MACHINE LEARNING AND OPTIMIZATION

Bayesian Optimization

FRANCESCO ARCHETTI
UNIVERSITA' MILANO BICOCCA

Optimization: Challenges and Opportunities in the Era of Big Data

Cardiff 6-8 November 2018

- What can Optimization do for ML?
  - ➢ The support vectors of a Support Vector Machines are given by solving a mathematical programming problem
  - ➢ ML models are complex processing machines.
    - Which components to use (Automatic Algorithm Configuration, AAC)
    - How to set hyper-parameters to their «optimal» values
  - ➢ ML faces optimization with new challenging problems and the need of a different mind-set with respect to classical optimization

Jain, P., & Kar, P. (2017) Non-convex optimization for machine learning. Foundations and Trends® in Machine Learning

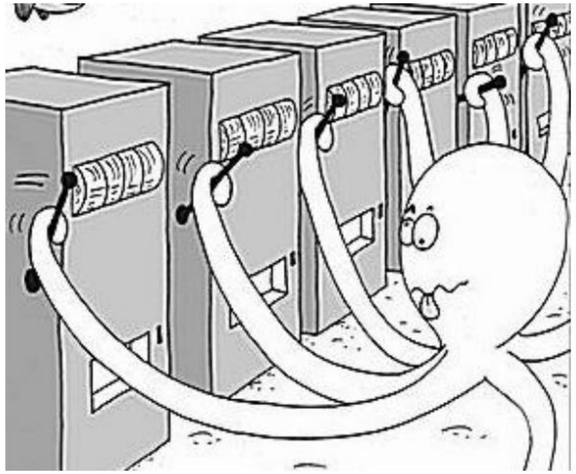
- Linear regression by least squares in the conditions of excess of covariates or data starvation. The sparse recovery approach (at most s non zero components) induces a non-convex constraint which makes the problem NP-hard
- Recommendation systems require the completion of the user-item matrix. A structural assumption is for that matrix to be low rank which makes the problem well posed. This formulation also has a convex objective but a non-convex constraint. The matrix completion problem is also NP-hard
- The traditional solution strategy to transform them into convex problems by convex relaxation. Problems: Relaxation gap, poor numerical performance, non feasibility and poor scaling properties.
- New methods tackle directly non convexity using problem structures, notably low rank, avoid NP hardness and provide a provably correct solution, offering speed and scalability
- The non-convex optimization primitives considered are: Non-convex projected gradient descent, alternating minimization, Expectation Maximization and stochastic non convex optimization

Sra, S., Nowozin, S., & Wright, S. J. (Eds.). (2012). Optimization for machine learning. MIT Press

- This book is is a collection of papers: the first part is «convexity» centered while the last 6 chapters deal with learning and uncertainty in ML and relate them with optimization
- In particular, they consider a Gaussian model and how the approximation to the function, the covariance matrix of the gradients and the Hessian allow to design efficient optimization algorithms for typical machine learning objective functions. Bayesian Optimization belongs to this algorithmic stream
- How to make the best use of a finite number of noisy function evaluations is formulated as a multi armed bandit problem. The main results extend to on-line optimization where the evaluation is associated to a reward, which sum we want to maximize. This induces a trade-off between exploration, trying to obtain more info, and exploitation i.e. selecting the option which seems to yield in expectation the highest reward.
- When the number of arms (options) is infinite we get continuous optimization with an optimistic strategy (Upper Confidence Bound). If the mapping from options to reward satisfies a Lipschitz condition convergence and guaranteed accuracy results can be derived

Multi armed bandit: independent beliefs

Bandits



- There are n machines.
- Each machine i returns a reward $y \sim P(y; \theta_i)$
  The machine's parameter θ_i is unknown

Minimization of a continuous function is a infinite multi-armed bandit

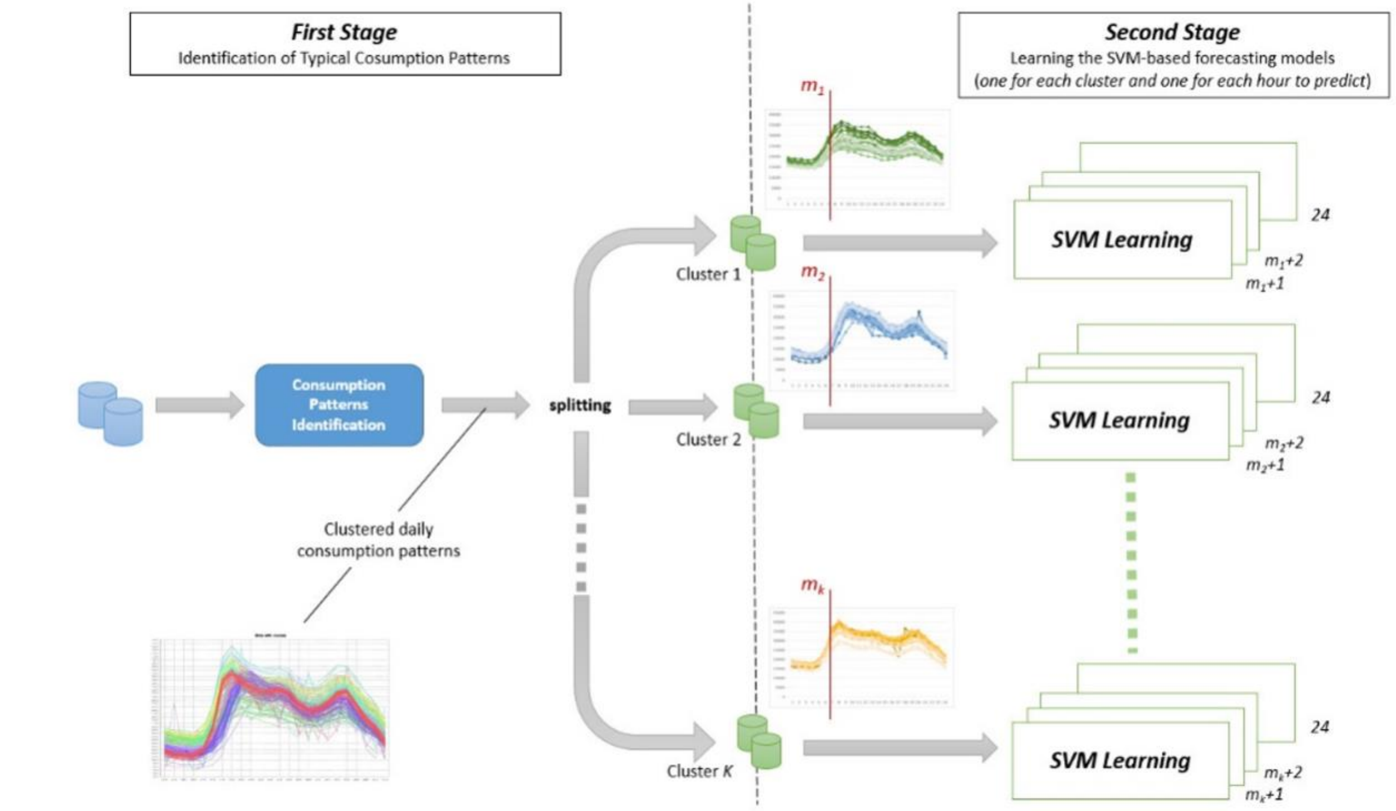
- What can ML do for Optimization? Classical optimization must evolve into a learning paradigm to take into account on-line data

  - ➢ New models of rationality: exploration vs exploitation. The need to model uncertainty/stochasticity integrating learning and decision
  - ➢ Optimization with partial information (Approximate Dynamic Programming aka Reinforcement Learning)

  - ➢ Which components to use (Automatic Algorithm Configuration, AAC)
  - ➢ How to set hyper-parameters to their «optimal» values

