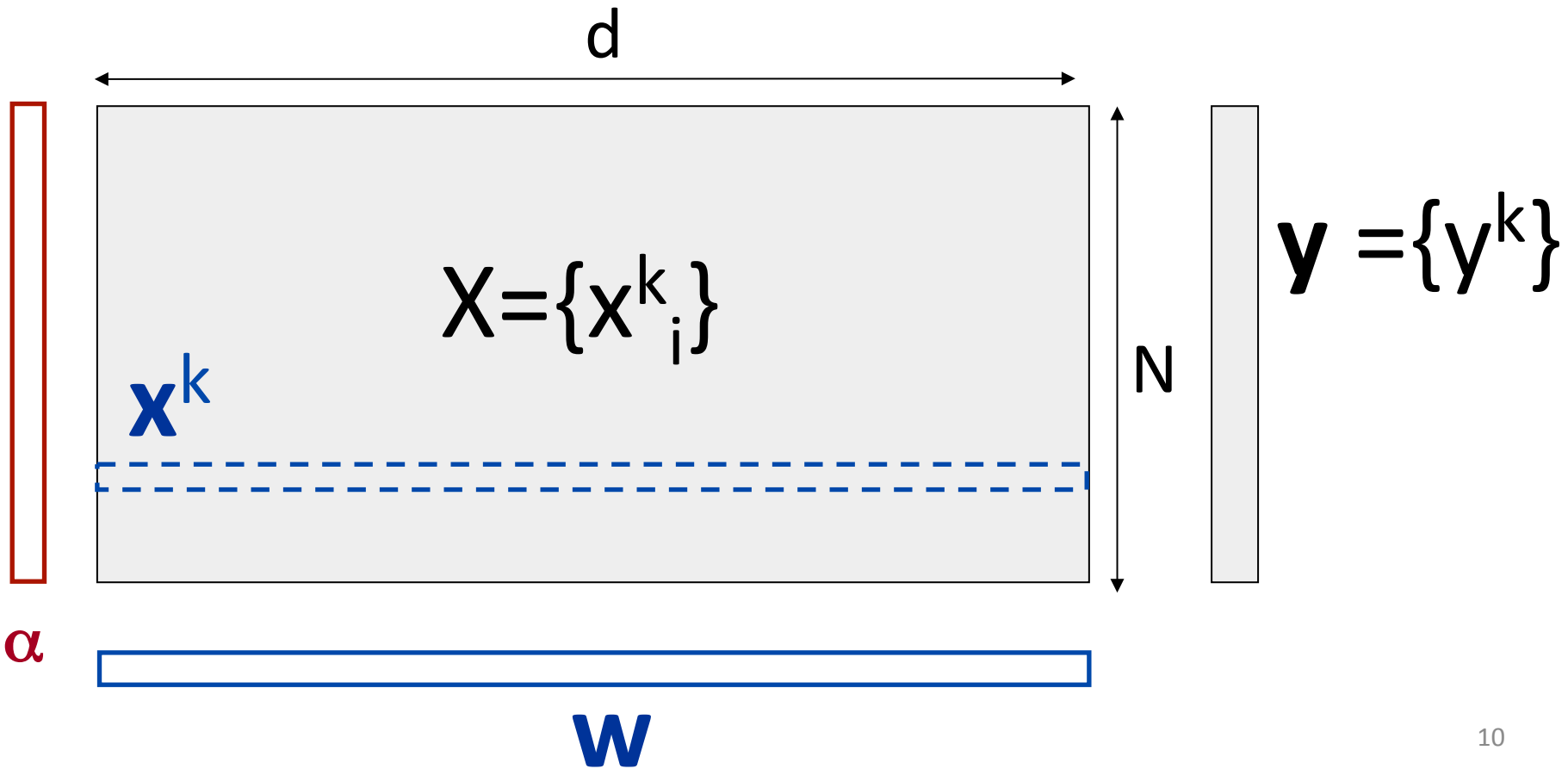

DATA SCIENTIST
THE SEXIEST JOB OF THE 21ST CENTURY

DATA SCIENTIST

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

Produce ~~3~~ 4 models:

