Overview on Automatic Tuning of Hyperparameters

Alexander Fonarev

http://newo.su

20.02.2016

Outline

- Introduction to the problem and examples
- Introduction to Bayesian optimization
- Overview of surrogate models
- Additional modifications
- Existing implementations

Hyperparameter examples

- Tree depth — decision trees
- Regularization coefficient — linear models
- Gradient descend step size — neural networks
- Normalization coefficient — data preprocessing

Examples of problems

- Training of the ranking in Yandex — days
- Parameters in NNs — tens

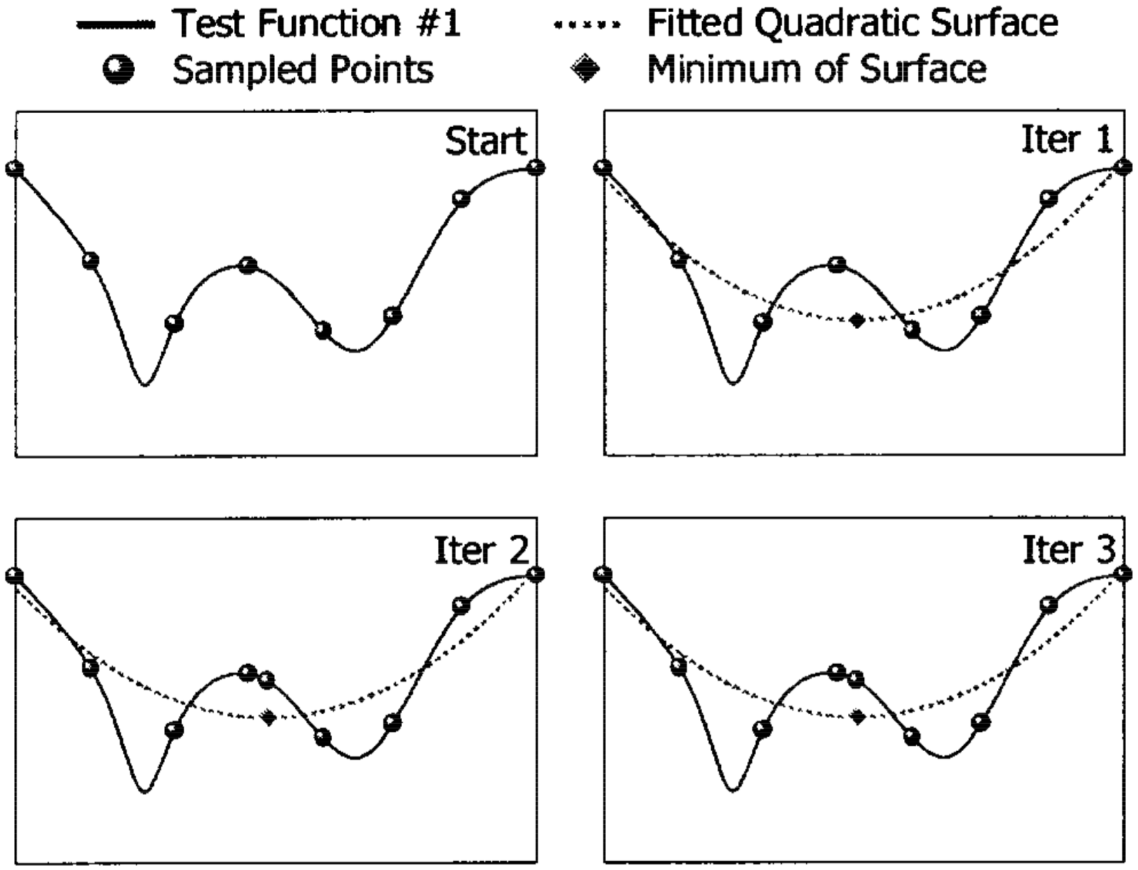
Popular and easy approaches

- Experience of experts
- Grid search
- **Random search**
- Manual coordinate-descend

"Random Search for Hyper-Parameter Optimization." James Bergstra and Yoshua Bengio. *Journal of Machine Learning Research*, 2012
https://www.oreilly.com/ideas/evaluating-machine-learning-models/page/5/hyperparameter-tuning