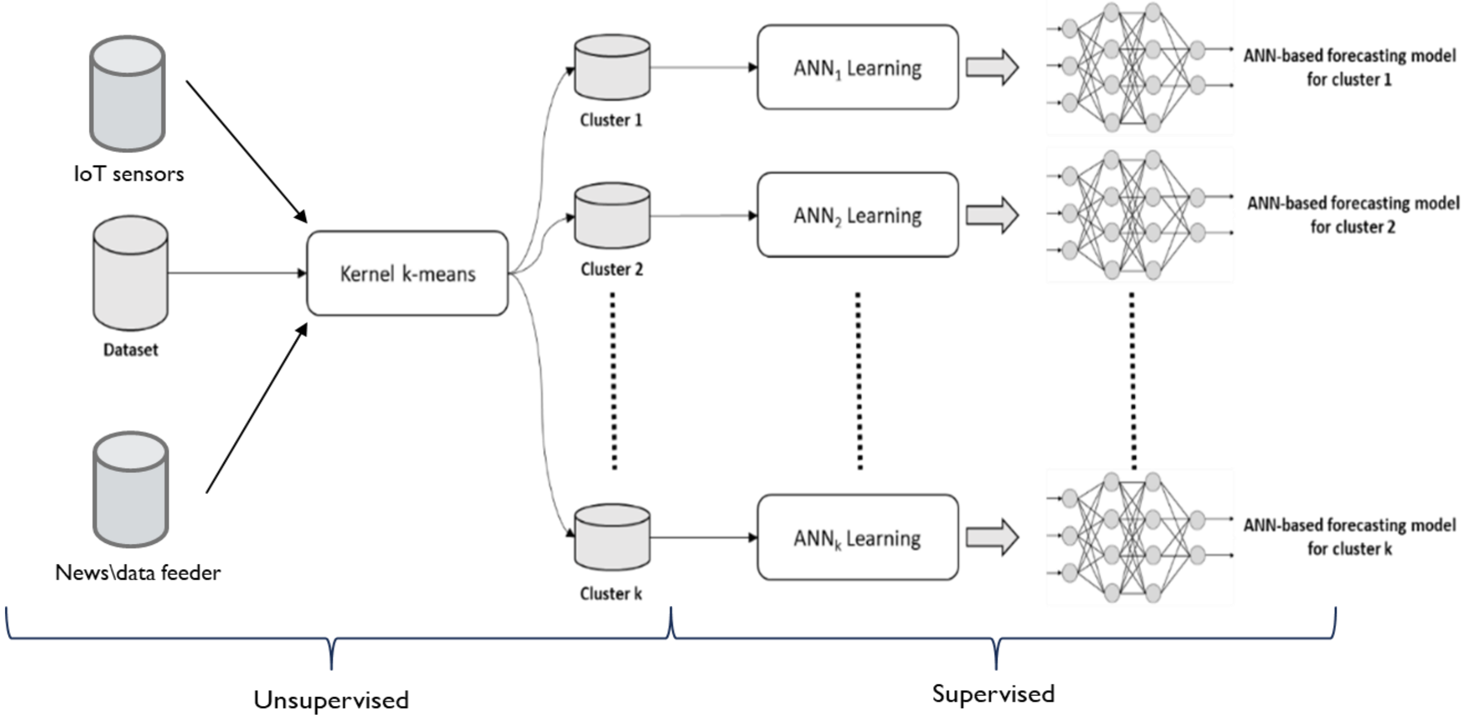
Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1), 148-175.

THE PREDICTIVE ANALYTICS PIPELINE


First Stage
Identification of Typical Cosumption Patterns
Second Stage
Learning the SVM-based forecasting models
(one for each cluster and one for each hour to predict)
Consumption Patterns Identification
splitting
Clustered daily consumption patterns
Cluster 1
Cluster 2
Cluster K
m1
m2
mk
SVM Learning
SVM Learning
SVM Learning
24
m1+2
m1+1
24
m2+2
m2+1
24
mk+2
mk+1

- *Candelieri, A. (2017) Clustering and support vector regression for water demand forecasting and anomaly detection, Water (Switzerland), 9 (3), 224.*
- *Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., ... & Zolotykh, N. (2018). Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization. Computers & Operations Research [available online, ahead of printing]*

TWO-STAGE MACHINE LEARNING FOR PREDICTIVE ANALYTICS



Candelieri, A., Giordani, I., Archetti, F. (2017) Automatic configuration of kernel-based clustering: an optimization approach, Lecture Notes in Computer Science, 10556 LNCS, pp. 34-49.

DESIGN CHOICES FOR KERNEL K-MEANS CLUSTERING

The overall number of hyper-parameters is 4 (for kernel k-means clustering) + ***k**9** (for ANN) where ***k*** is the number of clusters. The design space is quite complex (continuous, categorical and conditional variables). Moreover, its dimension changes according to the value of some variables

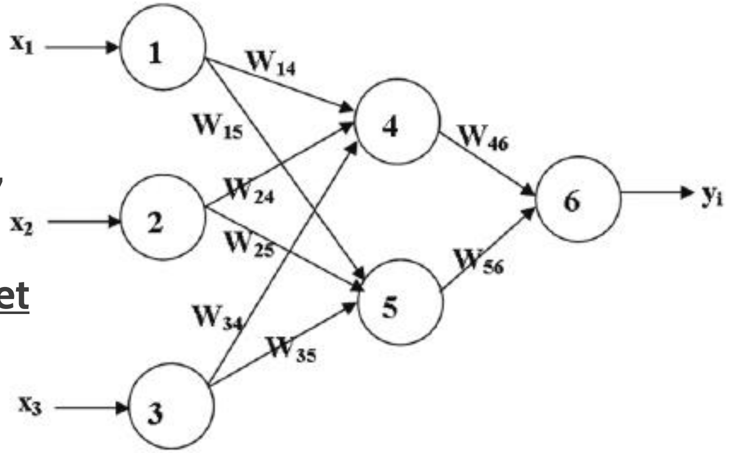
Hyperparameter	Type	Description
k	integer	Number of clusters. Possible values are form 3 to 9
kernel type	categorical	Type of kernel used in the kernel based clustering. Possible choices: linear, spline, rbf, laplace, bessel, polynomial
σ	numeric, conditioned	Hyperparameter of the rbf, laplace and Bessel kernels
degree	integer, conditioned	Hyperparameter of the Bessel and polynomial kernels

DESIGN CHOICES FOR THE ANN'S

Hyperparameter	Type	Description
hidden layers	integer	Number of hidden layers in the artificial neural network. Possible values are 1,2 or 3
neurons in the hidden layer 1	integer	Number of neurons in the hidden layer 1. Possible values are from 1 to 20
neurons in the hidden layer 2	integer, conditioned	
neurons in the hidden layer 3	integer, conditioned	
algorithm	categorical	Type of algorithm used to train the artificial neural network. Possible values are: backprop, rprop+, rprop-, sag
learning rate	numeric, conditioned	Learning rate of the backprop algorithm. Possible values are in the range [0.1, 1.0]
error function	categorical	Function used to compute training error. Possible values are: sse and ce (cross entropy)
activation function	categorical	Function used to compute the output of every neuron. Possible values are: logistic and tanh
linear output	logical	This hyperparameter defines whether to use a linear combination in the output layer of the artificial neural network or not. Possible values are TRUE or FALSE.

PARAMETERS VS HYPER-PARAMETERS

- Parameters are set automatically during the training phase, while hyper-parameters drive the training and are usually set by the user
- An example: an Artificial Neural Network
- Training set + Validation set
- Parameters → weights (w_{ij}) to be learned (through backpropagation **on the training set**)
- Hyper-parameters → number of hidden layers, neurons in every layer, activation function, learning rate, etc.
- Hyper-parameters are tuned to optimize the performance of the ANN **on the validation set**