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

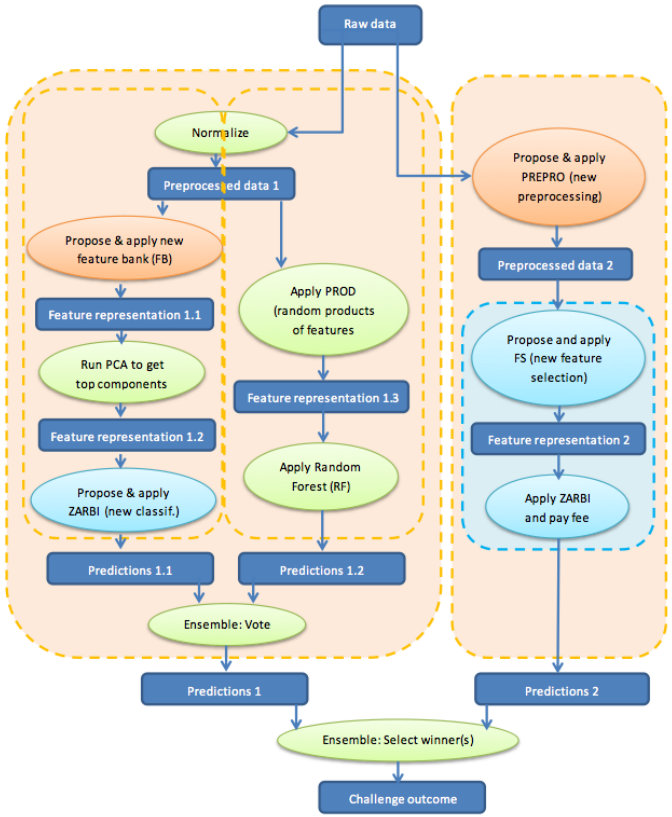


What are hyper-parameters

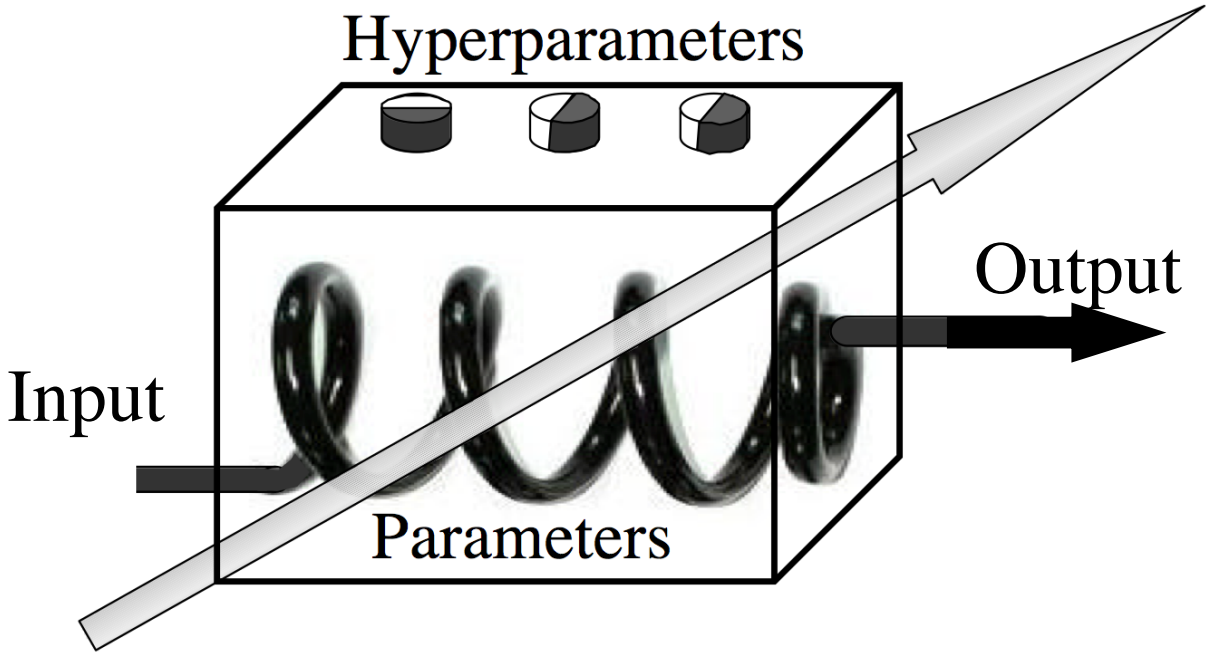
- **Preprocessing** (feature selection, feature construction, feature or pattern normalizations, etc.) **[~ 3 HP]**
- **Model** (linear, kernel, kNN, decision trees, etc.) **[1 HP]**
- Model **hyper-parameters** (number of neighbors in kNN, kernel choice polynomial or Gaussian, kernel parameters q, σ, 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

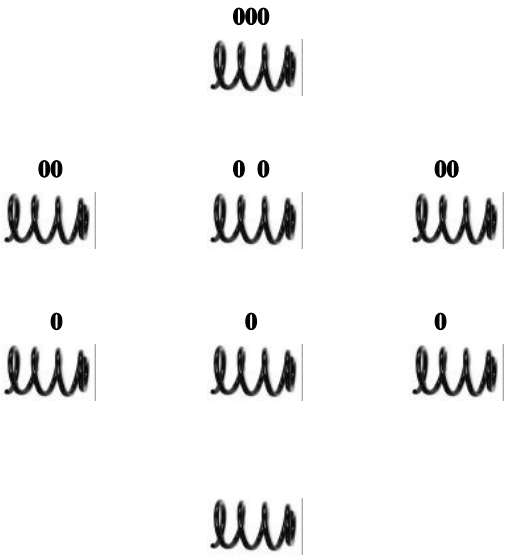
a m c .;) a m .;)

a am s

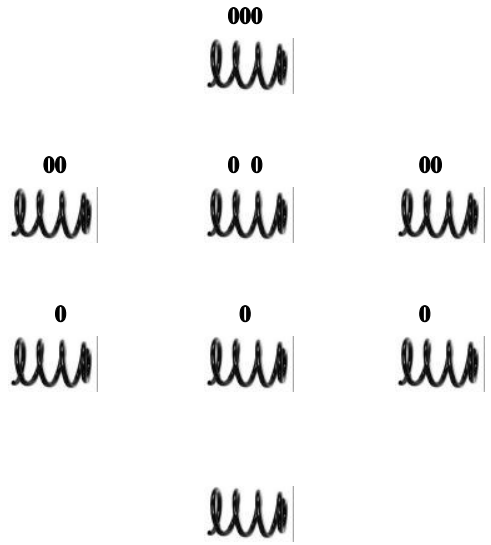
$$f^{**} = \text{argmin}_{\boldsymbol{\theta}} R_2[f^*, D] \text{ , such that } f^* = \text{argmin}_{\boldsymbol{\alpha}} R_1[f, D]$$

