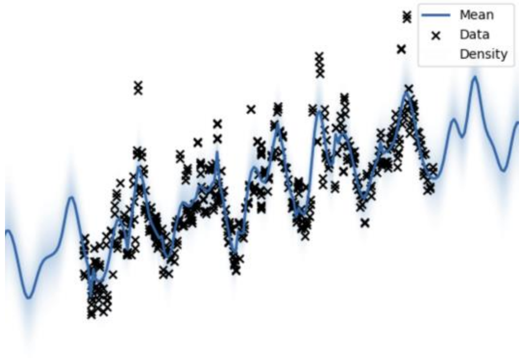
Automated Kernel Search using Evolutionary Algorithms

Louis Schlessinger

May 3rd, 2019

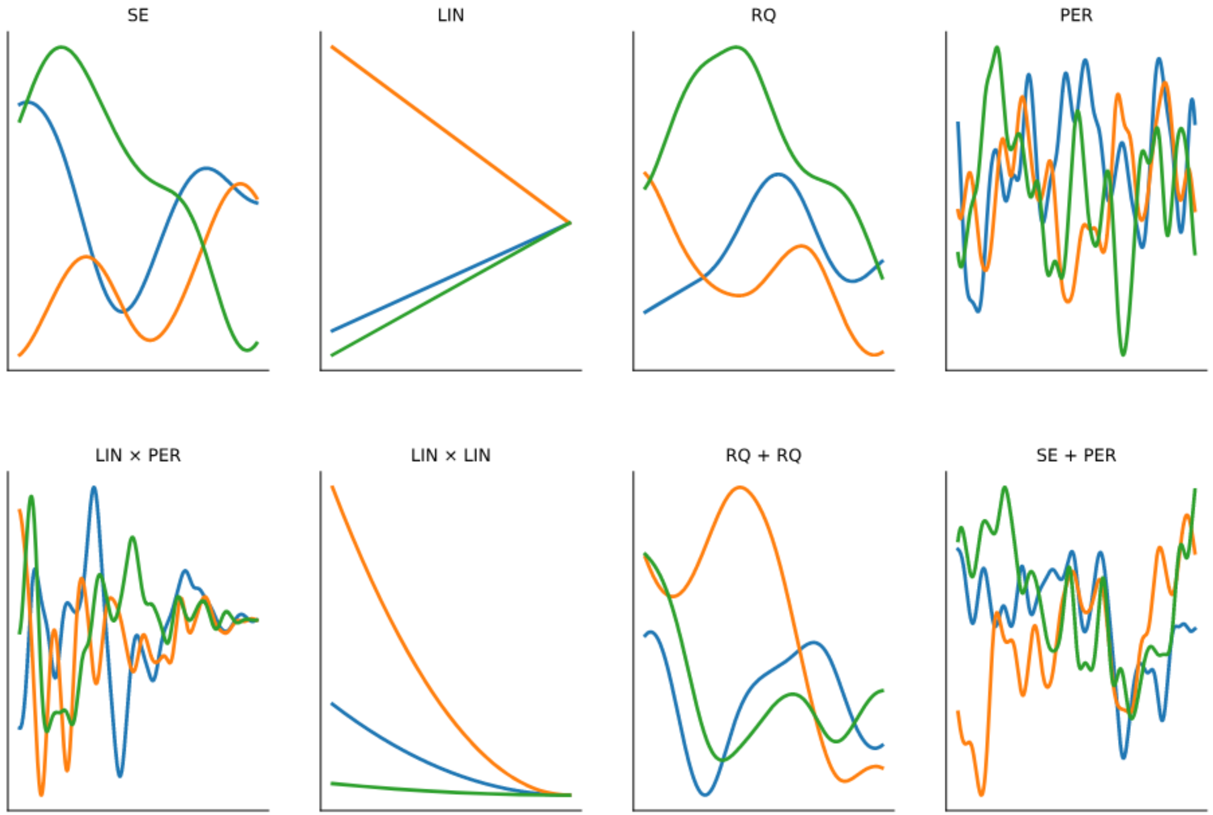


Introduction

Problem Setup

- The popular kernel-based, nonparametric **Gaussian process** (GP) model is able to discover patterns and structure in data.
- **Covariance functions** (or **kernels**) encode structural assumptions about which kinds of functions are likely.
- The task of selecting an appropriate kernel is crucial for **generalization** and nontrivial because the space of possible kernels is infinite.
- Consequently, it has been called a "**black art**" and is either left for experts or an off-the-shelf option is used.
- The goal here is to automatically construct a covariance function for a Gaussian Process model using Genetic Programming.