LOSS FUNCTION

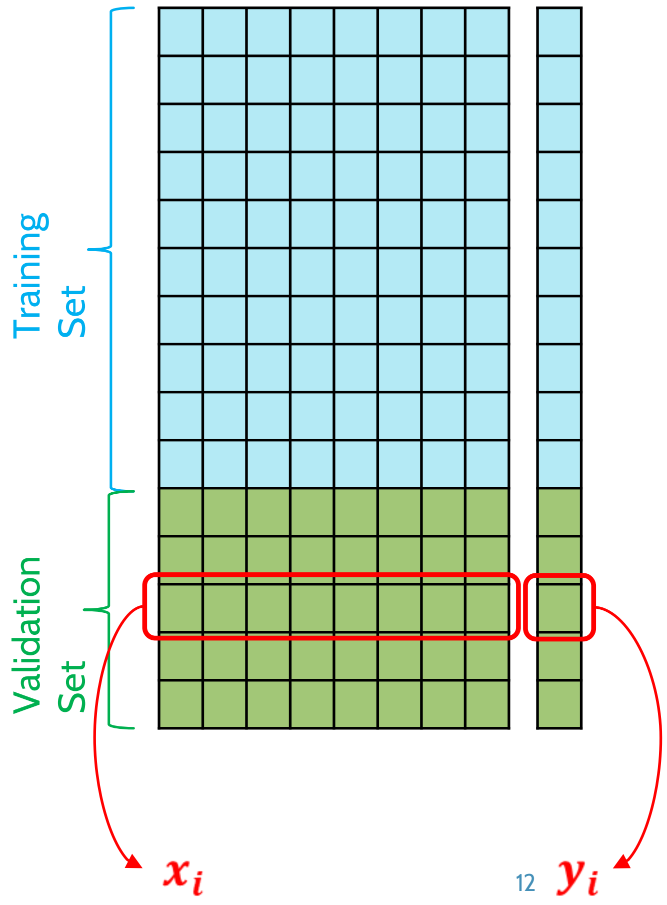
When considering a single ML algorithm, the loss function $f(\gamma)$, in the space of hyperparameters $\gamma \in \Gamma$, can be observed, with some noise, through a validation set X_V:

$$f(\gamma) = L\left(h_\gamma, X_V\right) + \varepsilon$$

Where h_γ is the prediction model obtained by running the ML algorithm on a given training set X_T to infer the value of its internal parameters and

$$L\left(h_\gamma, X_V\right) = \frac{1}{|X_V|} \sum_{(x_i, y_i) \in X_V} l(h_\gamma(x_i), y_i)$$

With $l(., y)$ the target loss function. The true function $f(\gamma)$ is unknown and can be only "glimpsed" at through noisy observations computed on the validation dataset



An example:

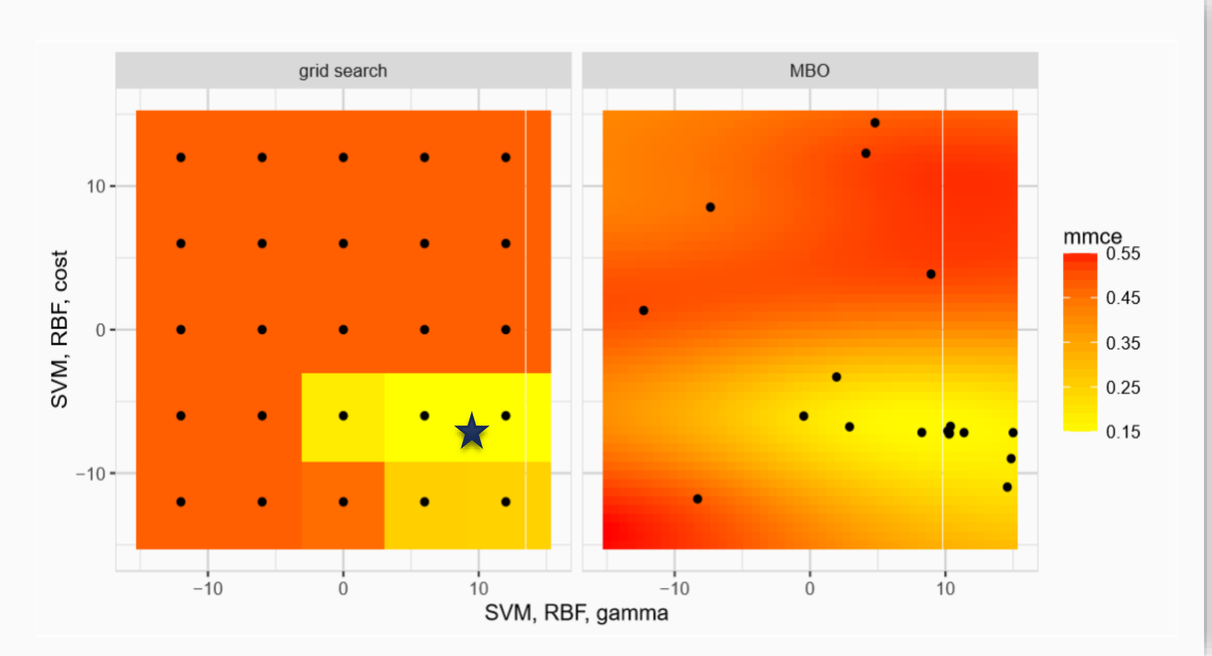
h_γ is an ANN trained on X_T with hyperparameters γ

during training the PARAMETERS are updated to minimize loss function (e.g. via backpropagation)

h_γ is then tested/validated on X_V → the loss function $f(\gamma)$ can be computed but PARAMETERS are updated no longer

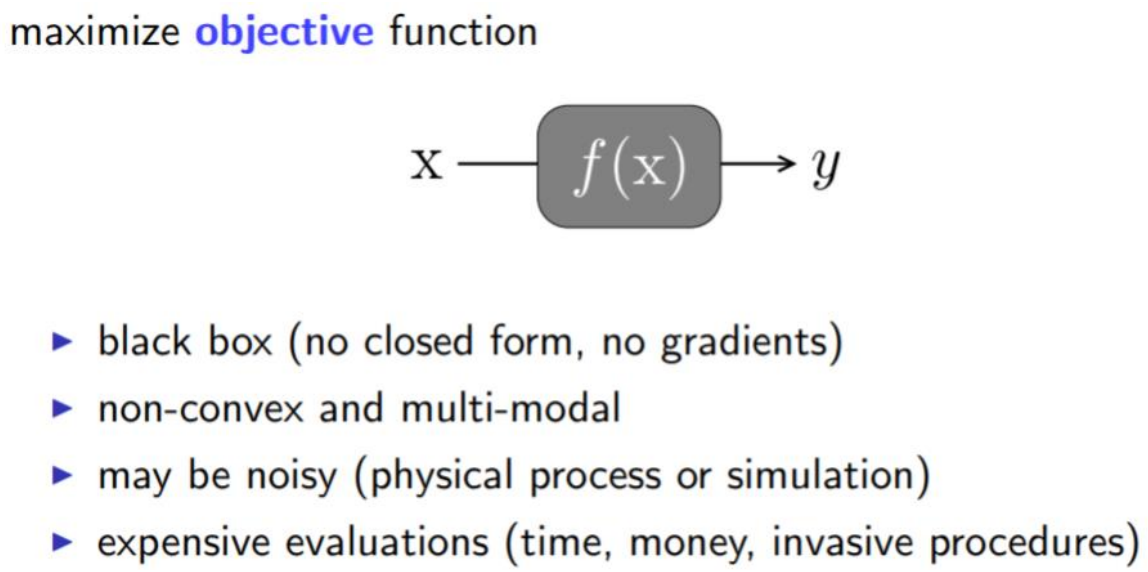
LOSS FUNCTION

- Functions that lack an analytical expression, are expensive to evaluate, and whose evaluations can be contaminated by noise: black-box function, no information on its analytical structure and it can have many local minima
- A deep learning network may easily have hundreds of hyperparameters and local minima!
- Grid/Random-Search vs Model-Based Optimization
- Global optimization problem
- Complex design space: continuous, discrete, categorical and conditional variables



- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, *104*(1), 148-175.

BLACK BOX OPTIMIZATION

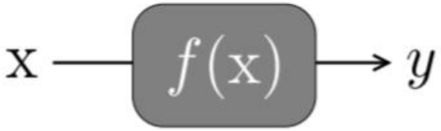


maximize **objective** function

- black box (no closed form, no gradients)
- non-convex and multi-modal
- may be noisy (physical process or simulation)
- expensive evaluations (time, money, invasive procedures)