Search strategies

Filters



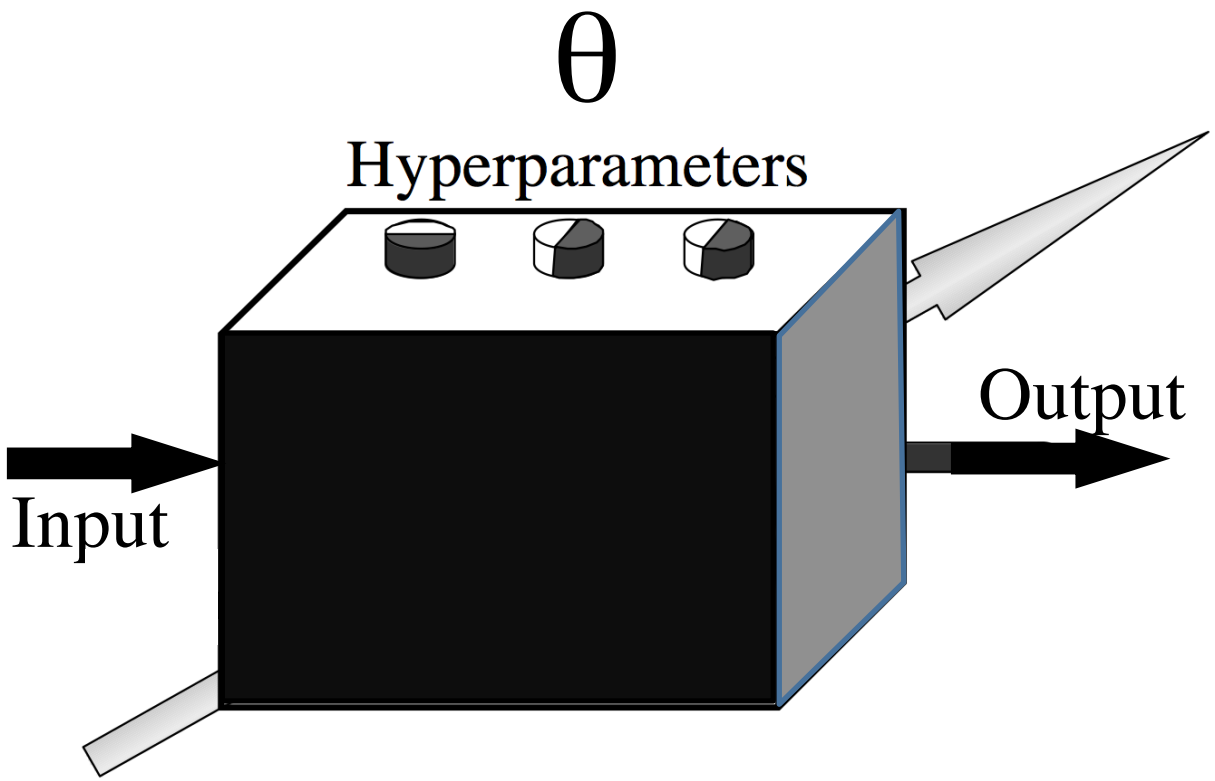
Wrappers



Embedded methods

Wrappers:

your learning machine is a black box



You cannot compute $\nabla_\theta R$ or $\partial R/\partial \theta_i$

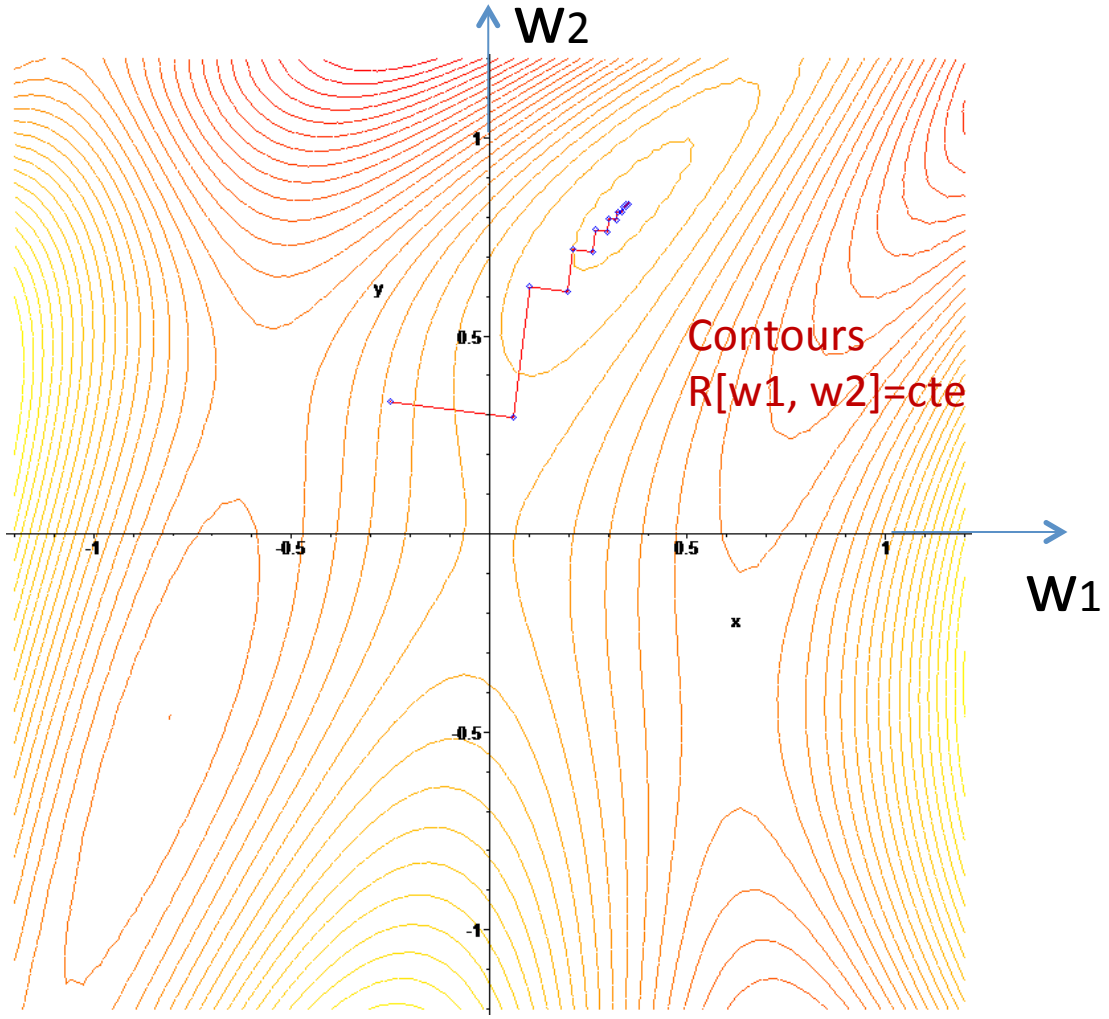
No gradient descent!

