BAYESIAN OPTIMIZATION FOR AUTOMATED MODEL SELECTION

Gustavo Malkomes
Chip Schaff
Roman Garnett
Washington University in St. Louis

Probabilistic Scientific Computing

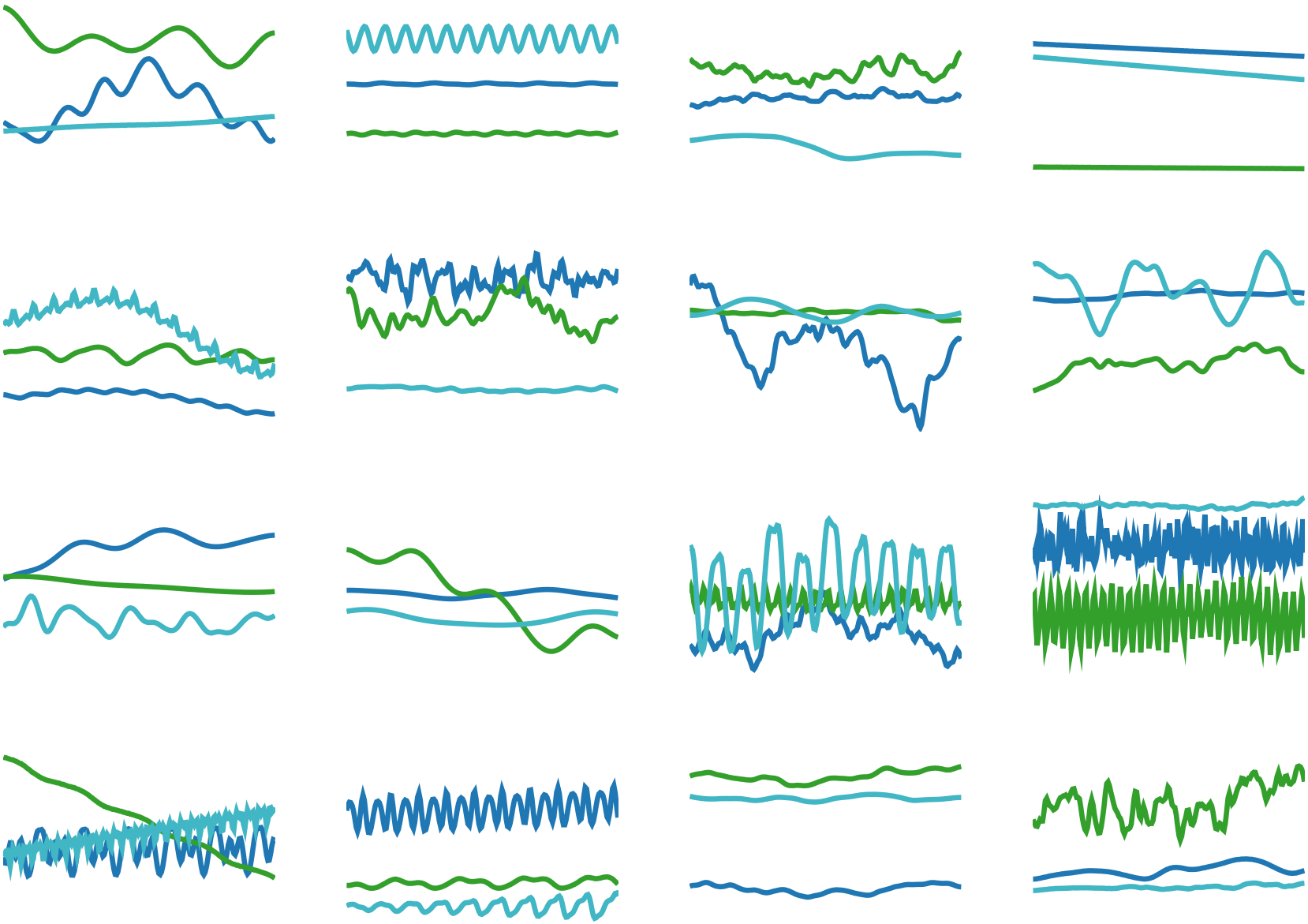
06.06.2017

INTRODUCTION

GP Model selection

Problem

- Gaussian processes (GPs) are powerful models able to express a wide range of structure in nonlinear functions.
- This power is sometimes a *curse,* as it can be very difficult to determine appropriate *models* (e.g., mean/covariance functions) to describe a given dataset.
- The choice of model can be *critical!* ...How would a nonexpert make this choice? (usually blindly!)
- Our goal here will be to *automatically* construct a useful model to explain a given dataset.