Gradient-free and global optimization

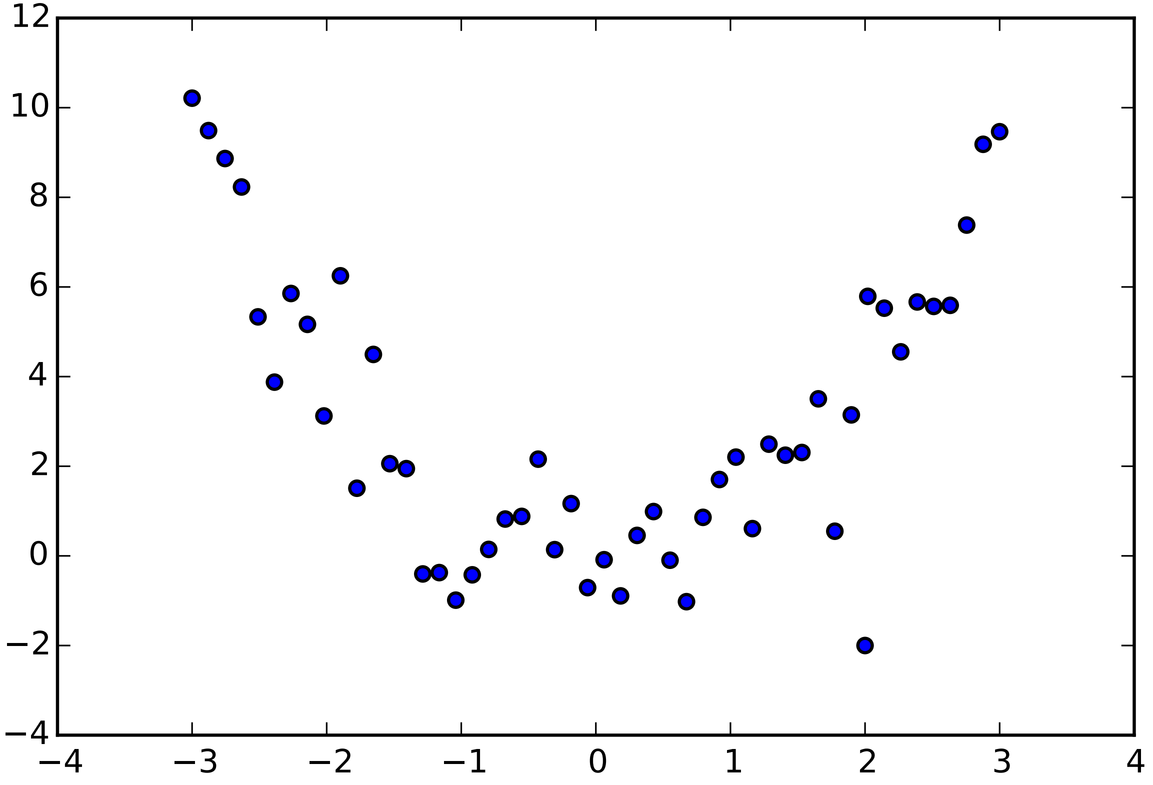
- Genetic algorithms
- Simulated annealing
- Response surfaces
- etc



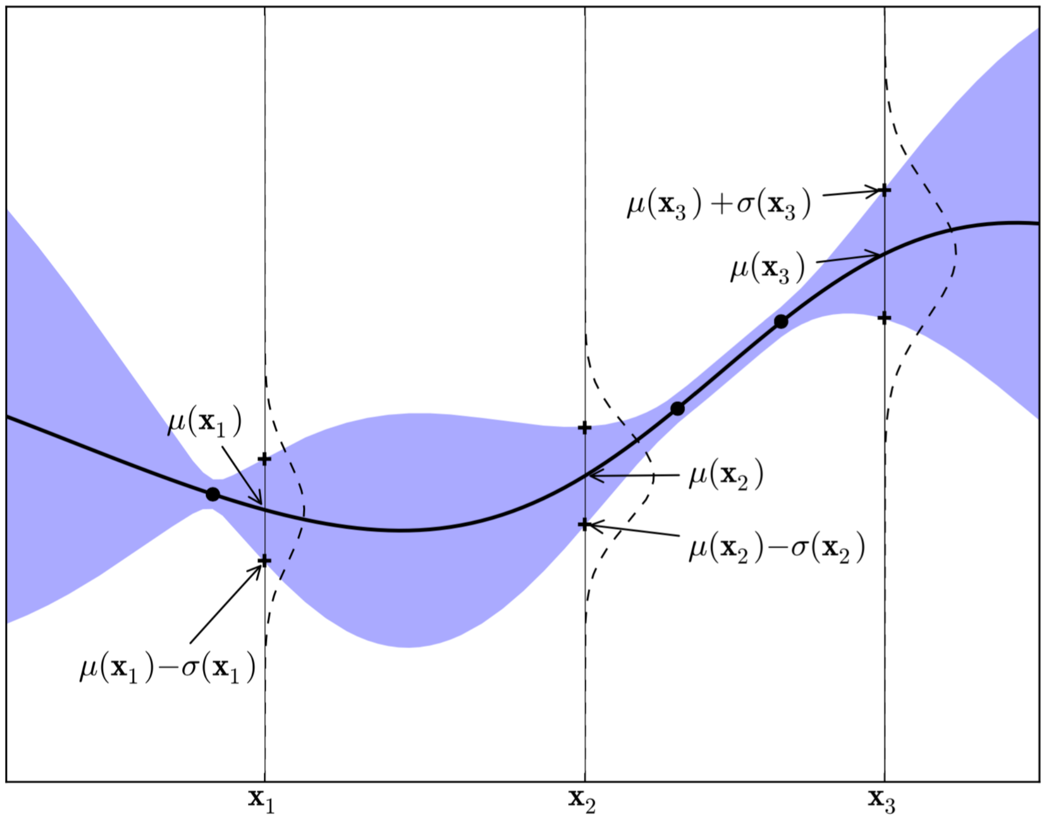
Response surfaces

Stochastic functions

How to optimize such functions?

