Presentation by Arthur Grabovsky

Discovering and Exploiting Additive Structure for Bayesian Optimization

Jacob R. Gardner
Department of Computer Science
Cornell University
jrg365@cornell.edu

Chuan Guo
Department of Computer Science
Cornell University
cg563@cornell.edu

Kilian Q. Weinberger
Department of Computer Science
Cornell University
kqw4@cornell.edu

Roman Garnett
Computer Science and Engineering
Washington University in St. Louis
garnett@wustl.edu

Roger Grosse
Department of Computer Science
University of Toronto
rgrosse@cs.toronto.edu

Hyperparameter Search

- Most methods in machine learning require hyperparameters
  - Regularization parameters for linear regression, neural network layers, neighbors in kNN, maximum tree depth, etc.

- Performance can crucially depend on their values – think unregularized linear regression with 100,000 predictors or kNN with k = n

- Hyperparameters need to be set properly for optimal or even acceptable performance

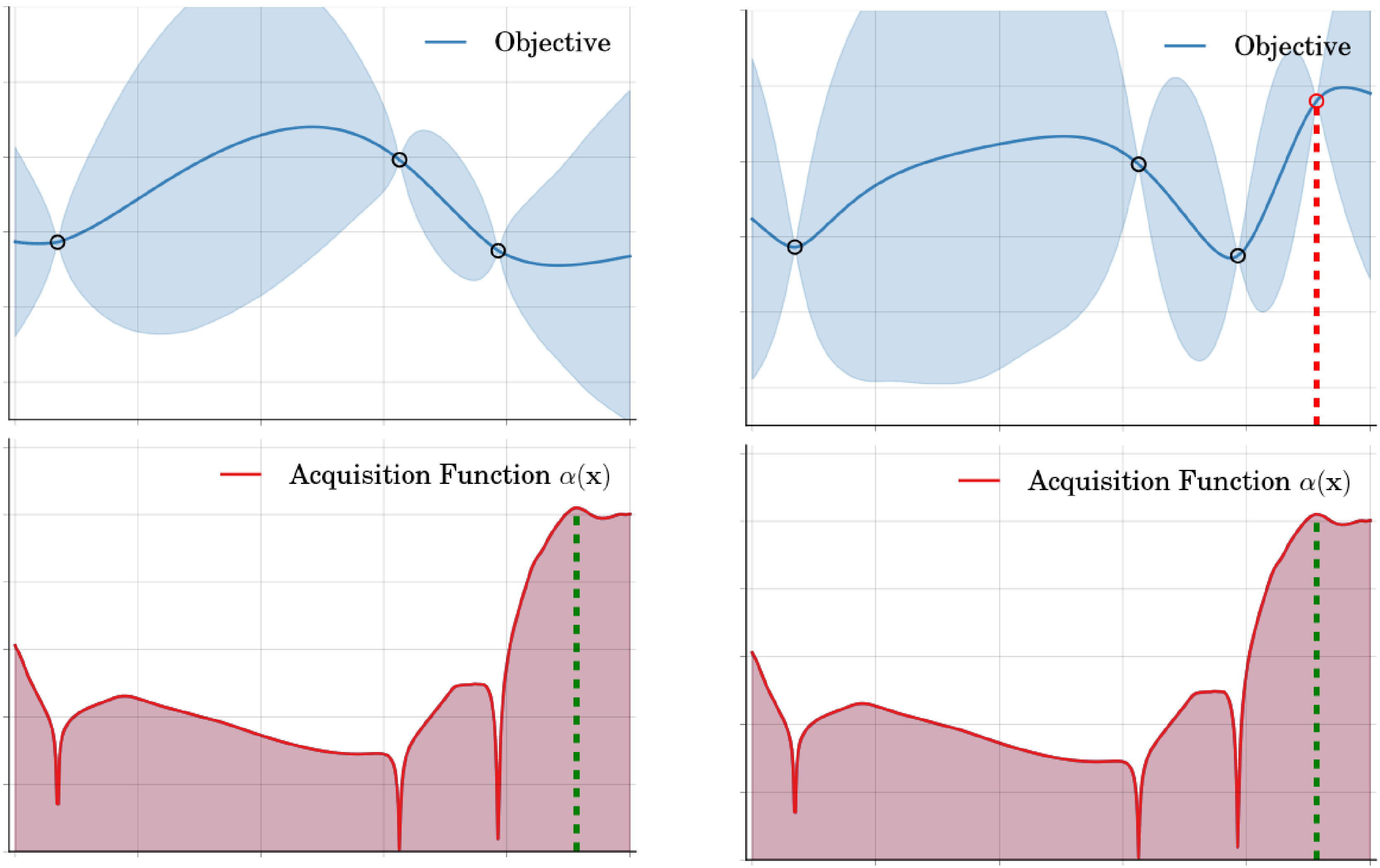
Difficulties with hyperparameter optimization