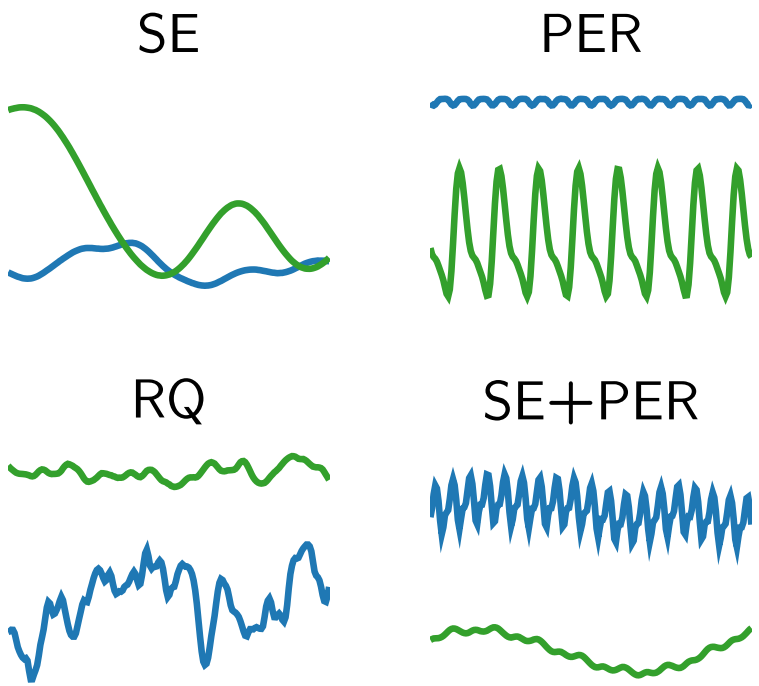


Simple grammar[1]

$K \mapsto \{\text{SE, RQ, LIN, PER, } \ldots\}$

$K \mapsto K * K$

$K \mapsto K + K$



[1] Duvenaud, et al. ICML 2013

The problem

We want to *automatically* search a space of GP models (i.e., parameterized mean/covariance functions with priors over their parameters)

$$\mathbb{M} = \{\mathcal{M}\}$$

to find the *best* one to explain our data.

Objective function

In the Bayesian formalism, given a dataset $\mathcal{D}$, we measure the quality of a model $\mathcal{M}$ using the (log) *model evidence,* which we wish to maximize:

$$g(\mathcal{M}; \mathcal{D}) = \log \int p(\mathbf{y} \mid \mathbf{X}, \theta, \mathcal{M}) p(\theta \mid \mathcal{M}) \, \mathrm{d}\theta$$

This is *intractable,* but we can approximate, e.g.:

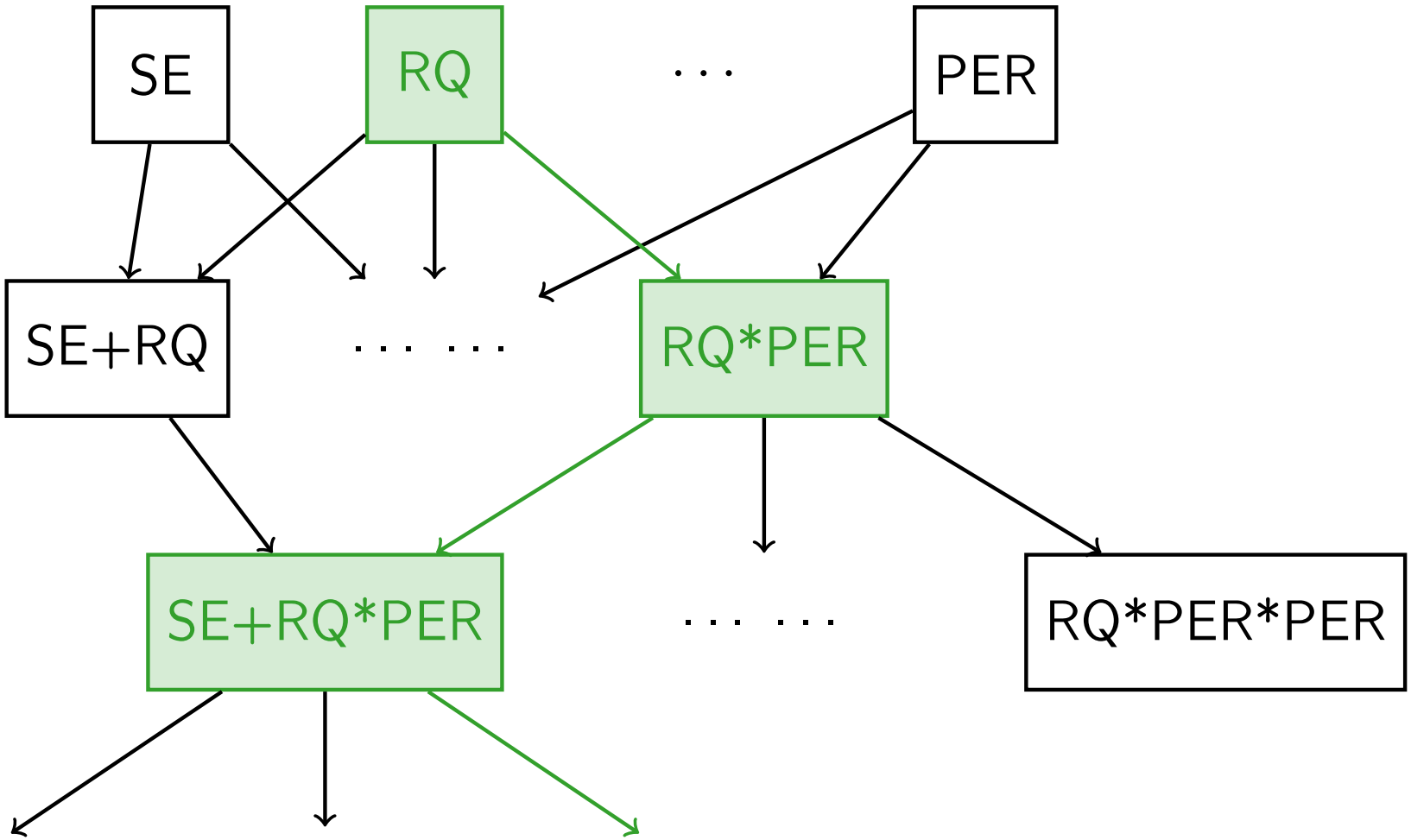
- Bayesian information criterion (BIC)
- Laplace approximation