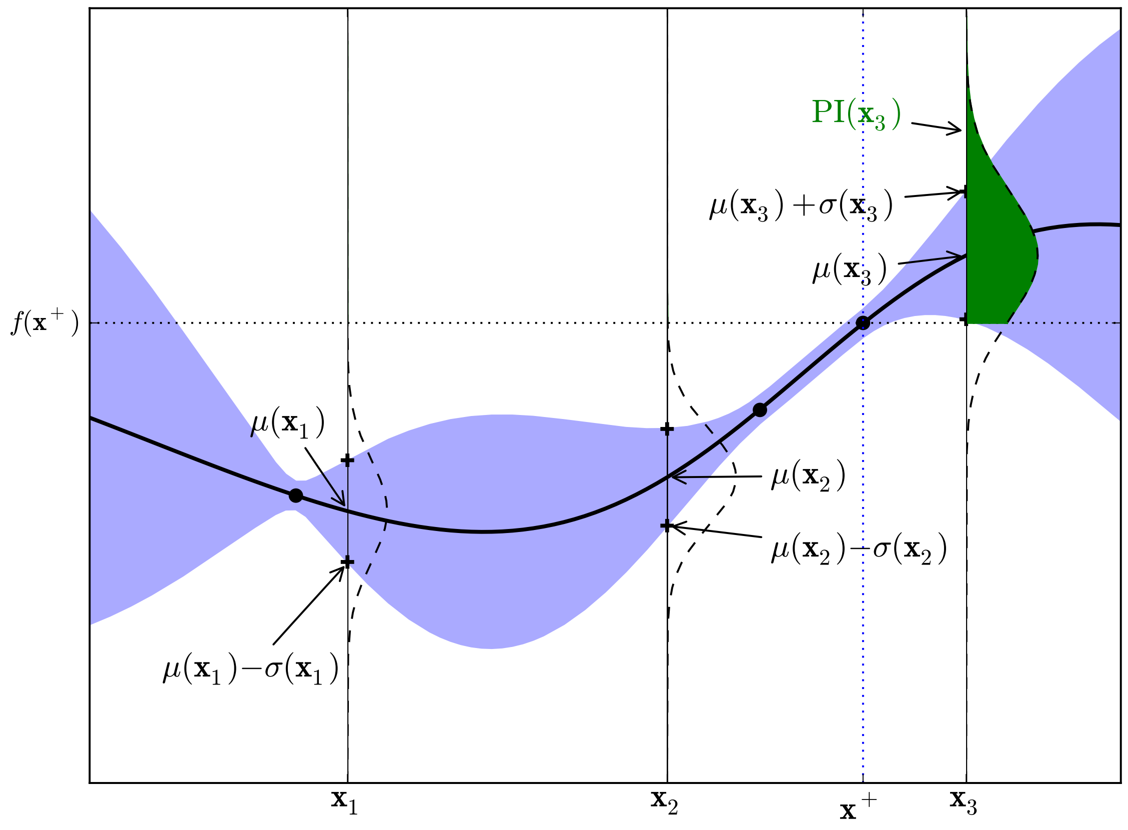


How to choose the next point to evaluate?