- Objective function unknown, no gradients, really expensive to evaluate

Typical solutions

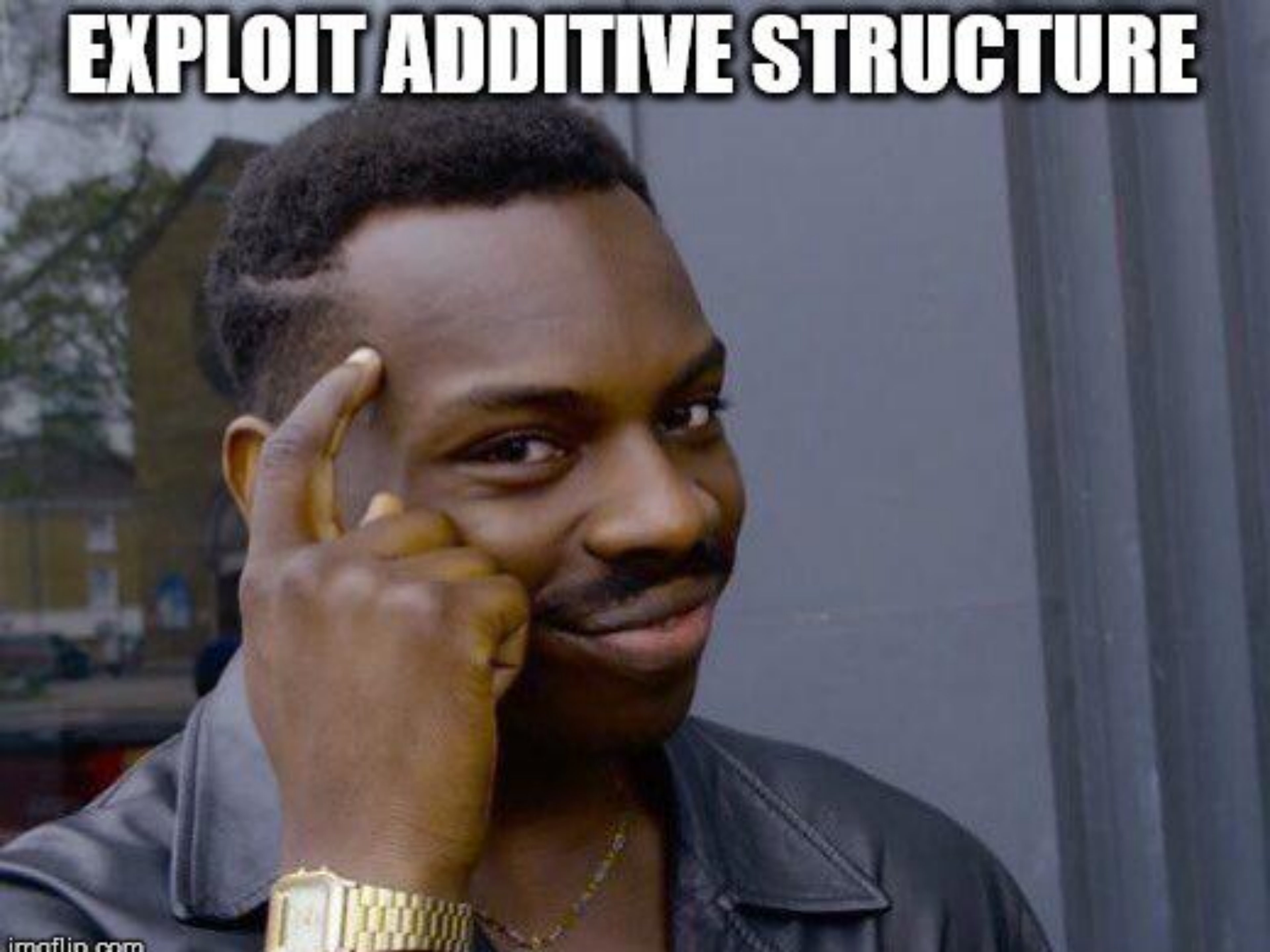
- Grid search, random search, Bayesian optimization