Optimization problem

We may now frame the model search problem as an *optimization problem.* We seek

$$\mathcal{M}^* = \underset{\mathcal{M} \in \mathbb{M}}{\arg\max}\, g(\mathcal{M}; \mathcal{D}).$$

Previous work: Greedy search[2]



[2]Duvenaud, et al., ICML 2013

OBSTACLES

Why this is a hard problem

The objective is nonlinear and nonconvex

- The mapping from models to evidence is *highly complex!*
- Even seemingly "similar" models can offer *vastly different* explanations of the data.
- . . . and this similarity depends on the *geometry* of the data!
- Imagine a bunch of isolated points. . .

The objective is expensive

Even estimating the model evidence is *very expensive.* Both the BIC and Laplace approximations require finding the MLE/MAP hyperparameters:

$$\hat{\theta}_{\mathcal{M}} = \arg\max_{\theta} \log p(\mathbf{y} \mid \mathbf{X}, \theta, \mathcal{M})$$

This can easily be $\mathcal{O}(1000N^3)$!

The domain is discrete

Another problem is that the space of models is *discrete;* therefore we can't compute *gradients* of the objective.

BAYESIAN OPTIMIZATION?

Why not?

A case for Bayesian optimization!

We have a

- nonlinear,
- gradient-free,
- expensive,
- black-box optimization problem...

... *Bayesian optimization!*

Overview of approach

We are going to model the (log) model evidence function with a *Gaussian process in model space:*

$$g \colon \mathbb{M} \to \log p(\mathbf{y} \mid \mathbf{X}, \mathcal{M})$$

$$p\big(g(\mathcal{M}; \mathcal{D})\big) = \mathcal{GP}(g; \mu_g, K_g).$$

(How are we going to construct this??)

Overview of approach

Given some observed models and their evidences:

$$\mathcal{D}_g = \Big\{ \big(\mathcal{M}_i, g(\mathcal{M}_i; \mathcal{D}) \big) \Big\},$$

We find the posterior $p(g \mid \mathcal{D}_g)$ and derive an *acquisition function*

$$\alpha(\mathcal{M}; \mathcal{D}_g)$$

that we *maximize* to select the next model for investigation.

(How are we going to maximize this??)

THE EVIDENCE MODEL

Evidence model: mean

We need to construct an *informative prior* over the log model evidence function:

$$p\big(g(\mathcal{M};\mathcal{D})\big) = \mathcal{GP}(g;\mu_g,K_g).$$

For the mean, we simply take a constant...

...what about the covariance?

The "kernel kernel"

The covariance K_g measures our prior belief in the correlation between the log model evidence evaluated at two kernels.