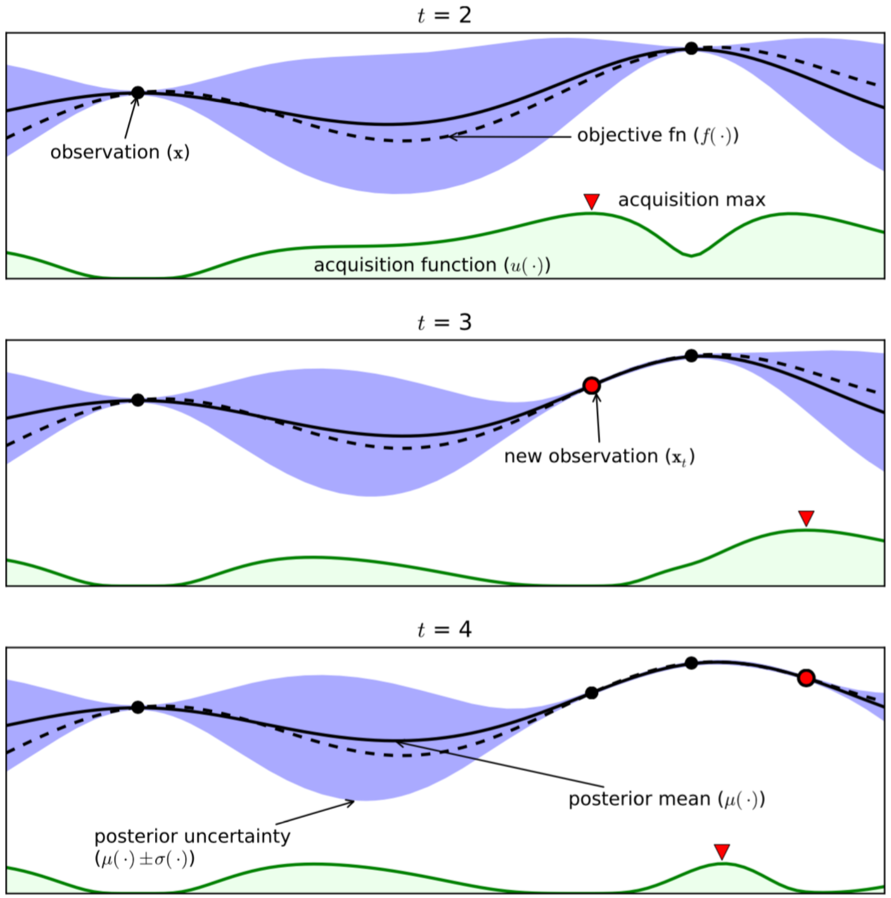


Brochu, Eric, Vlad M. Cora, and Nando De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning." *arXiv preprint arXiv:1012.2599* (2010).

Probability of improvement

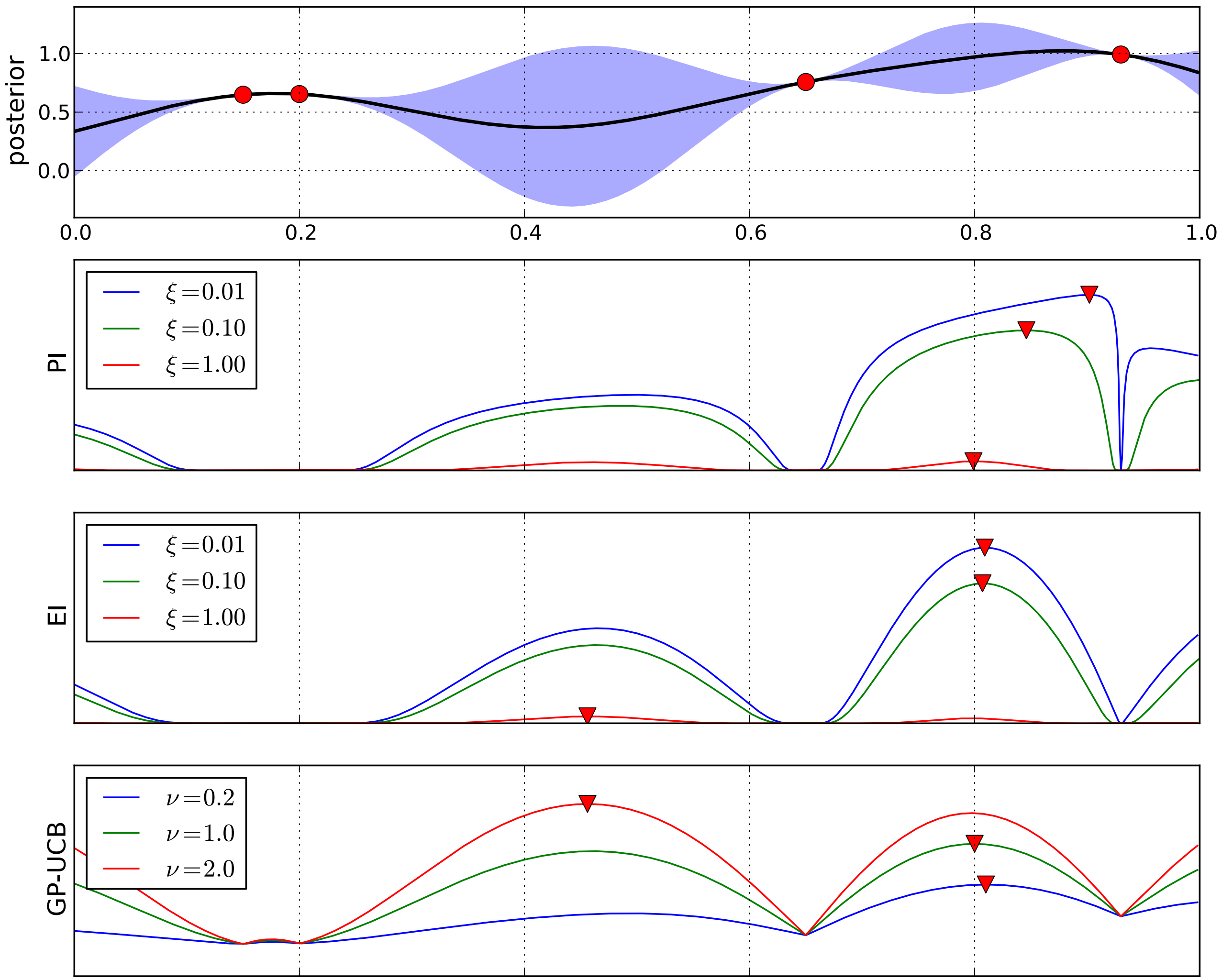


Acquisition function



Hutter, Frank, Jörg Lücke, and Lars Schmidt-Thieme. "Beyond Manual Tuning of Hyperparameters." 2015.
Shahriari, Bobak, et al. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." 2015.