BLACK BOX OPTIMIZATION

maximize **objective** function



- black box (no closed form, no gradients)
- non-convex and multi-modal
- may be noisy (physical process or simulation)
- expensive evaluations (time, money, invasive procedures)

local optimization tools are **not suitable**

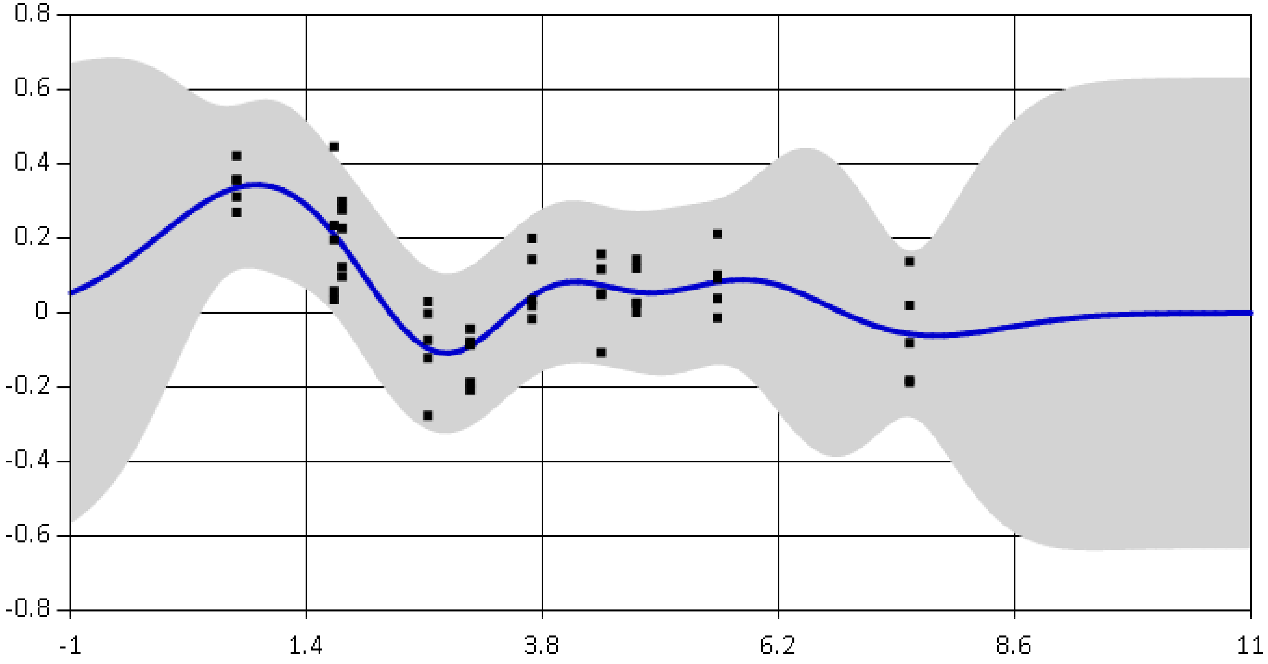
- often require gradients
- may get trapped in local optima
- do not handle measurements and structural uncertainty gracefully
- require too many evaluations (not data efficient)

GLOBAL OPTIMIZATION

- ***Exploration vs Exploitation dilemma*** !! → convexity based optimization provides very effective model for exploitation. A limit of classical numerical optimization is that it underpins the assumption of complete knowledge Need of a "cognitive model" where you learn, at a cost, something about your problem through function evaluations which you then have to exploit
- Global Optimization: Deterministic and Stochastic methods
  - *Sergeyev, Y. D., & Kvasov, D. E. (2017). Deterministic global optimization: An introduction to the diagonal approach. Springer.*
  - *Archetti, F., & Betro, B. (1978). A priori analysis of deterministic strategies for global optimization problems. Towards Global Optimization, 2, 31*
  - *Zhigljavsky, A. A. (2012). Theory of global random search (Vol. 65). Springer Science & Business Media.*
- Bayesian Optimization is part of a larger family: Sequential Model Based Optimization (SMBO)
  - *Močkus, J. (1975). On Bayesian methods for seeking the extremum. In Optimization Techniques IFIP Technical Conference (pp. 400-404). Springer, Berlin, Heidelberg.*
  - *Archetti, F., & Betro, B. (1980). Stochastic models and optimization. Bollettino della Unione Matematica Italiana, 5(17-A), 295*
  - *Zhigljavsky, A., & Zilinskas, A. (2007). Stochastic global optimization (Vol. 9). Springer Science & Business Media*
- Nature inspired algorithms: Bats, Ants, Fireflies, Krill Herd, Cuckoo, African Buffalo, Dolphin, Mushroom Reproduction Optimization
  - *Sergeyev, Y. D., Kvasov, D. E., & Mukhametzhanov, M. S. (2018). On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. Scientific reports, 8(1), 453.*
- BO is the method of choice in the academic and industrial ML community

«THE JUNGLE OF STOCHASTIC OPTIMIZATION»

TWO TYPES OF UNCERTAINTY: MEASUREMENT UNCERTAINTY

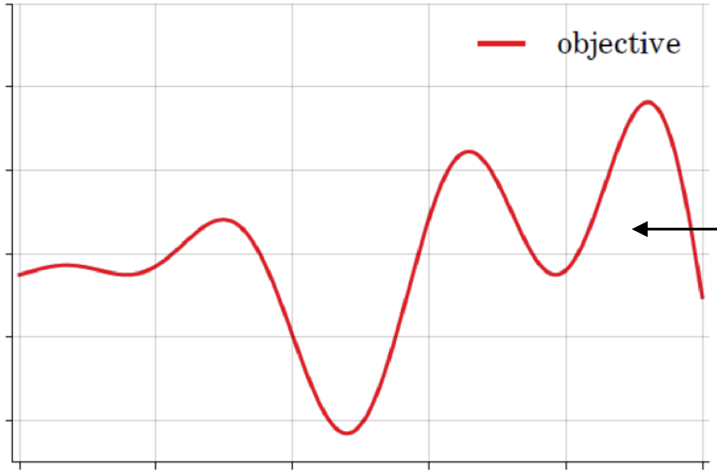


Blue is the average value

The dots are noisy observations result of each function evaluation

STRUCTURAL UNCERTAINTY

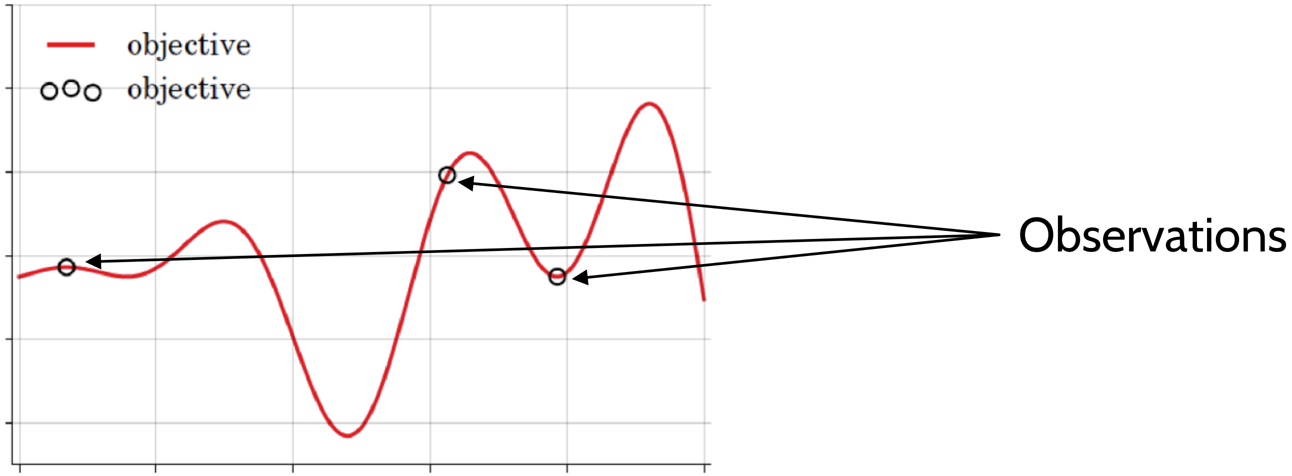
- Which model to choose



The (black-box) objective function
(i.e. loss function)