Here we consider two kernels to be "similar" for a given dataset $\mathcal{D}$, *if they offer similar explanations for the latent function at the observed locations.*

The "kernel kernel"

A model $\mathcal{M}$ *induces* a prior distribution over latent function values at given locations $\mathbf{X}$:

$$p(\mathbf{f} \mid \mathbf{X}, \mathcal{M}) = \int p(\mathbf{f} \mid \mathbf{X}, \theta, \mathcal{M}) p(\theta) \, \mathrm{d}\theta$$

This is an (infinite) mixture of multivariate Gaussians, each of which is a potential explanation of the latent function values $\mathbf{f}$ (and thus for the observed data $\mathbf{y}$).

The "kernel kernel"

Given input locations $\mathbf{X}$, we suggest two models $\mathcal{M}$ and $\mathcal{M}'$ should be similar when the latent explanations

$$p(\mathbf{f} \mid \mathbf{X}, \mathcal{M}) \qquad p(\mathbf{f} \mid \mathbf{X}, \mathcal{M}')$$

are similar; i.e., they have *high overlap.*

Measuring overlap: Hellinger distance

Omitting many details, we have a solution: the so-called *expected Hellinger distance*

$$\bar{d}_H^2(\mathcal{M}, \mathcal{M}'; \mathcal{D})$$

(the expectation is over the *hyperparameters* of each model).

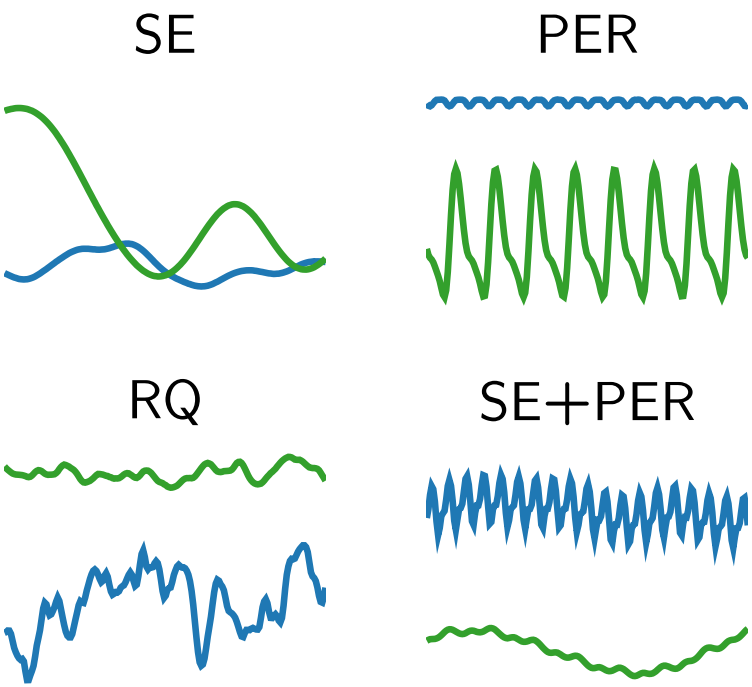
The "kernel kernel"

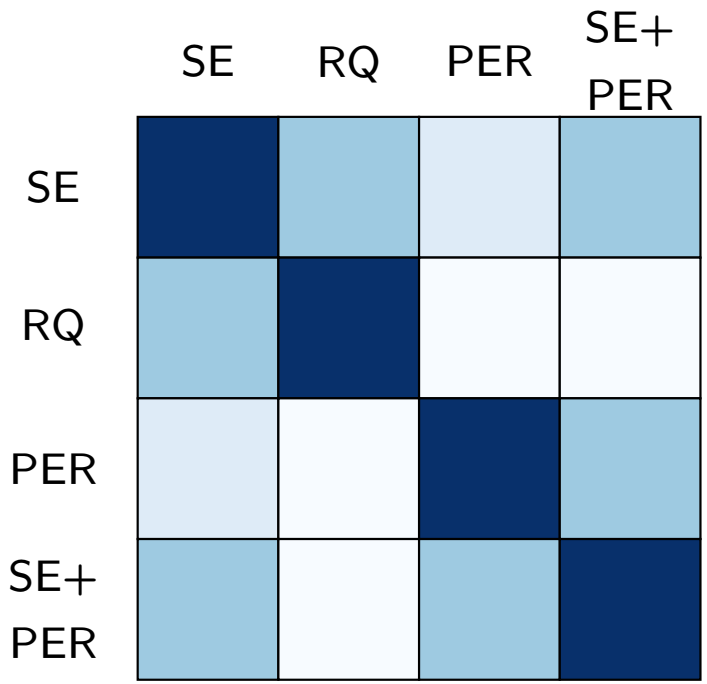
Now our "kernel kernel" between two models $\mathcal{M}$ and $\mathcal{M}'$, given the data $\mathcal{D}$, is defined to be

$$K_g(\mathcal{M}, \mathcal{M}'; \mathcal{D}, \ell) = \exp\left(-\frac{1}{2\ell^2}\bar{d}_H^2(\mathcal{M}, \mathcal{M}'; \mathcal{D})\right).$$