Bayesian Optimization



Bayesian Optimization

Pros	Cons
Smarter decisions lead to faster convergence	Implementation is not easy
bigml rapidminer DataRobot Used in practice PredictionIO™ Wise.io	Dependent on own hyperparameters
	KEY ISSUE Can't really be used in high dimension (exponential complexity) What to do?


EXPLOIT ADDITIVE STRUCTURE

Objective Function Structure Types

Structure	Example	Complexity
Fully Dependent	$f(x) = x_1 x_2 x_3 x_4 x_5$	Exponential
Fully Independent	$f(x) = x_1 + x_2 + x_3 + x_4 + x_5$	Linear
Mixed	$f(x) = x_1 x_2 x_3 + x_4 + x_5$	Subexponential

Knowing additive structure gives exponential reduction in complexity
(Kandasamy et. al 2015)

Bayesian Optimization Flow

1. Get initial sample from objective function
2. Update posterior (refit kernel)
3. Optimize acquisition function
4. Sample objective function at point x^*
5. Repeat until satisfied

Bayesian Optimization Flow, Structure Discovery

1. Get initial sample from objective function
2. Discover objective function structure
   - 2.1 Sample model (partition)

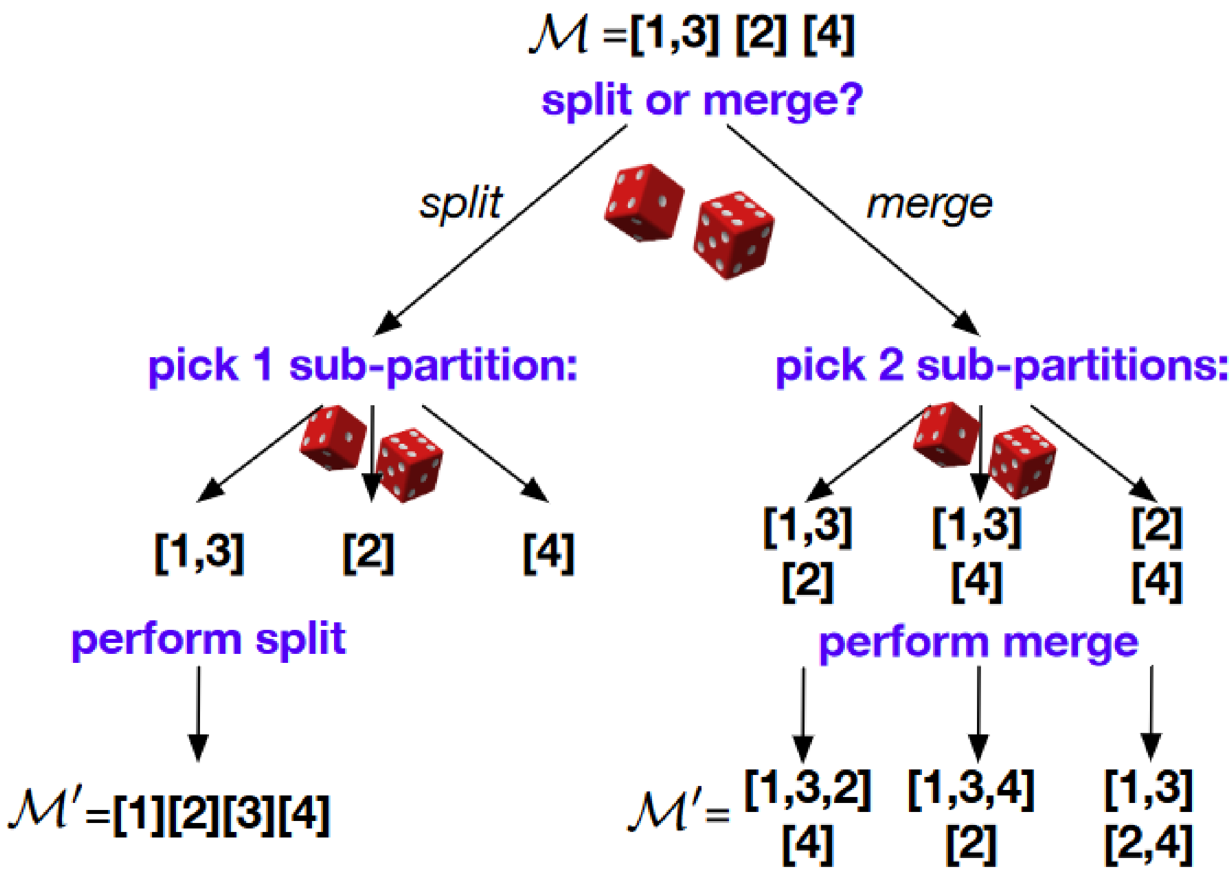
     $M_k = [1,2,3][4][5]$
   - 2.2 Fit additive kernel