STRUCTURAL UNCERTAINTY

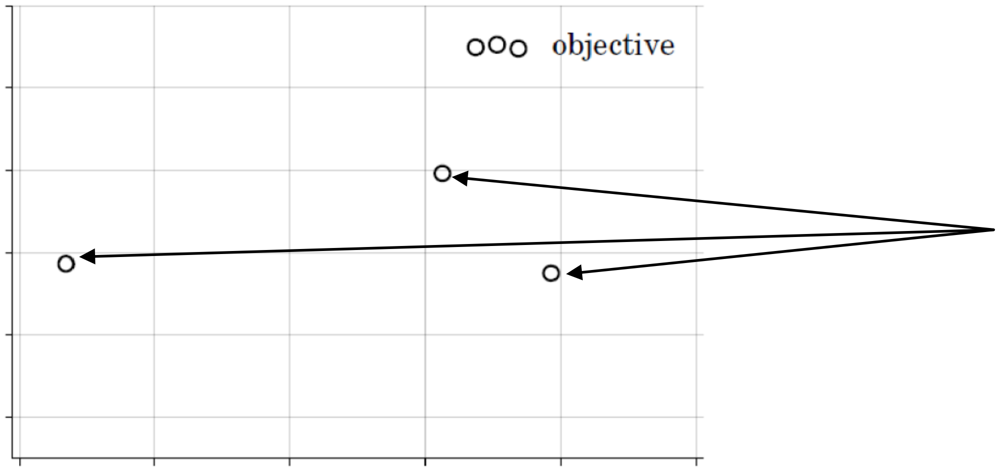
- Which model to choose



Observations

STRUCTURAL UNCERTAINTY

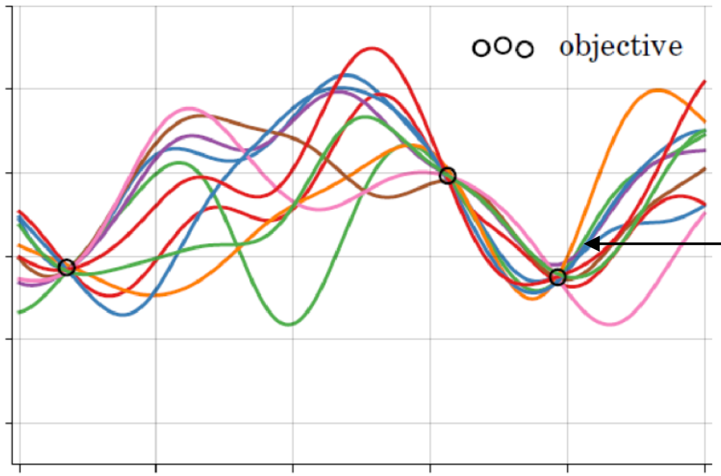
- Which model to choose



Observations

STRUCTURAL UNCERTAINTY

- Which model to choose

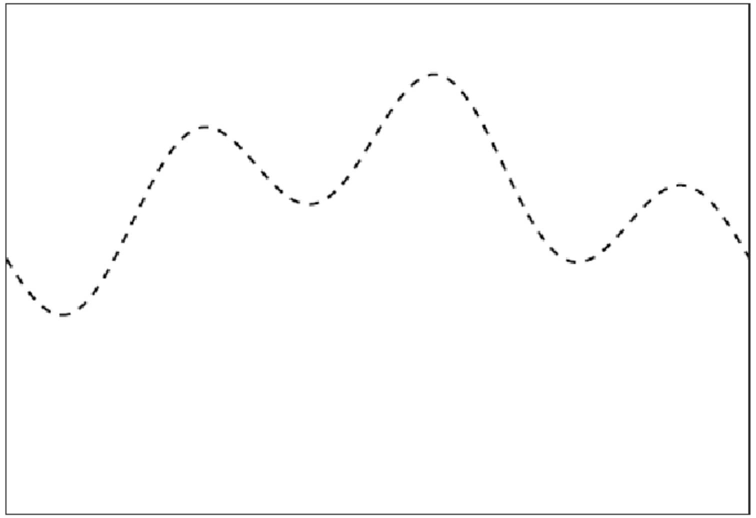


There are many functions (models) that passes through the observations

SMBO IN A NUTSHELL

BO is part of a larger family: Sequential Model Based Optimization (SMBO)

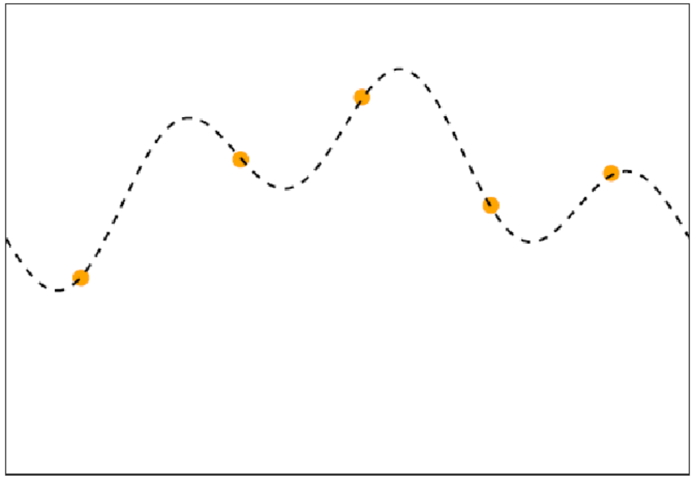
given the ability to query a black-box (- - -) repeat the following:



SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

1: given ● existing data

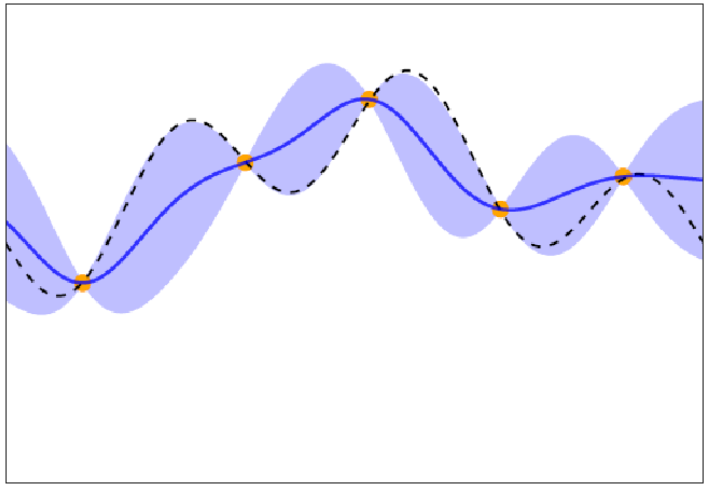


SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

1: given ● existing data
fit a (probabilistic) model

Which gives mean and variance

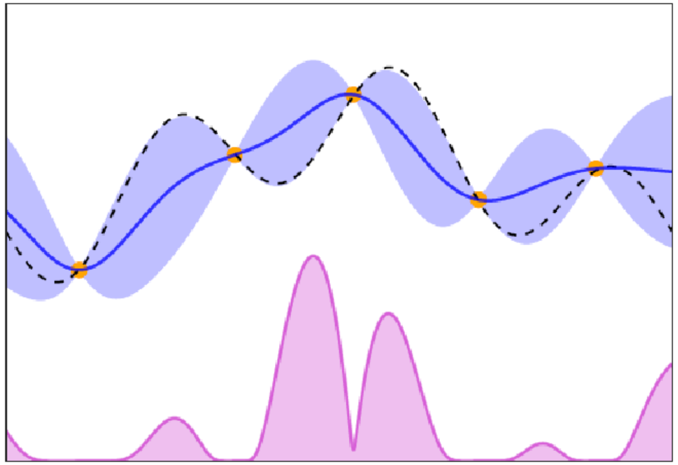


SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

1: given ● existing data
fit a (probabilistic) model

2: induce your favourite utility
which uses mean value and variance in order to blend exploration and exploitation



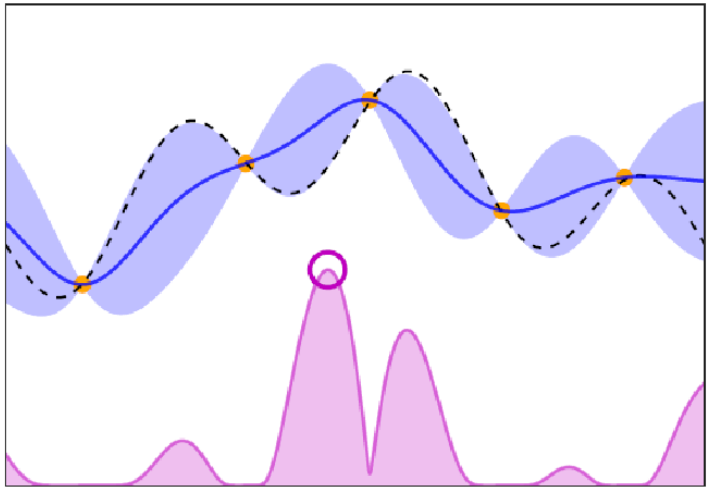
SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