Different acquisition strategies



Built surrogate

Probability of improvement (PI)

$PI(x) = P(f(x) \geq f(x^{+}) + \xi)$

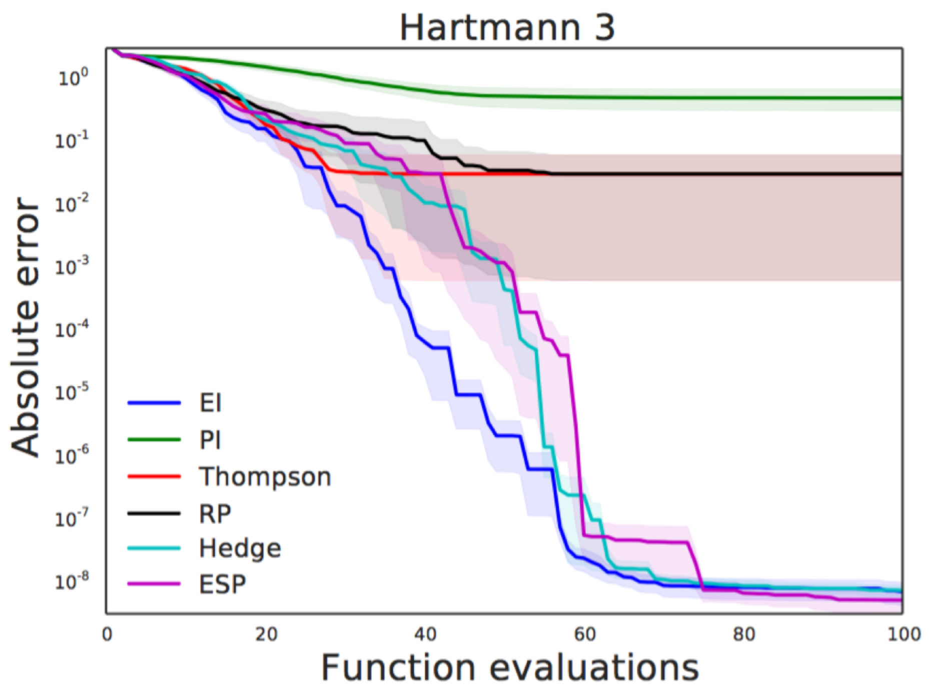
Expected improvement (EI)

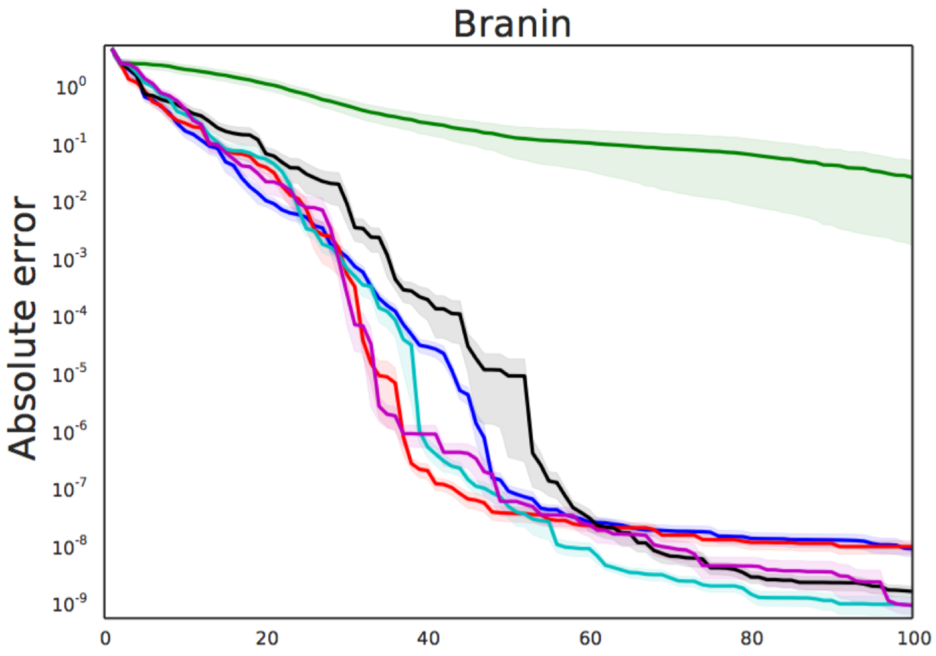
$EI(x) = \mathbb{E}(\max\{0, f(x) - f(x^{+}) - \xi\})$

Upper Confidence Bound (UCB)

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

Comparison of acquisition functions





Shahriari, Bobak, et al. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." 2015.

Common algorithm

1. Initial design — evaluate the black box in some points
2. Adaptive design
   a) Build a surrogate
   b) Find the argmax of Expected Improvement
   c) Evaluate the black box in this point
   d) Go to step 2

How to perform the initial design?