Crucially, this *depends on the data distribution!*

"Kernel kernel:" Illustration





OPTIMIZING THE ACQUISITION FUNCTION

Overview of approach

We have defined a model over the model evidence function. We still need to figure out how to maximize the *acquisition function* (e.g., expected improvement)

$$\mathcal{M}' = \arg\max_{\mathcal{M} \in \mathbb{M}} \alpha(\mathcal{M}; \mathcal{D}_g).$$

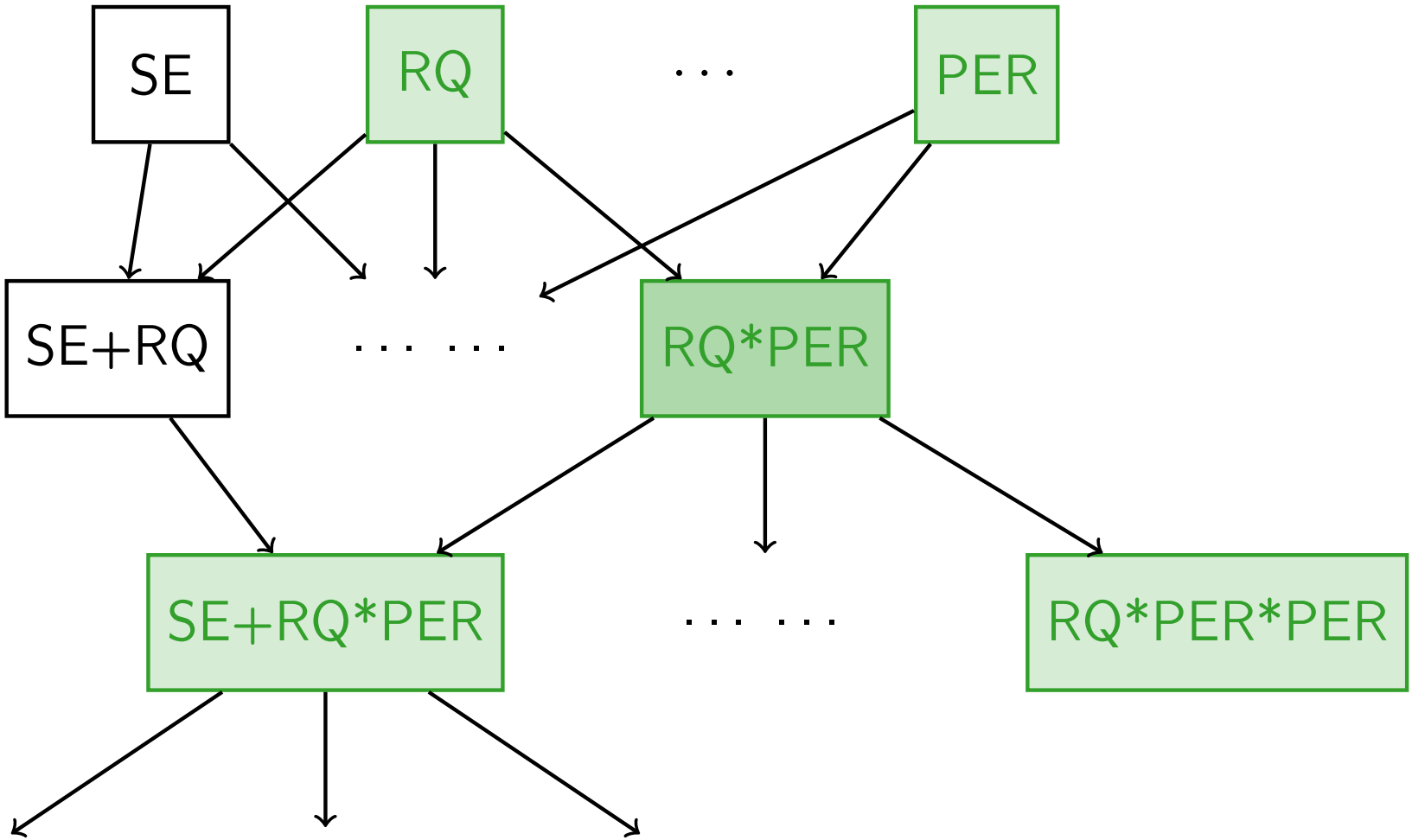
Active set construction

Our idea: dynamically maintain a bag of (~ 500) *candidate models* and optimize α on that smaller set.