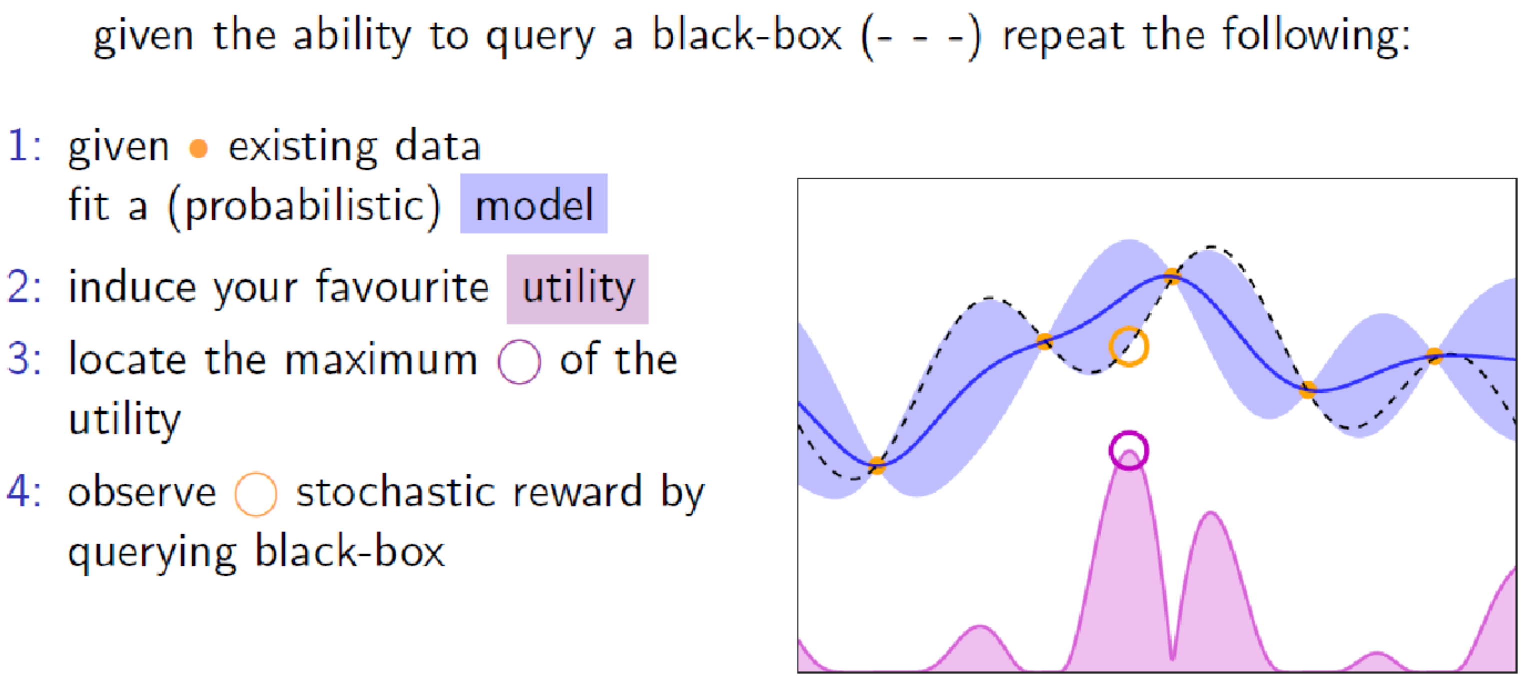
1: given ● existing data
fit a (probabilistic) model

2: induce your favourite utility

3: locate the maximum ○ of the utility



SMBO IN A NUTSHELL



given the ability to query a black-box (- - -) repeat the following:

1: given ● existing data
fit a (probabilistic) model

2: induce your favourite utility

3: locate the maximum ○ of the utility

4: observe ○ stochastic reward by querying black-box

SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

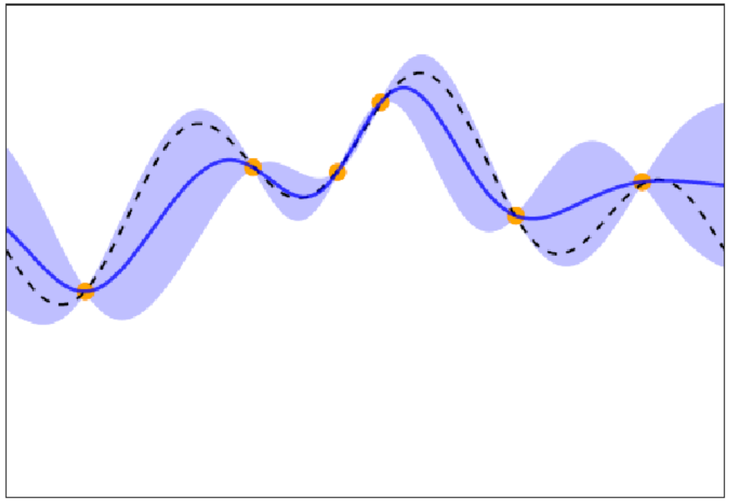
1: given ● existing data
fit a (probabilistic) model

2: induce your favourite utility

3: locate the maximum ○ of the
utility

4: observe ○ stochastic reward by
querying black-box

5: update model



SMBO IN A NUTSHELL

given the ability to query a black-box (- - -) repeat the following:

1: given ● existing data fit a (probabilistic) model

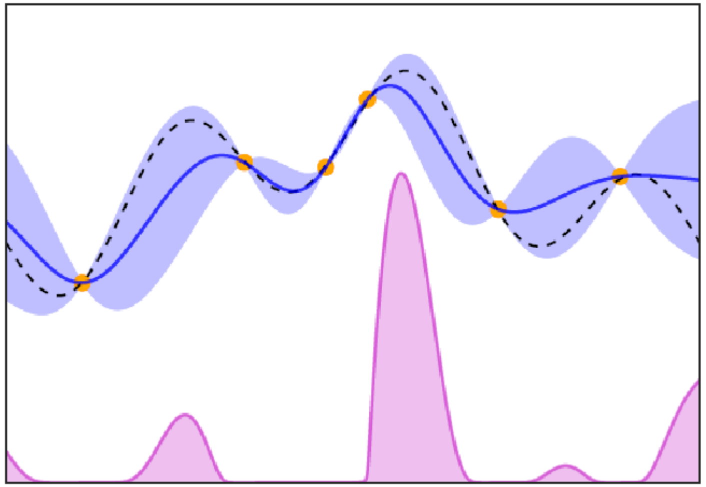
2: induce your favourite utility

3: locate the maximum ○ of the utility

4: observe ○ stochastic reward by querying black-box

5: update model

6: go to step 2...



All models are wrong but some are useful

BAYESIAN OPTIMIZATION & GAUSSIAN PROCESSES

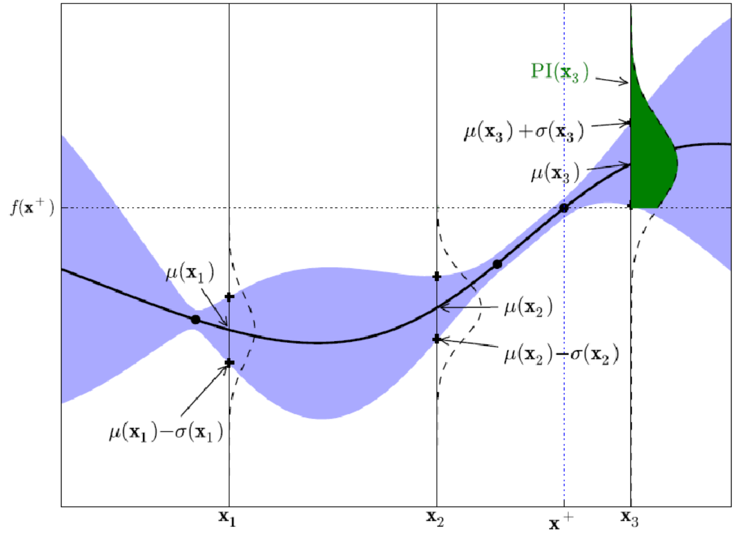


Figure 4: *Gaussian process from Figure 2, additionally showing the region of probable improvement. The maximum observation is at* x^+. *The darkly-shaded area in the superimposed Gaussian above the dashed line can be used as a measure of improvement,* $I(\mathrm{x})$. *The model predicts almost no possibility of improvement by observing at* x_1 *or* x_2, *while sampling at* x_3 *is more likely to improve on* $f(\mathrm{x}^+)$.

- ❑ A GP is a distribution over functions, completely specified by its mean function $\mu(x)$ and covariance (aka kernel) function $k(x, x')$

- ❑ A GP is analogous to a function, but instead of returning a scalar for an arbitrary x, it returns the mean and the variance of a normal ditribution over the possible value of $f(x)$

- ❑ BO uses a covariance function (aka kernel) which defines the covariance of any two function values *f(* x_i *)* and *f(* x_j *)* with x_i and x_j belonging to search space.