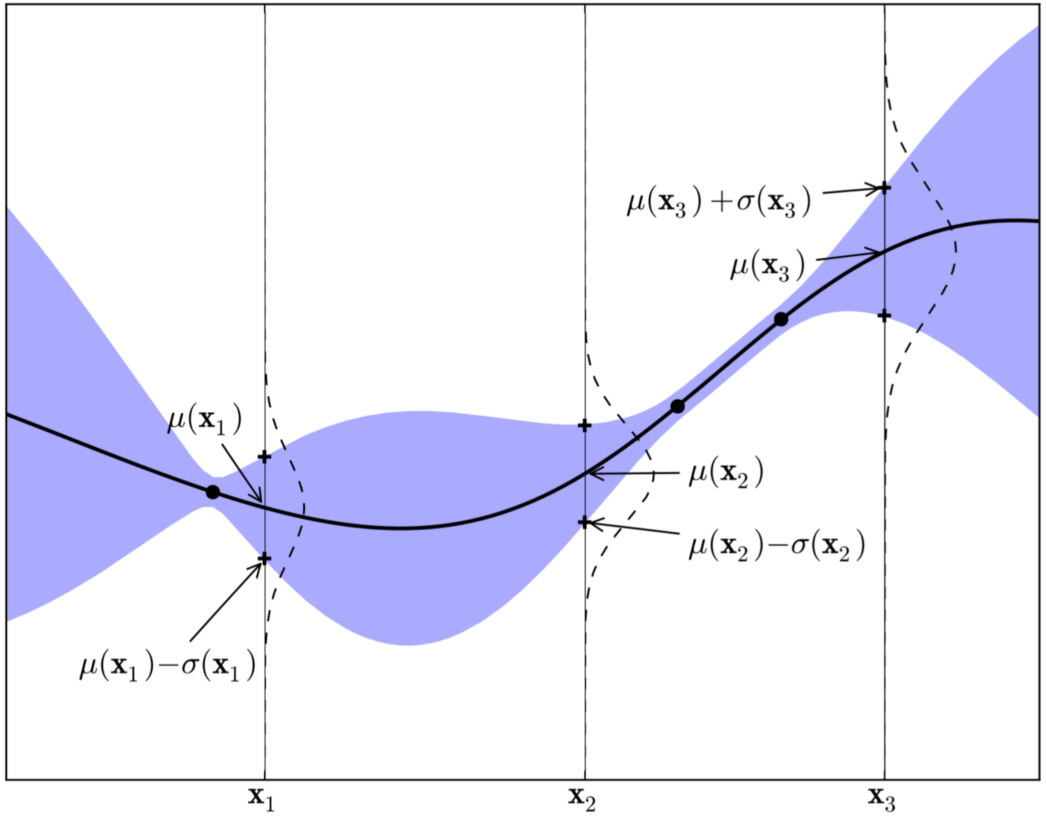
- Best from previous experiments
- Ask for experts
- Random
- Grid
- Several optimal design criteria

https://en.wikipedia.org/wiki/Optimal_design

Types of surrogates

- Gaussian Processes (GP)
- Tree of Parzen Estimators (TPE)
- Sequential Model-based Algorithm Configuration (SMAC)

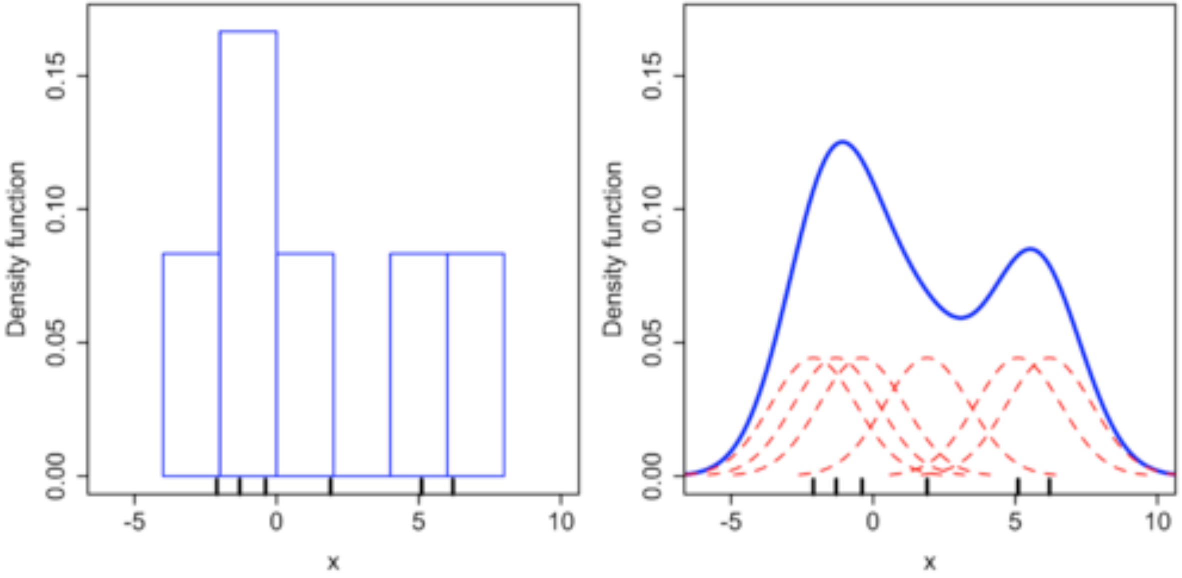
Gaussian Process (GP)



http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf
https://www.youtube.com/watch?v=4vGiHC35j9s
https://github.com/JasperSnoek/spearmint (Spearmint)