To construct this set, we will heuristically encourage *exploitation* and *exploration.*

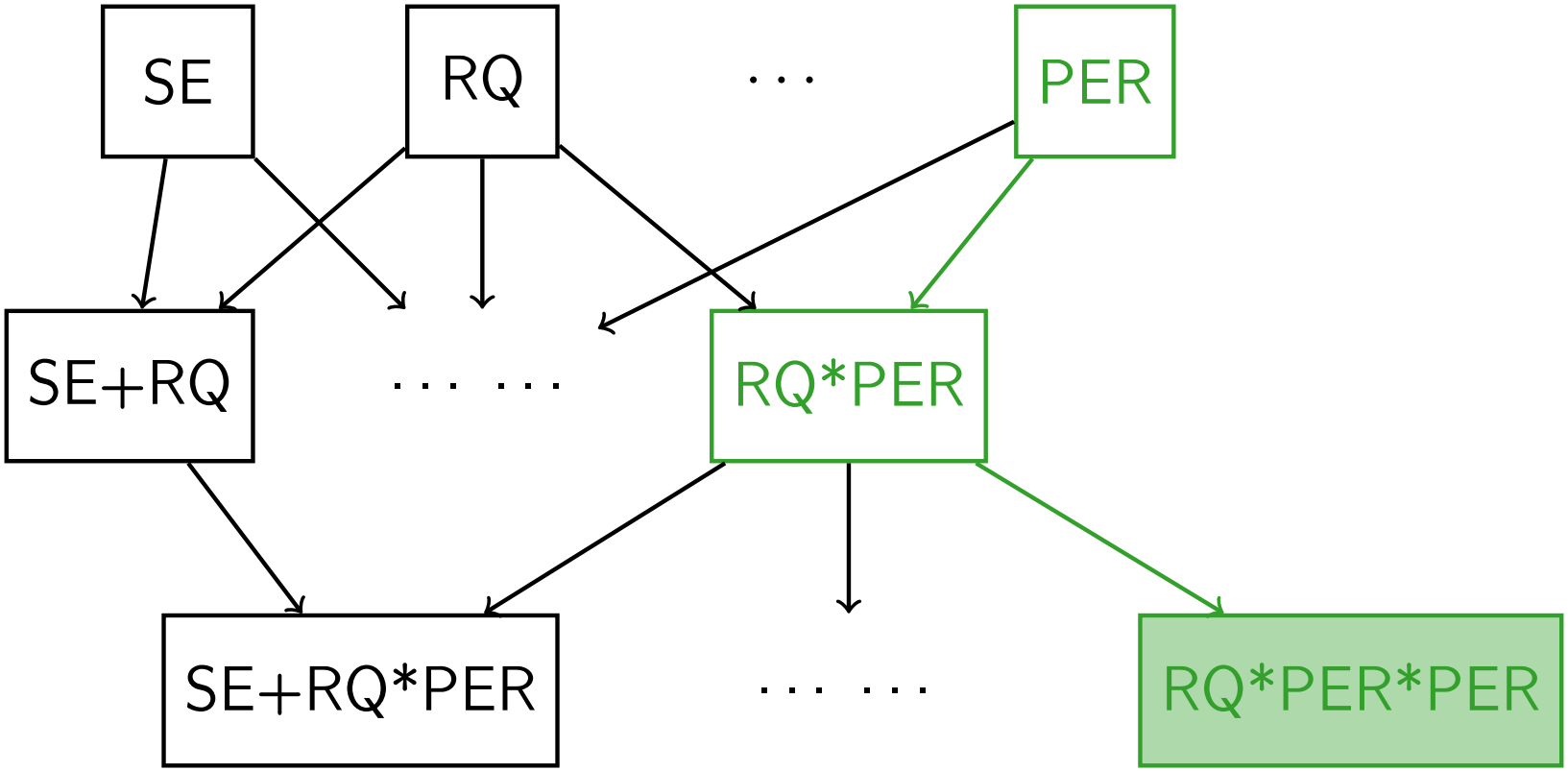
Active set construction: Exploitation

Exploitation: add models near *the best-yet seen.*



Active set construction: Exploration

Exploration: add models generated from (short) *random walks* from the empty kernel.



EXPERIMENTS

Experimental setup

- We compare our method (Bayesian optimization for model selection, BOMS) against the greedy search method from Duvenaud, et al. ICML 2013.
- *Laplace approximation* for estimating model evidence.
- Budget of 50 model evidence computations.