- ❑ A common choice is the squared exponential kernel:

$$k(x_i, x_j) = e^{-\frac{1}{2}\gamma \|x_i - x_j\|^2}$$

UPDATING THE GP

- The function evaluations can be affected by noise: $y_i = f(x) + \varepsilon$, where ε is i.i.d. Gaussian noise with variance λ.

- The set of function evaluations performed so far is :

$$D_{1:n} = \ \{x_i, y_i\}\ with\ i = 1, \dots, n$$

- The equations of $\mu_n(x)$ and $\sigma_n^2(x)$ are the following:

$$\mu_n(x) = \mathrm{k}(X_n, x)^T [K(X_n, X_n) + \mathrm{\lambda I}]^{-1} Y_n$$

$$\sigma_n^2(x) = k(x, x) - \mathrm{k}(X_n, x)^T [K(X_n, X_n + \mathrm{\lambda I})]^{-1} \mathrm{k}(X_n, x)$$

Issues in GP

- $X_n = \{x_1, \dots, x_n\}$ is the vector of evaluated points and $Y_n = \{y_1, \dots, y_n\}$ is the vector of observed noisy function evaluations

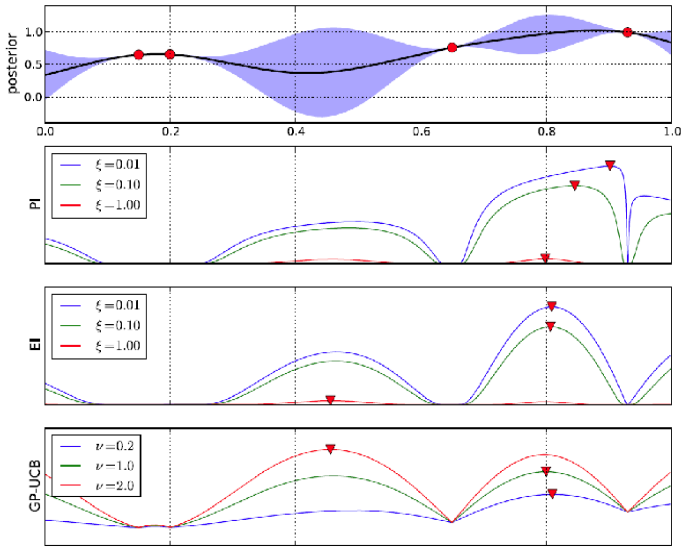
- The covariance matrix $K(X_n, X_n)$ has entries $[K(X_n, X_n)]_{i,j} = k(x_i, x_j)$, and the vector $\mathrm{k}(X_n, x) = \{k(x_1, x), \dots, k(x_n, x)\}$ contains the covariances between the new input x and the observed data points

- The actual convergence is disturbed by numerical instability problems in the inversion: $[K(X_n, X_n)]^{-1}$ (in the noise-free setting) due to near singularity of the matrix *K,* as *n* increases

- The introduction of noise, $[K(X_n, X_n) + \lambda I]^{-1}$, reduces the instability but induces a slower convergence

- The gradient of a GP is also a GP which opens the way to joint estimation of gradient and Hessian along with *f*. Quasi-Newton methods can be interpreted as particular instances of Gaussian regression

ACQUISITION FUNCTIONS



Built surrogate

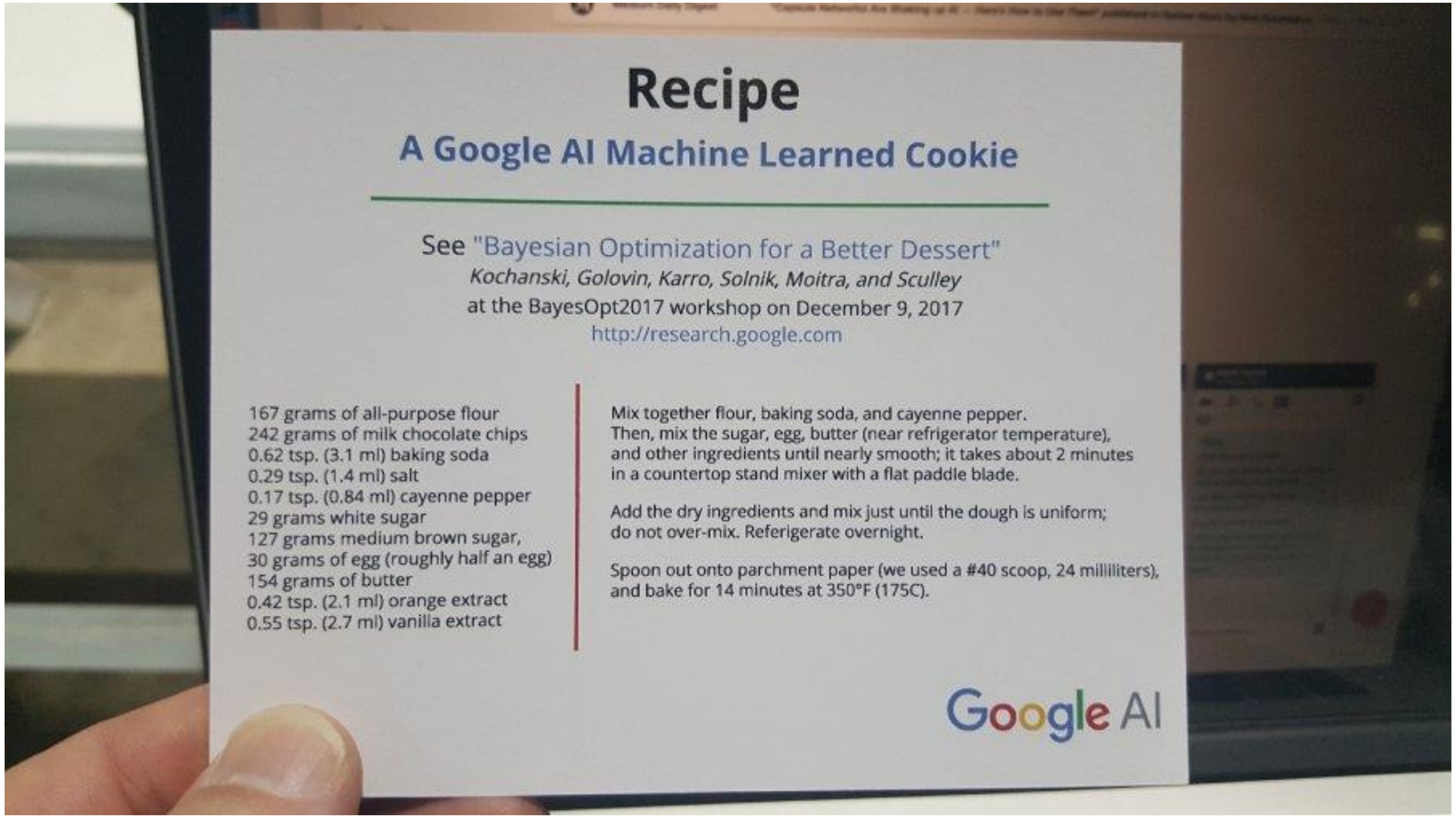
Probability of improvement (PI)
$PI(x) = P(f(x) \geq f(x^+) + \xi)$

Expected improvement (EI)
$EI(x) = \mathbb{E}(\max\{0, f(x) - f(x^+) - \xi\})$

Upper Confidence Bound (UCB)
$UCB(x) = \mathbb{E}f(x) + \nu\sigma(x)$

- new generation utility functions: Predictive Entropy Search, Knowlegde Gradient, Thomson sampling
- *Žilinskas, A., & Calvin, J. (2018). Bi-objective decision making in global optimization based on statistical models. Journal of Global Optimization, 1-11*

SUCCESS OF BAYESIAN OPTIMIZATION «THE MACHINE LEARNED COOKIE»



Recipe

A Google AI Machine Learned Cookie

See "Bayesian Optimization for a Better Dessert"
Kochanski, Golovin, Karro, Solnik, Moitra, and Sculley
at the BayesOpt2017 workshop on December 9, 2017
http://research.google.com

167 grams of all-purpose flour
242 grams of milk chocolate chips
0.62 tsp. (3.1 ml) baking soda
0.29 tsp. (1.4 ml) salt
0.17 tsp. (0.84 ml) cayenne pepper
29 grams white sugar
127 grams medium brown sugar,
30 grams of egg (roughly half an egg)
154 grams of butter
0.42 tsp. (2.1 ml) orange extract
0.55 tsp. (2.7 ml) vanilla extract

Mix together flour, baking soda, and cayenne pepper. Then, mix the sugar, egg, butter (near refrigerator temperature), and other ingredients until nearly smooth; it takes about 2 minutes in a countertop stand mixer with a flat paddle blade.