     $K_k = K(x_{123}, x_{123}) + K(x_4) + K(x_5)$
   - 2.3 Optimize acquisition function for x^*_k
   - 2.4 Repeat k times (50 in the paper)
3. Set x^* to be the point from $(x_1, \ldots, x_k)$ that maximizes marginalized acquisition function

   $$p(f(\mathbf{x}^*) \mid \mathcal{D}, \mathbf{x}^*) \approx \frac{1}{k} \sum_{j=1}^{k} p(f(\mathbf{x}^*) \mid \mathcal{D}, \mathbf{x}^*, \mathcal{M}_j)$$
4. Sample objective function at point x^*
5. Repeat until satisfied

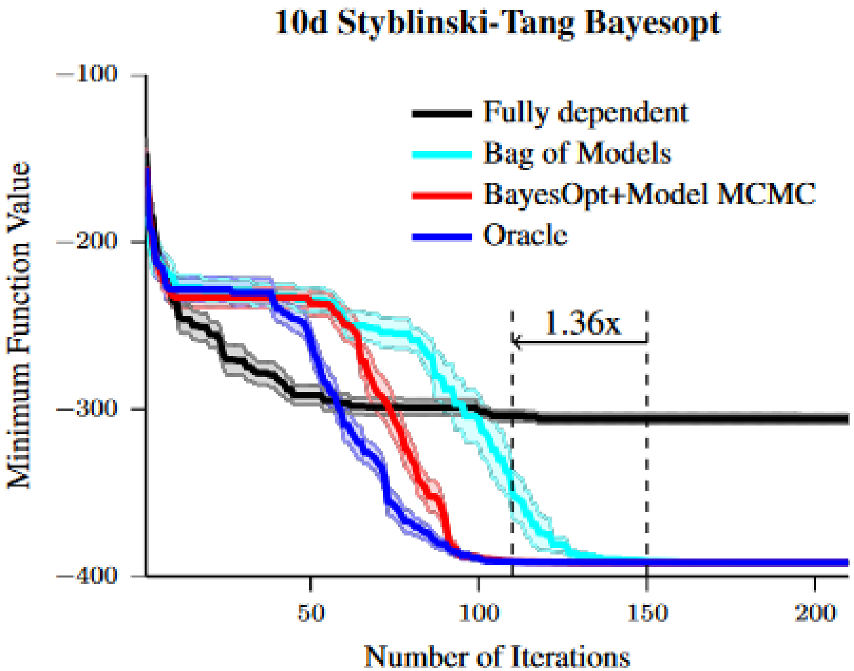
Metropolis-Hastings Model Sampling

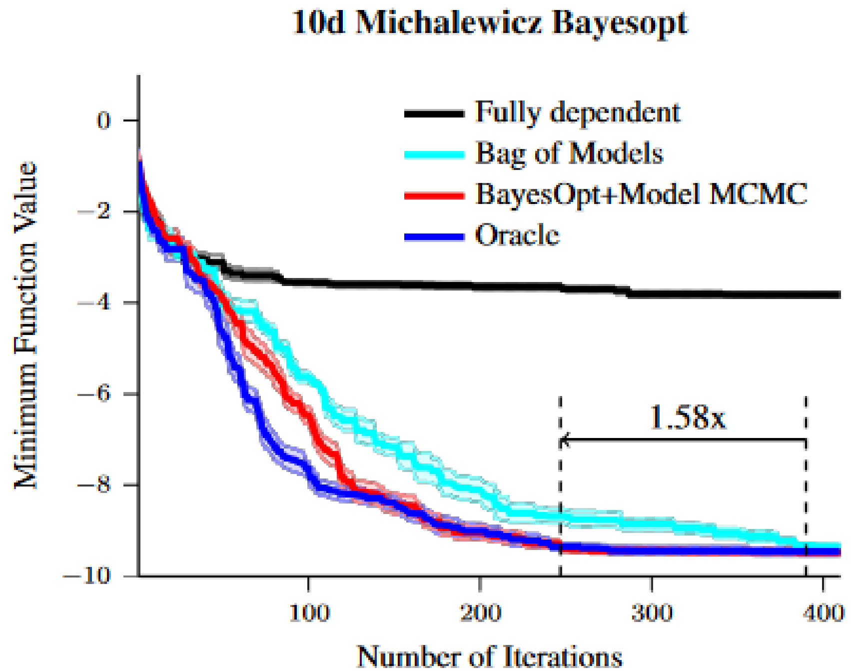
1. Sample from proposal distribution



2. Accept sample with probability $A(\mathcal{M}' \mid \mathcal{M}_j) = \min\left(1, \frac{p(\mathbf{y}_i \mid \mathbf{X}_i, \mathcal{M}')g(\mathcal{M}_j \mid \mathcal{M}')}{p(\mathbf{y}_i \mid \mathbf{X}_i, \mathcal{M}_j)g(\mathcal{M}' \mid \mathcal{M}_j)}\right)$