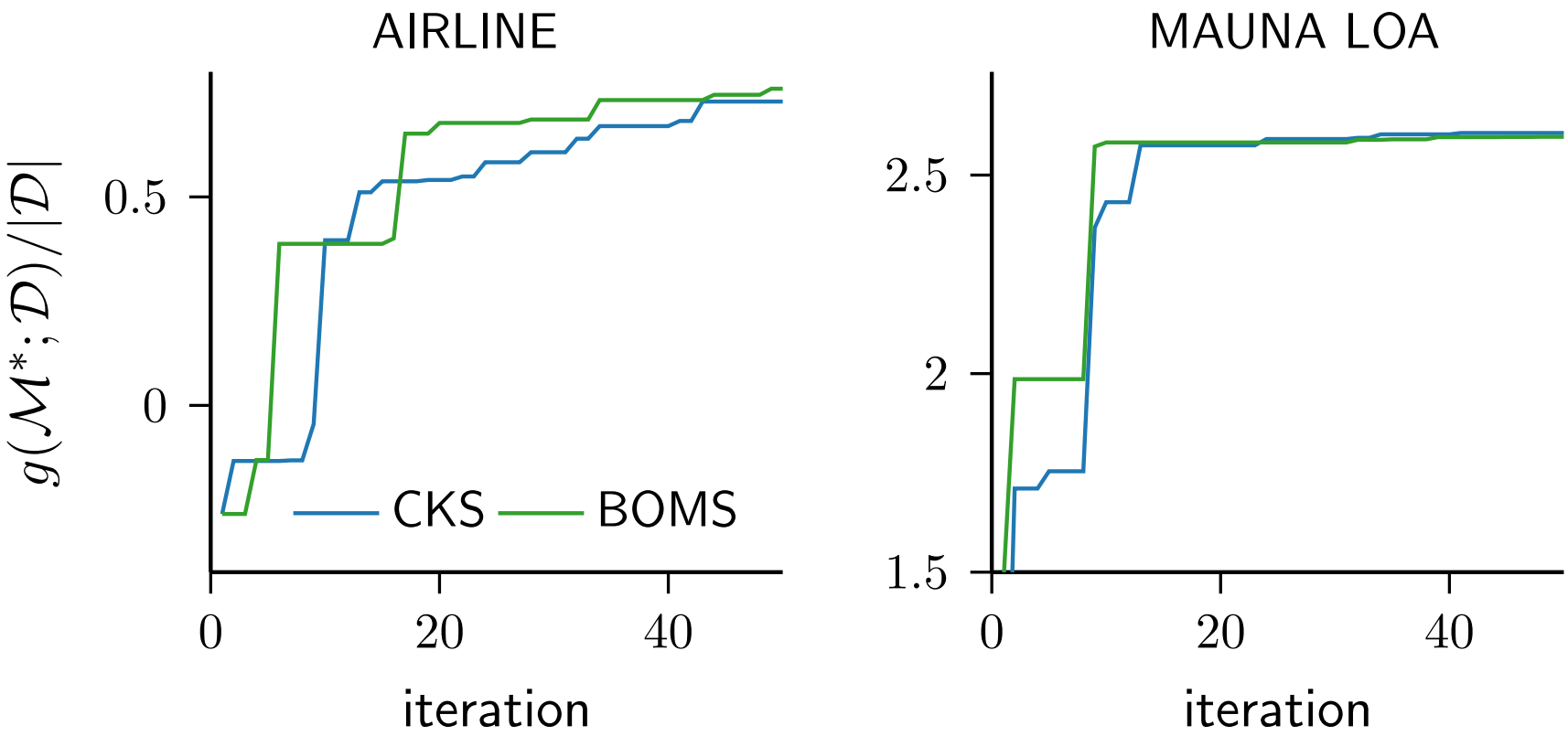
Model space: CKS grammar

- For time-series data, the base kernels were SE, RQ, LIN, and PER.
- For higher-dimensional data, the base kernels were SE_i and RQ_i.