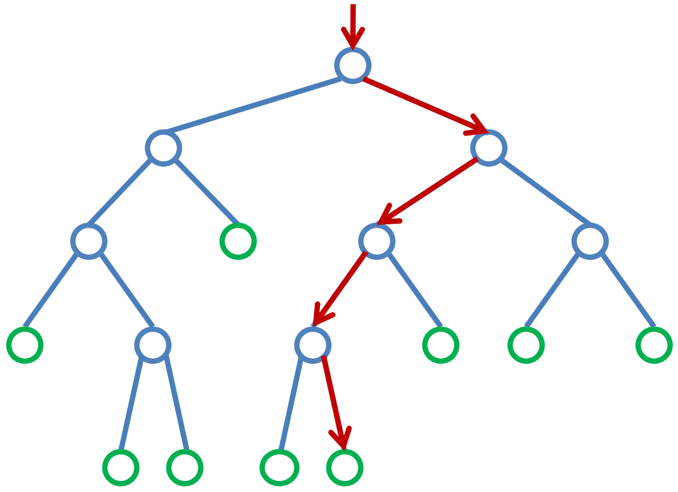
Tree of Parzen Estimators (TPE)

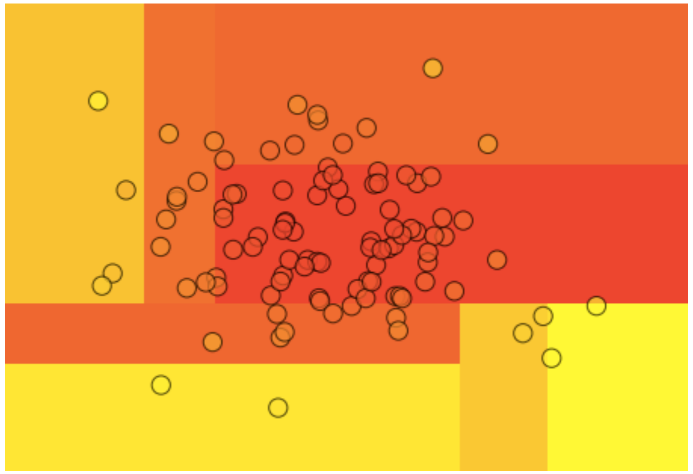
Estimates $p(y)$ and $p(x|y)$ instead of $p(y|x)$. The core idea — nonparametric density approximations of *x*.



http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf
https://github.com/hyperopt/hyperopt (Hyperopt)

Reminder: Random Forest

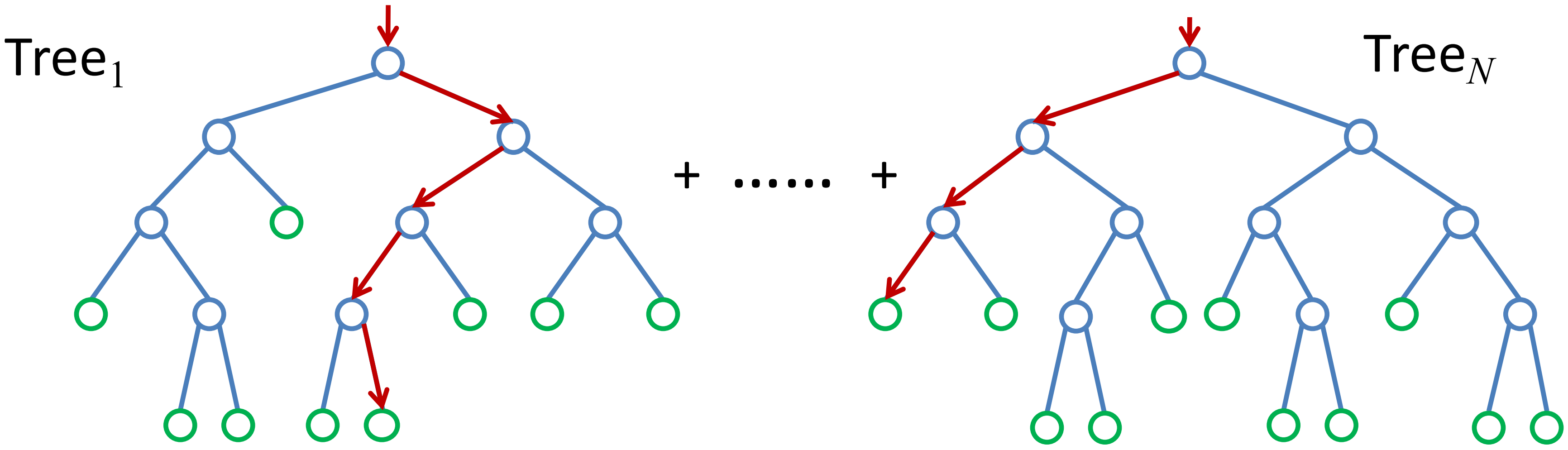




Not robust (has high variance)

Reminder: Random Forest

Average of slightly different trees:



https://www.youtube.com/watch?v=3kYujfDgmNk

Sequential Model-based Algorithm Configuration (SMAC)

- Mean in each point — prediction of RF
- Variance in each point — variance of predictions of separate trees from RF

https://github.com/automl/pysmac (SMAC)

How to find the maximum of EI?