Add the dry ingredients and mix just until the dough is uniform; do not over-mix. Referigerate overnight.

Spoon out onto parchment paper (we used a #40 scoop, 24 milliliters), and bake for 14 minutes at 350°F (175C).

Google AI

HYPERPARAMETERS OPTIMIZATION & AUTOMATIC ALGORITHM CONFIGURATION

Corporate Solutions

- **SIGOPT** is a San Francisco based company offering BO, also hyperparameters optimization of ML algorithms, as a cloud-based service.
- **Google's Vizier** is an "internal" service now used in Hypertune, the Google's module to optimize hyperparameters of algorithms in Google's Machine Learning platform
- **Microsoft Azure** Hyperparameter tuning
- **Amazon SageMaker** Hyperparameter Optimization
- **SAS Autotune**

Open-source solutions

- **Spearmint**, **HyperOpt**, **AutoWEKA**
- **Sherpa** (NIPS 2018), offering Random Search, Grid Search, BO with GP and Population Based Tuning (PBT)
- **mlrMBO**, extensible framework for SMBO using GP as well as Random Forest

BO AS A GENERAL OPTIMIZATION TOOL

- Pump Scheduling Optimization in Water Distribution Networks
  - ➢ *Candelieri, A., Perego, R., & Archetti, F. (2018). Bayesian optimization of pump operations in water distribution systems. Journal of Global Optimization, 71(1), 213-235*
- Constrained BO
- Safe BO
  - *Candelieri A., Archetti F. Sequential Model Based Optimization with black-box constraints via Machine Learning based feasibility determination, Proceedings of LeGO - 14th Int'l Global Optimization Workshop [ahead of printing]*
- multi-objective BO
- BO and RL
  - *Candelieri, A., Perego, R., Archetti, F. (2018), Intelligent Pump Scheduling Optimization in Water Distribution Networks, in Proceedings of International Conference on Learning and Intelligent Optimization (LION) 2018 [ahead of printing]*

CONCLUDING REMARKS

- Smart/fast data and hyperparameter estimation opens up to optimization a vast set of new problems and entirely new setting of traditional fields

- Successful optimization methods will be defined not only in the lab but in the market as well. Data efficiency and flexibility to adapt to structurally different datasets and delivery platforms is already critical

- Optimization methods will have to interact with the environment i.e. incorporate some learning element, balancing exploration and exploitation and accepting uncertainty as the key modelling feature, as in dynamic/stochastic programming, BBO, BO and RL.

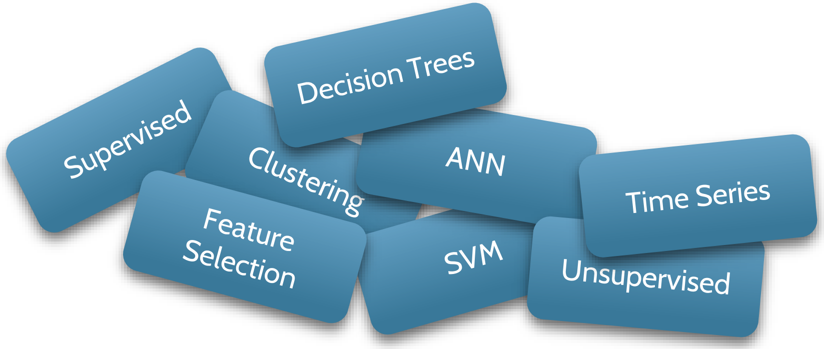
- We should not approach data analysis as a «cool science» experiment: the fundamental aim in collecting, analyzing and deploying data is to make better decisions in a context of bounded rationality and partial knowledge.

“SCIENCE IS BUILT UP WITH FACTS, AS A HOUSE IS WITH STONES. BUT A COLLECTION OF FACTS IS NO MORE A SCIENCE THAN A HEAP OF STONES IS A HOUSE”.

Henri Poincaré

“SCIENCE IS BUILT UP WITH FACTS, AS A HOUSE IS WITH STONES. BUT A COLLECTION OF FACTS IS NO MORE A SCIENCE THAN A HEAP OF STONES IS A HOUSE”.

Henri Poincaré

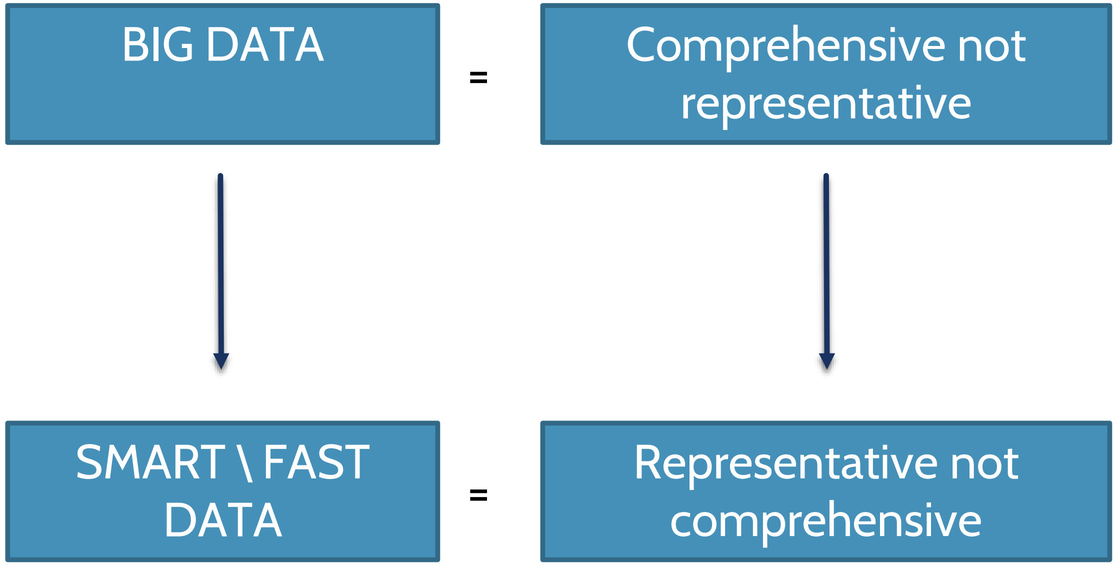


PREDICTIVE ANALYTICS: SMART & FAST DATA

- Water/energy consumption
- Traffic: number of vehicles per hour/minute
- Finance: daily closing price … or High Frequency Trading
- Manufacturing: vibration/energy consumption per minute driving the machine by sensor readings

- Forecasting and Anomaly Detection

- These data are characterized by different sources of uncertainty/variability



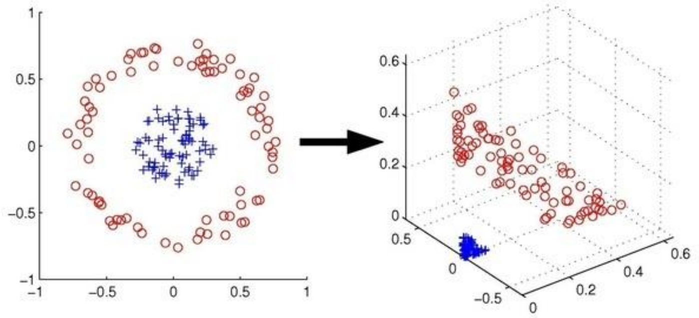
BIG DATA = Comprehensive not representative

SMART \ FAST DATA = Representative not comprehensive

CLUSTERING AS THE WORK HORSE OF ANALYTICS

- Clustering can be framed as an optimization problem which could be computationally hard
  - Fersini, E., Messina, E., & Archetti, F. (2014). A p-median approach for predicting drug response in tumour cells. *BMC bioinformatics*, *15*(1), 353.
- Relational data: points can be «linked» obtaining graphs/networks
- Considering the network structure allows us for defining new clustering algorithms and brings into action a new set of indices

Input Space



Feature Space