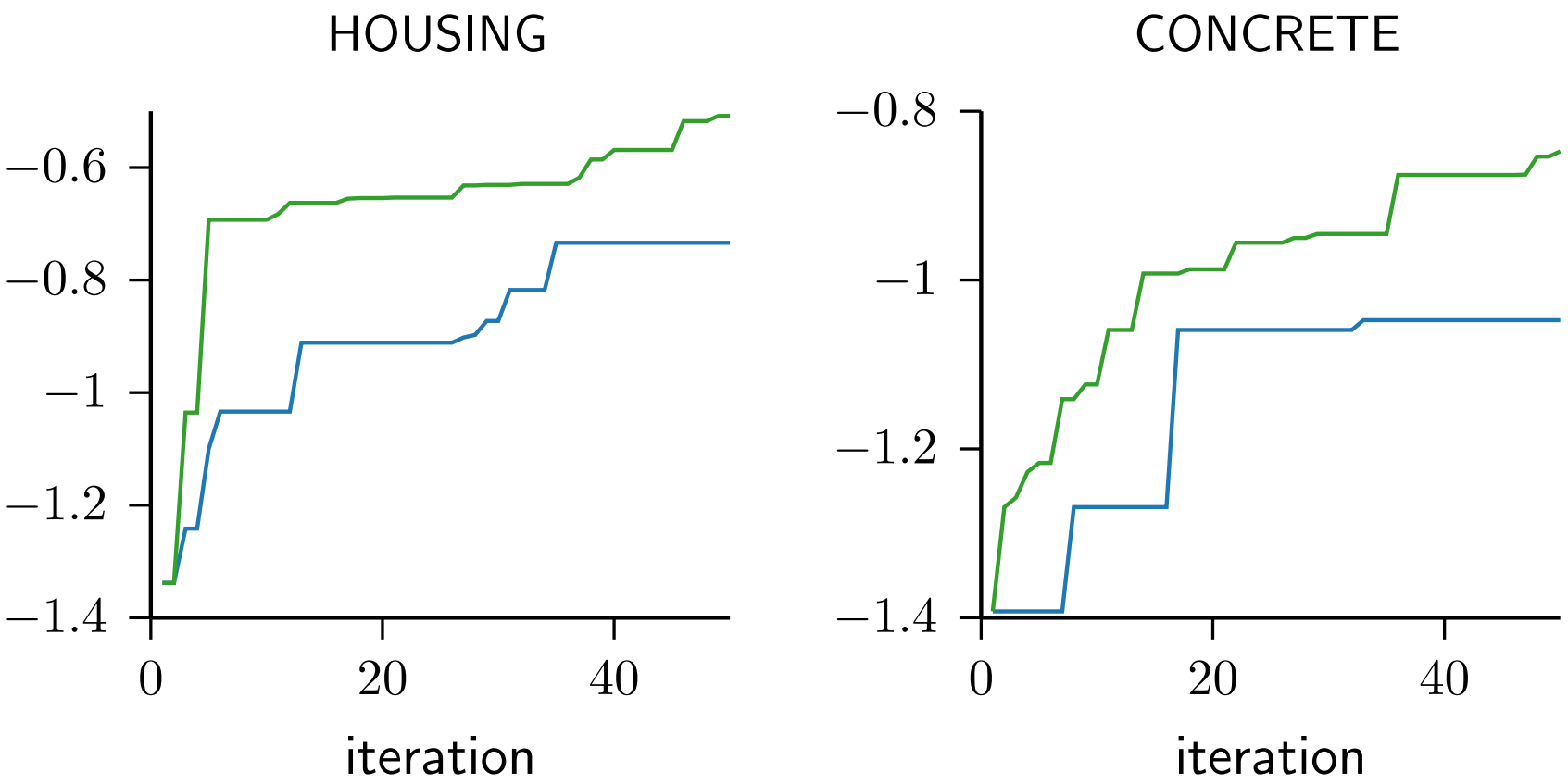
Experimental setup: Details for BOMS

- First model selected was SE.
- Acquisition function was *expected improvement per second.*

Results: Time series



Results: High-dimensional data



Notes

- The overhead of our method in terms of running time is *approximately 10%*.
- The *vast majority* of the time is spent optimizing hyperparameters (random restart, etc.).
- We offer some advice for automatically selecting reasonable *hyperparameter priors* for given data that we adopt here.

Other options

For Bayesian optimization, may want to choose another family of kernels, e.g.,

- *Additive* decompositions (Kandasamy, et al., ICML 2015)
- Low-dimensional *embeddings* (Wang, et al., IJCAI 2013, Garnett, et al. UAI 2014)

Both would be *convenient* for optimization for other reasons (e.g., easier optimization of the acquisition function)

LOOKING FORWARD

Looking forward

These results are promising, but the real promise of such methods is in the *inner loop* of another procedure (e.g., Bayesian optimization or Bayesian quadrature)!

Future code snippet?

```
data = [];
models = [SE];

for i = 1:budget
  % use mixture of models in acquisition function
  x_next = maximize_acquisition(data, models);
  y_next = f(x_next);
  data = data + [x_next, y_next];

  % update bag of models
  models = update_models(data, models); % BOMS
end
```

THANK YOU!

Questions?