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

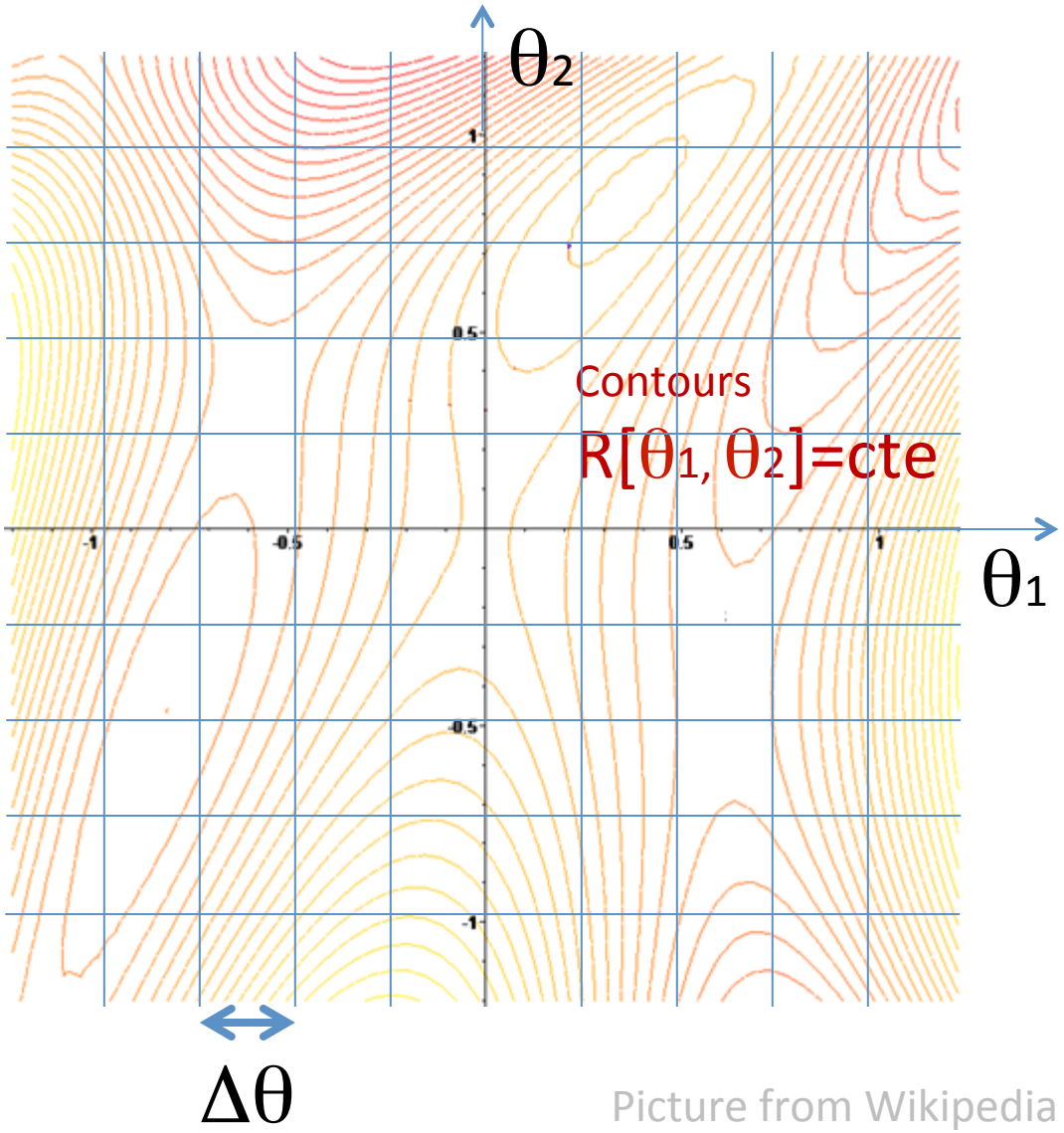
What you have to do:

Grid search:

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

$\Delta\theta$ small: too slow.

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



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 evalulations of $R[\boldsymbol{\theta}]$! 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!!)