Results, Simulation

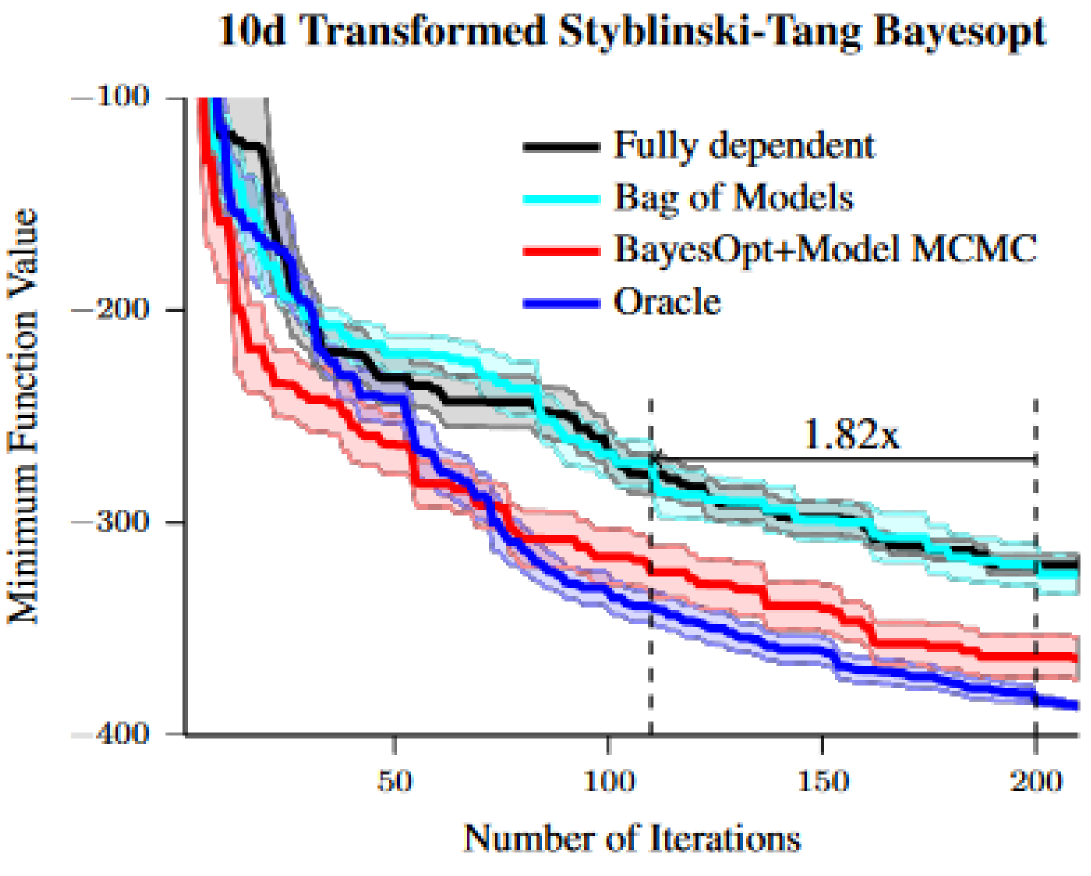




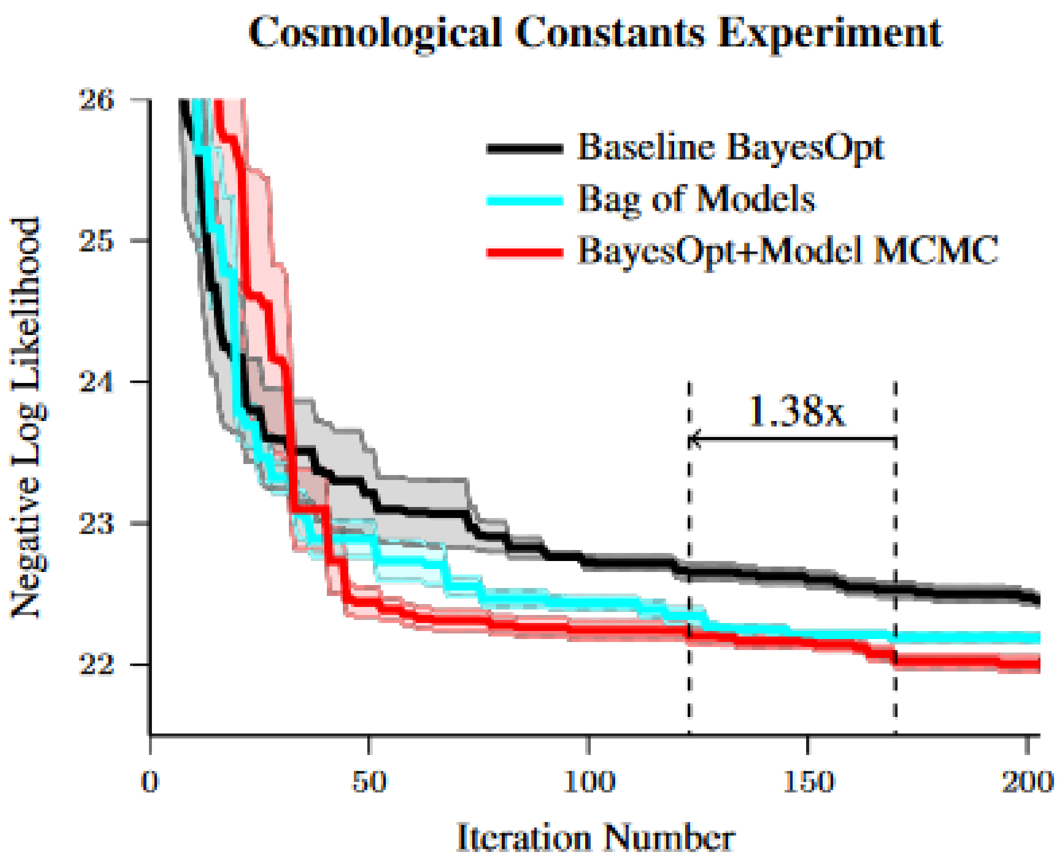
$$\text{Stybtang}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d} x_i^4 - 16x_i^2 + 5x_i$$