Use global optimization. Some options:

- Random search
- Genetic algorithms
- Simulated annealing
- Response surfaces

Parallelization

- Batch
- Asynchronous adaptive design

http://javad-azimi.com/index_files/Nips2010.pdf
http://arxiv.org/pdf/1202.5597.pdf
https://bayesopt.github.io/papers/2015/gonzalez-batch.pdf

What else?

- Data structures for fast optimum search
- Cold start problem
- Different parameters require different learning time
- Labels transformation
- Different parameter types

Open source implementations

Package	License	URL	Language	Model
SMAC	Academic non-commercial license.	http://www.cs.ubc.ca/labs/beta/Projects/SMAC	Java	Random forest
Hyperopt	BSD	https://github.com/hyperopt/hyperopt	Python	Tree Parzen estimator
Spearmint	Academic non-commercial license.	https://github.com/HIPS/Spearmint	Python	Gaussian process
Bayesopt	GPL	http://rmcantin.bitbucket.org/html	C++	Gaussian process
PyBO	BSD	https://github.com/mwhoffman/pybo	Python	Gaussian process
MOE	Apache 2.0	https://github.com/Yelp/MOE	Python / C++	Gaussian process

Hutter, Frank, Jörg Lücke, and Lars Schmidt-Thieme. "Beyond Manual Tuning of Hyperparameters.” 2015.

How to evaluate method?

- Use real data
- Use synthetic data

http://infinity77.net/global_optimization/test_functions.html#test-functions-index
https://github.com/andyfaff/ampgo

Comparison of different approaches

Experiment	#evals	SMAC Valid. loss	SMAC Best loss	Spearmint Valid. loss	Spearmint Best loss	TPE Valid. loss	TPE Best loss
branin (0.398)	200	0.655±0.27	0.408	**0.398**±0.00	**0.398**	0.526± 0.13	0.422
har6 (-3.322)	200	-2.977±0.11	-3.154	**-3.133**±0.41	**-3.322**	-2.823±0.18	-3.039
Log.Regression	100	8.6±0.9	7.7	**7.3**±0.2	**7.0**	8.2±0.6	7.5
LDA ongrid	50	**1269.6**±2.9	**1266.2**	1272.6±10.3	**1266.2**	1271.5±3.5	**1266.2**
SVM ongrid	100	**24.1**±0.1	**24.1**	24.6±0.9	**24.1**	24.2±0.0	**24.1**
HP-NNET convex	100	**19.5**±1.5	**17.0**	20.6±0.3	20.1	**19.5**±1.6	17.4
HP-NNET convex	200	**18.3**±1.9	**15.2**	20.0±0.9	17.3	18.5±1.4	16.2
HP-NNET MRBI	100	51.5±2.8	**46.1**	52.2±3.3	46.5	**50.0**±1.7	47.3
HP-NNET MRBI	200	**48.3**±1.80	**46.1**	51.4±3.2	46.5	48.9±1.4	46.9
HP-DBNET convex	100	**16.4**±1.2	**14.5**	20.74±6.9	15.5	17.29±1.7	15.3
HP-DBNET convex	200	**15.4**±0.8	**14.0**	17.45±5.6	14.6	16.1±0.5	15.3
Auto-WEKA	30h	**27.5**±4.9	**22.3**	40.64±7.2	31.9	35.5±2.9	28.8
Log.Regression 5CV	500 folds	**8.1**±0.2	**7.8**	8.2±0.1	7.9	8.9±0.5	8.1
HP-NNET convex 5CV	500 folds	**18.2**±1.5	**16.9**	23.0±5.0	19.7	20.9±1.3	18.6
HP-NNET MRBI 5CV	500 folds	**47.9**±0.7	47.2	52.8±5.1	**46.6**	50.8 ±1.4	48.2

http://www.cs.ubc.ca/~hutter/papers/13-BayesOpt_EmpiricalFoundation.pdf

Terminology

- Bayesian optimization
- Reinforcement learning
- Surrogate model
- Kriging

What else?

- Data structures for fast optimum search
- Cold start problem
- Different parameters require different learning time
- Labels transformation

Summary

- We usually need to optimize stochastic functions
- Surrogate model should be fit to the data
- Several good implementations already exist
- Random search is much better than you thought!

Thank you!

Alexander Fonarev

http://newo.su