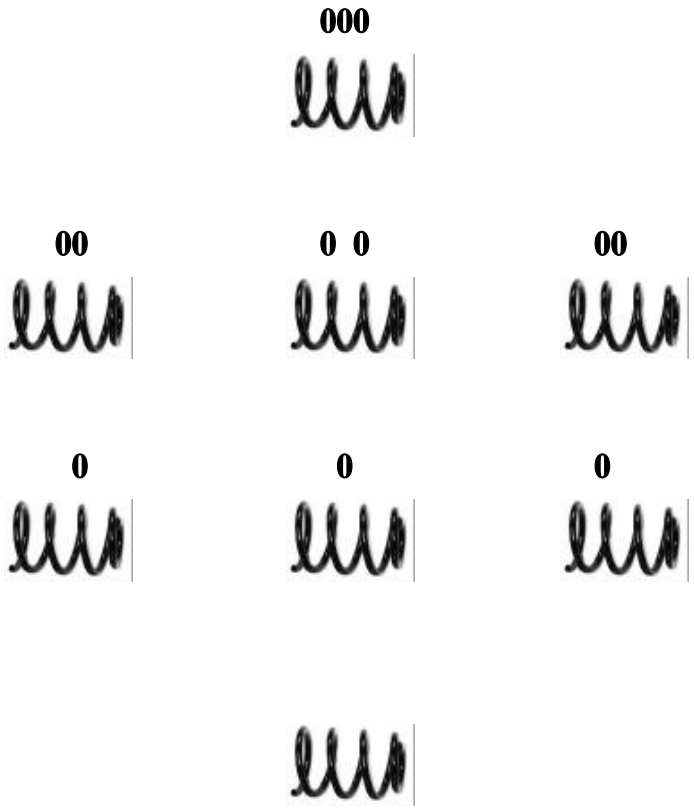
Your options

1. **Reduce the number of hyper-parameters.**

2. **Learn how to do "fancy" search.**

You chose:
Reduce HP space

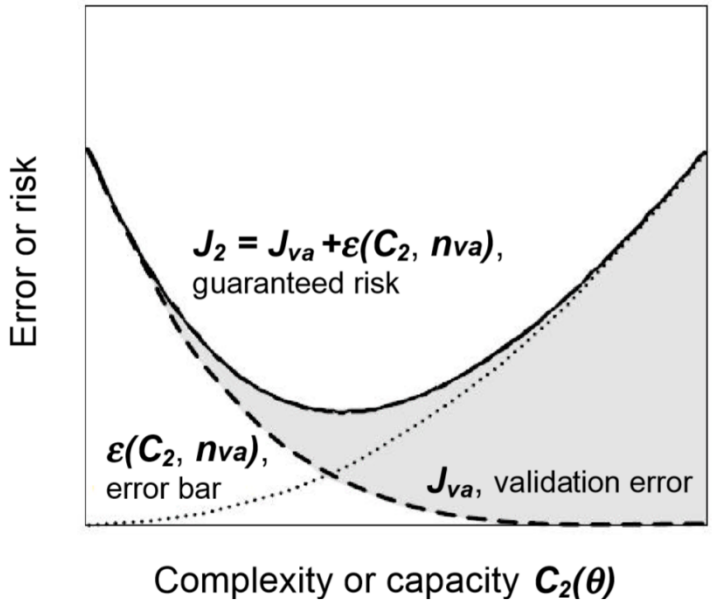
Filter methods



You chose:
Reduce HP space

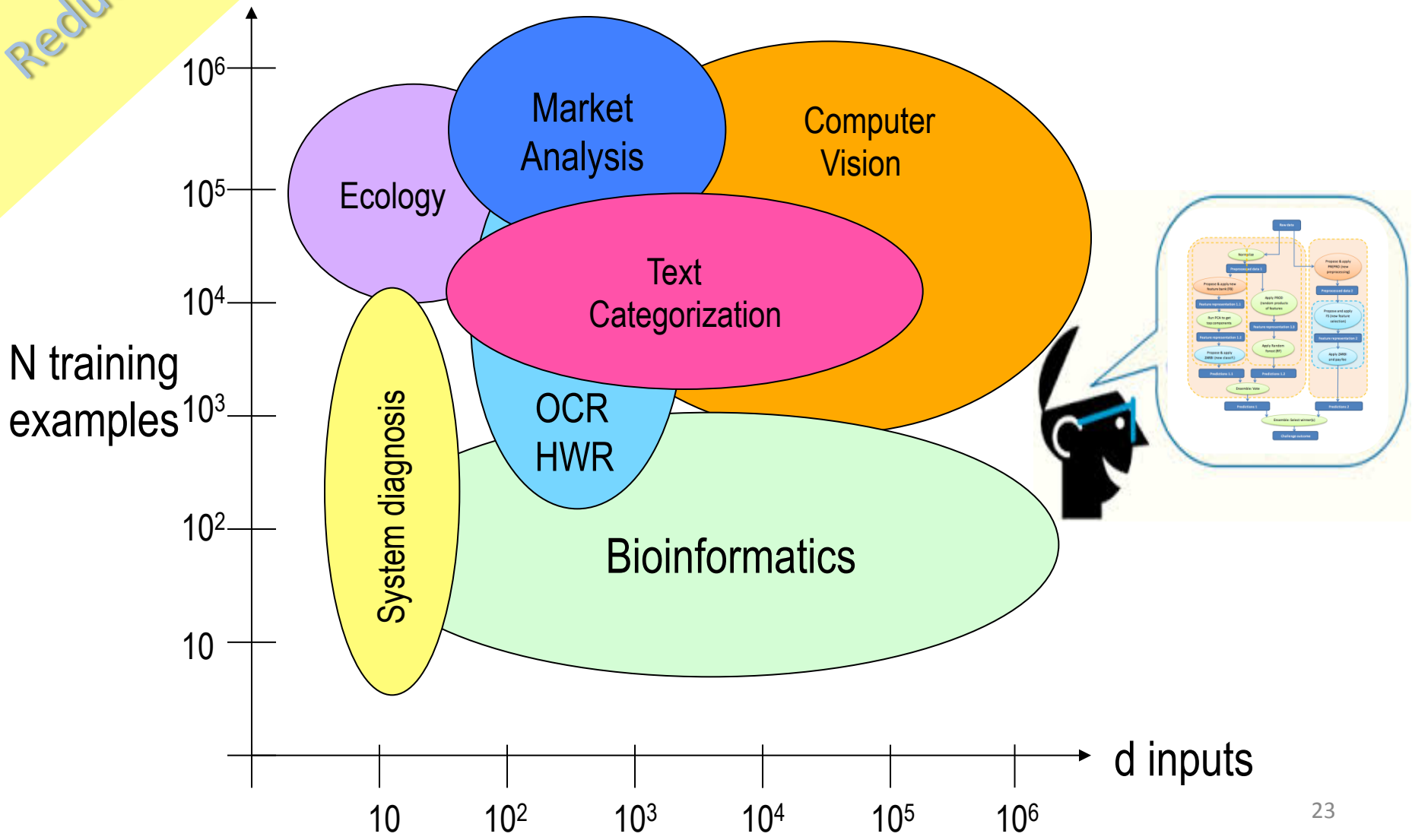
Good choice!

- Reducing HP space ALSO reduces the risk of overfitting!