$$\text{Michalewicz}(\mathbf{x}) = -\sum_{i=1}^{d} \sin(x_i)\sin^{2m}\left(\frac{ix_i}{\pi}\right)$$

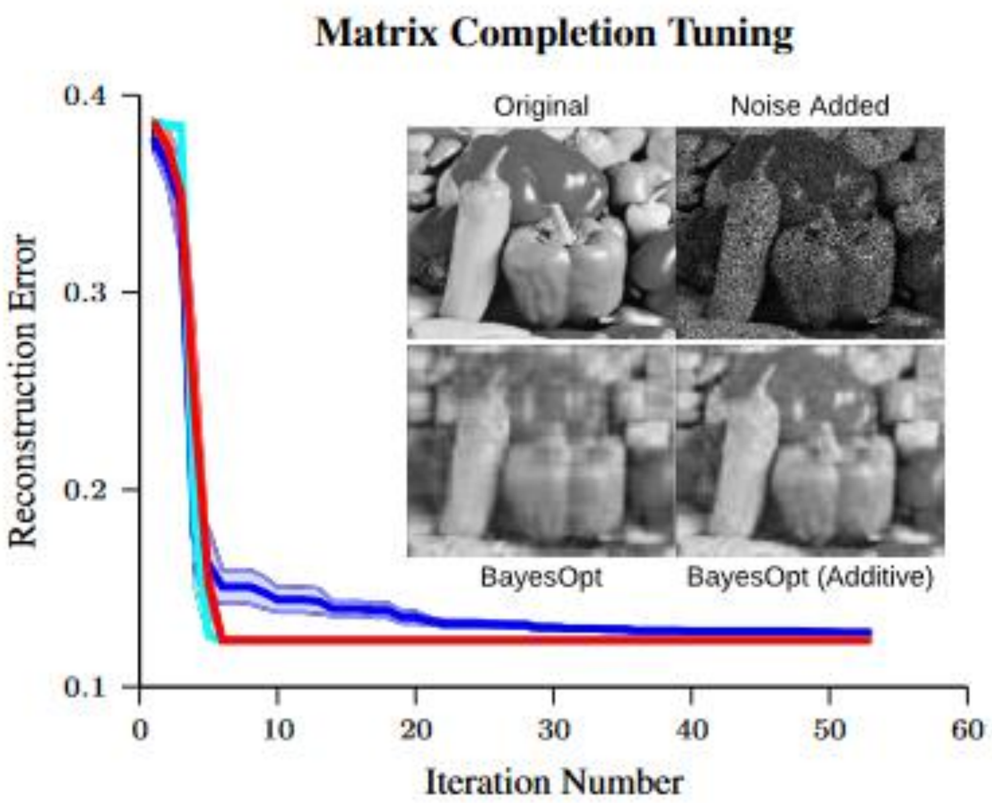
Results, Simulation



Results, Real Data



Results, Real Data



Conclusion

- Bayesian optimization can select optimal hyperparameter settings with fewer iterations
- …but is very slow in high dimensions (over 100 hyperparameters)
- One possible solution – exploit additive structure
- Works very well when additive structure is present, not much worse when it isn't
- Can be a powerful extension to Auto ML applications
- Not free - if the objective function is not too expensive this can be slower
  - Need to evaluate k extra models but each model simpler
- Doesn't solve all the problems – high dimensionality still a problem, but now less so