You chose:
Reduce HP space

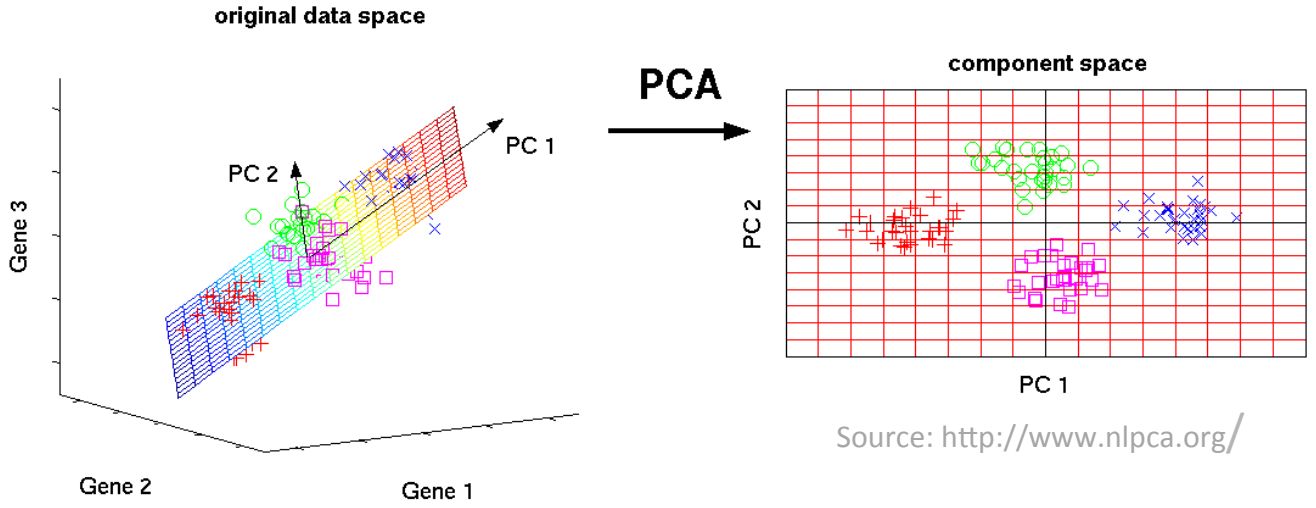
Prior knowledge



You chose:
Reduce HP space

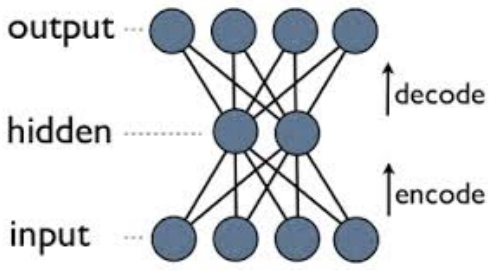
Unsupervised learning

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



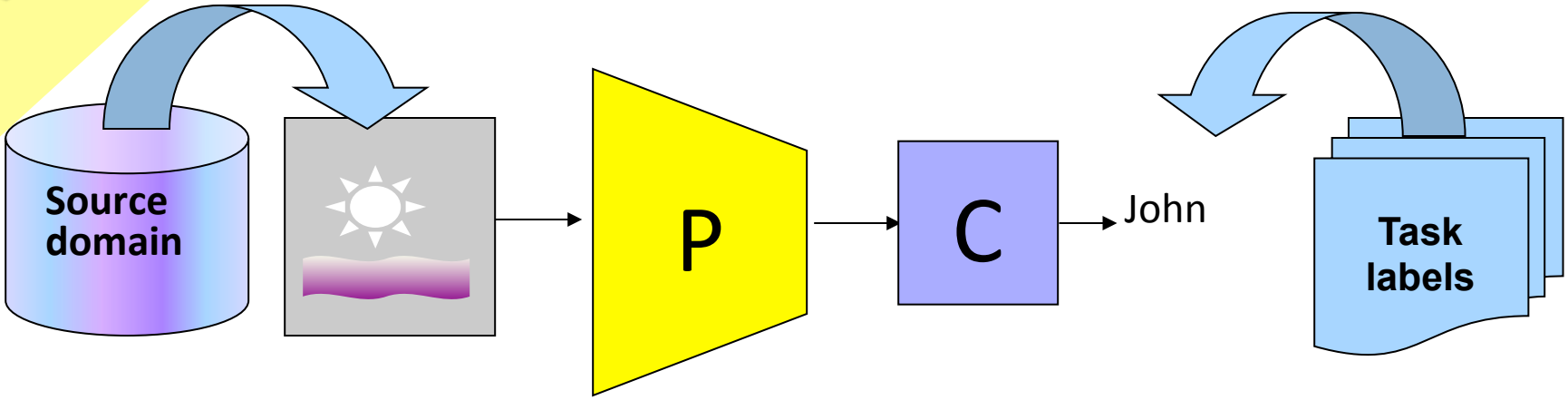
Source: http://www.nlpca.org/

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



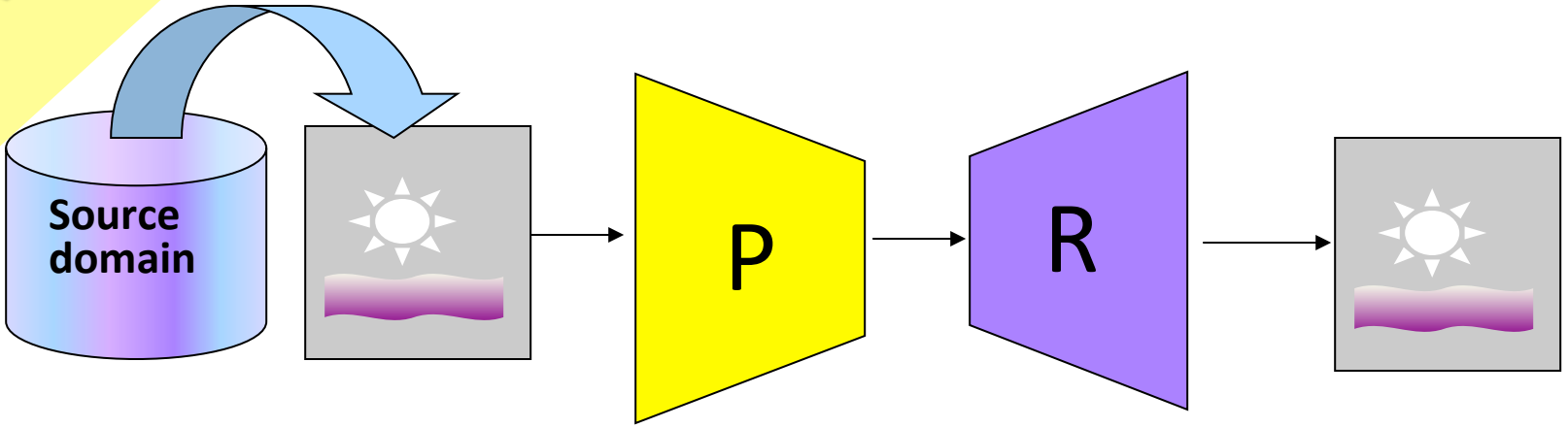
You chose:
Reduce HP space

Transfer learning



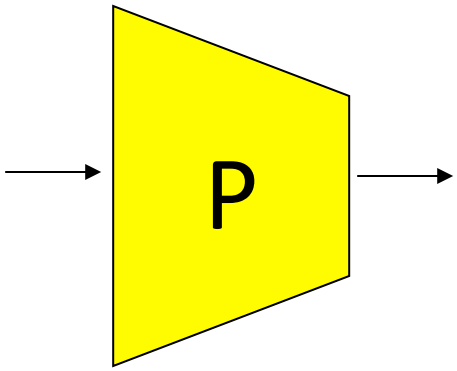
You chose:
Reduce HP space

Transfer learning



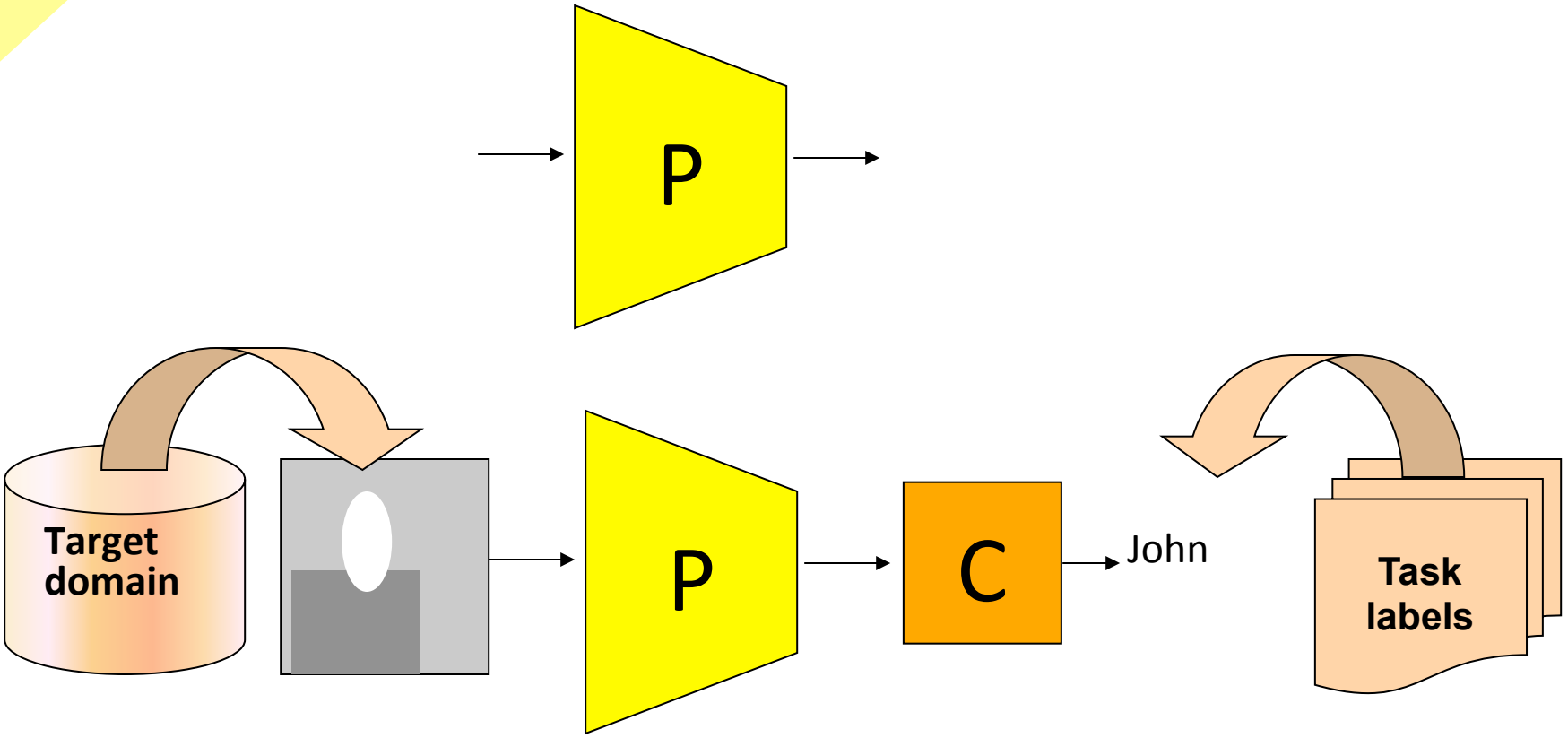
You chose:
Reduce HP space

Transfer learning



You chose:
Reduce HP space

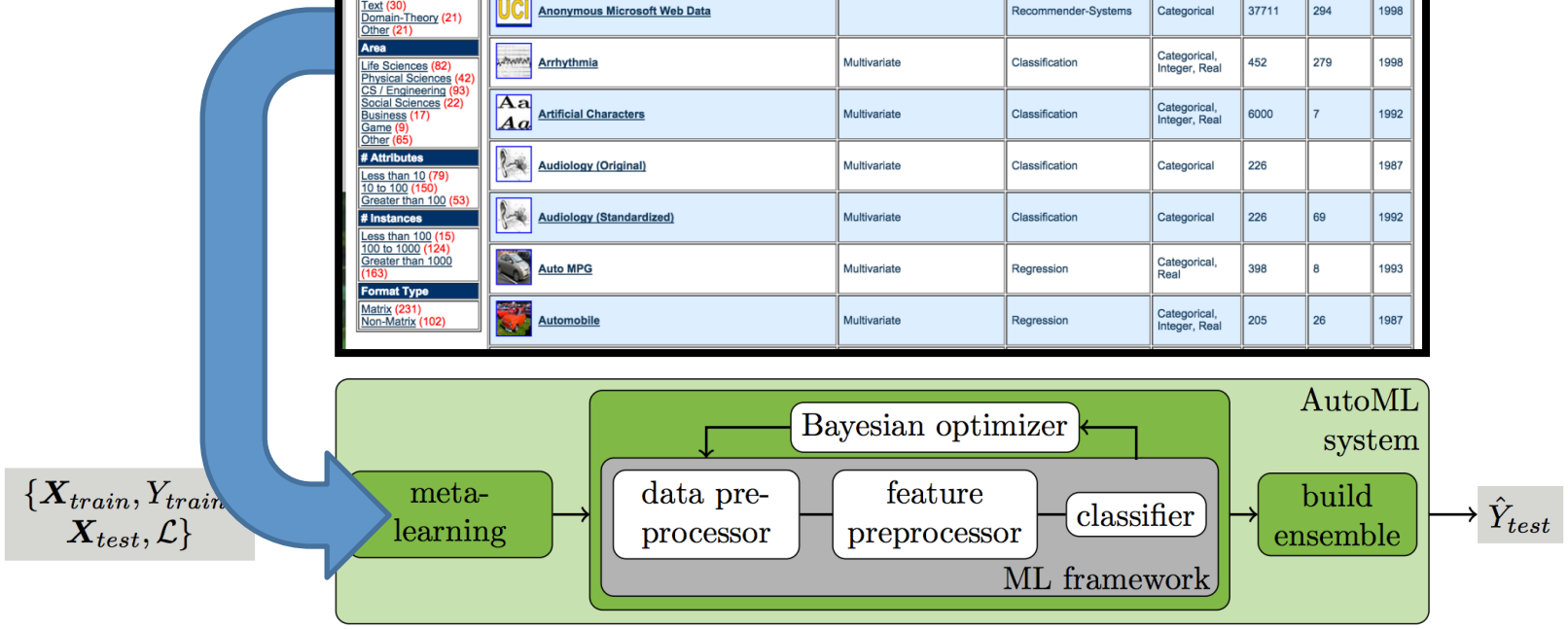
Transfer learning



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

You chose:
Reduce HP space

Meta learning



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 chose:
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.

You chose:
Fancy search

"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}:

  $T \leftarrow$ temperature(k) **annealing schedule**

  $\theta_{new} \leftarrow$ random_neighbor(θ) **transition probability**

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

    $\theta \leftarrow \theta_{new}$

return θ

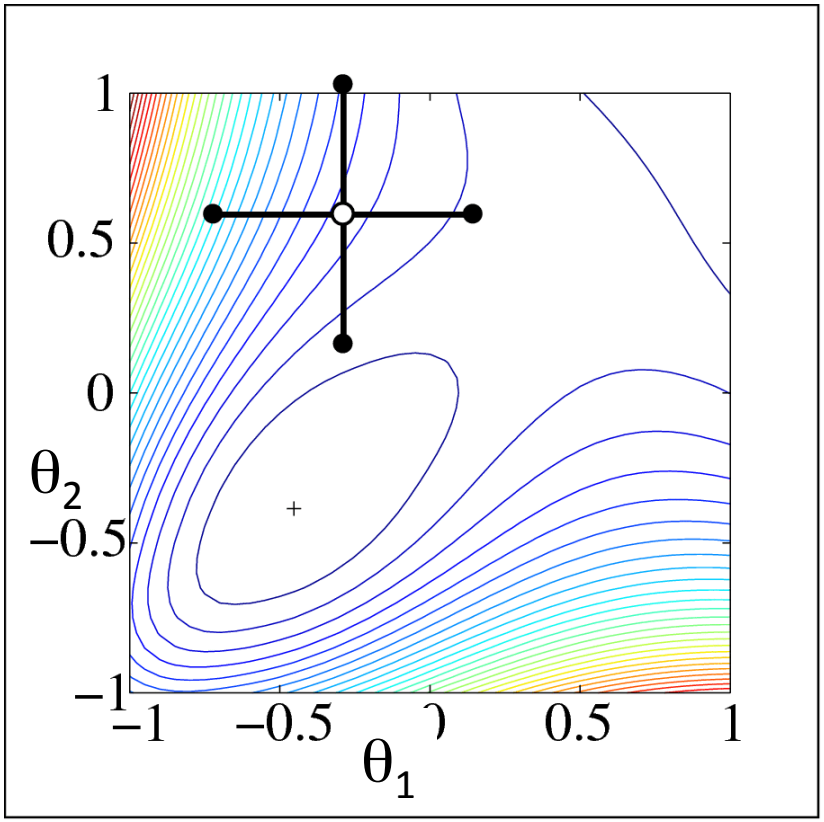
acceptance probability:

P(E, E', T) = 1 is E'<E and exp(-(E'-E)/T) otherwise

You chose:
Fancy search

Greedy search

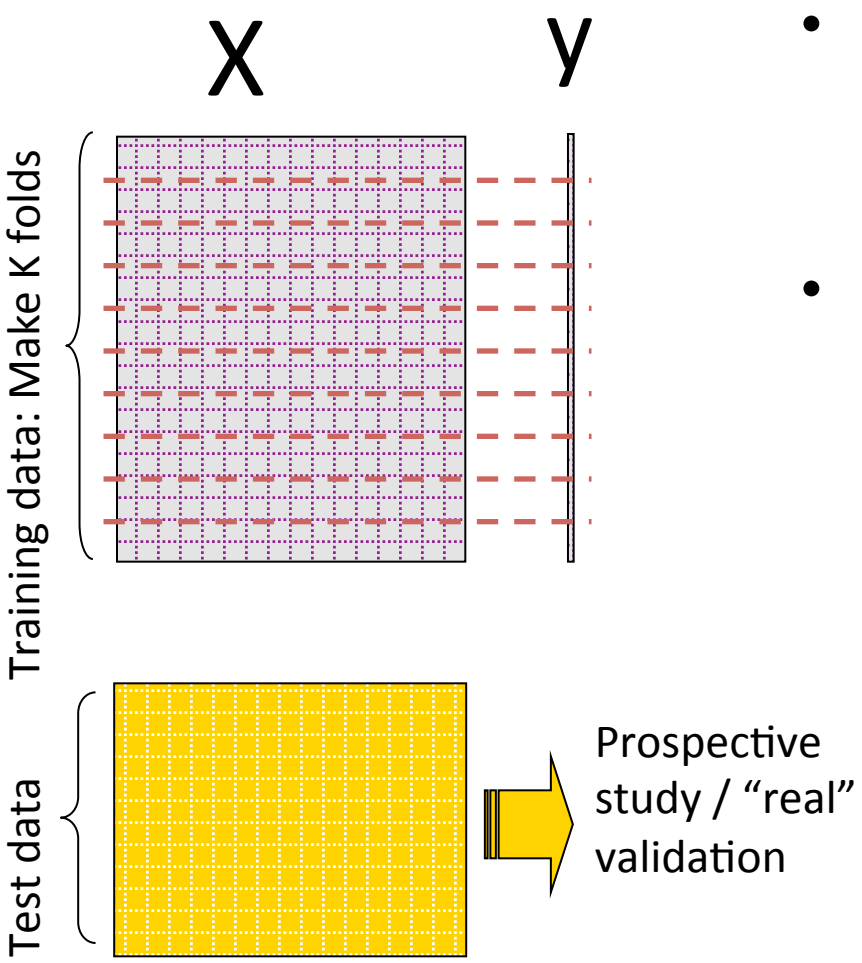
Example: pattern search



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

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

Cross-validation



- **Learning = adjusting:**

  parameters ($\mathbf{w}$ vector).

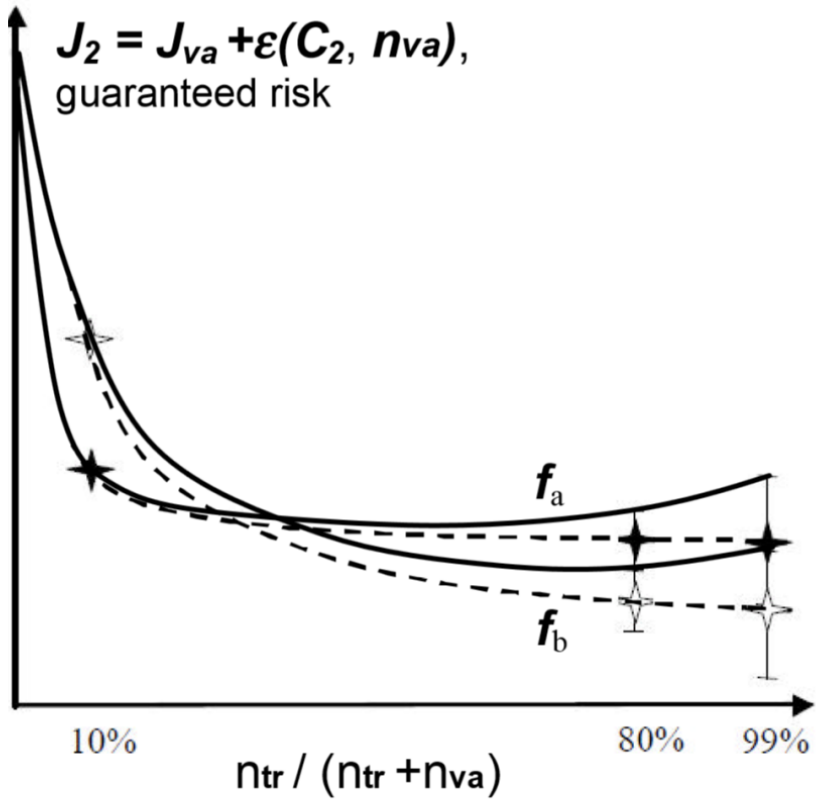
  hyper-parameters (γ, ν, q, σ).

- ***Cross-validation* with K-folds:**

For various values of γ, ν, q, σ:

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

$$E(Y|X; D) \sim \Sigma_{model} E(Y|X; model)\, P(model|D)$$

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

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