GP models can represent many types of structures



GP Regression

- Unknown latent function $f: \mathcal{X} \mapsto \mathbb{R}$
- Given dataset $\mathcal{D} = (X, y)$
- Assume additive i.i.d. Gaussian noise such that

$$y_i = f(x_i) + \mathcal{N}\left(0, \sigma_n^2\right)$$

- Place GP prior on f:

$$p(\, f \mid \theta \,) = \mathcal{GP}\big(f; \mu(x; \theta), k(x, x'; \theta)\big)$$

Fitness Function

- Quality of fit of a GP to $\mathcal{D}$ taken to be the log marginal likelihood

$$\log p(\boldsymbol{y} \mid X, \theta)$$

- Where,

$$p(\boldsymbol{y} \mid X, \theta) = \int p(\boldsymbol{y} \mid \boldsymbol{f}, X) p(\boldsymbol{f} \mid X, \theta) d\boldsymbol{f}$$

- This balances data fit and model complexity
- Under additive i.i.d. Gaussian noise, we can compute this analytically

Objective

- However, we need to optimize θ jointly with σ_n^2 to maximize log marginal likelihood

$$\theta_{MLE} = \underset{\theta}{\operatorname{argmax}} \log p(y \mid X, \theta)$$

- Typically done using quasi-Newton method (e.g. L-BFGS)
- So, the fitness of a model is naively taken to be

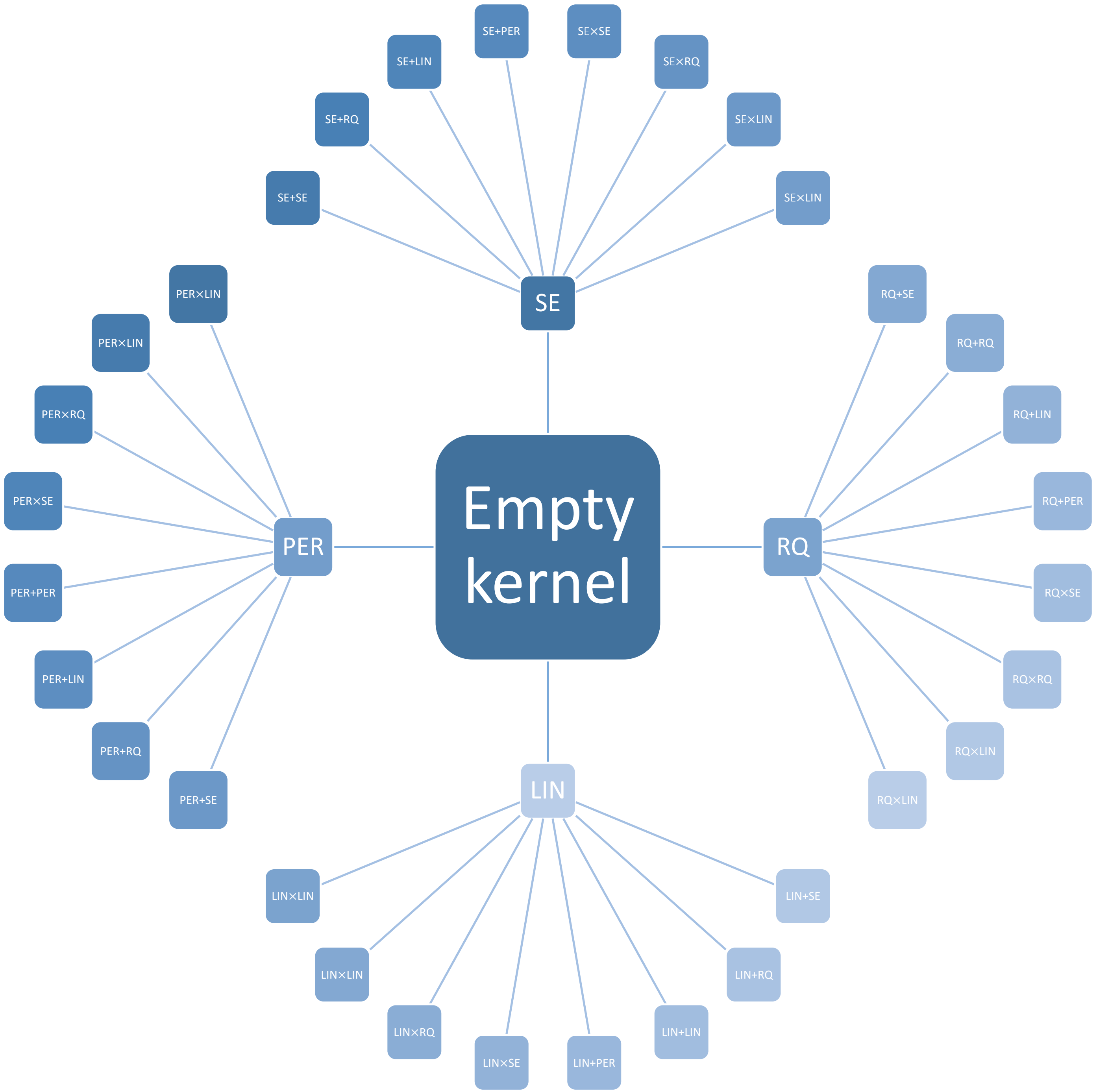
$$\log p(y \mid X, \theta_{MLE})$$

Compositional Kernel Space

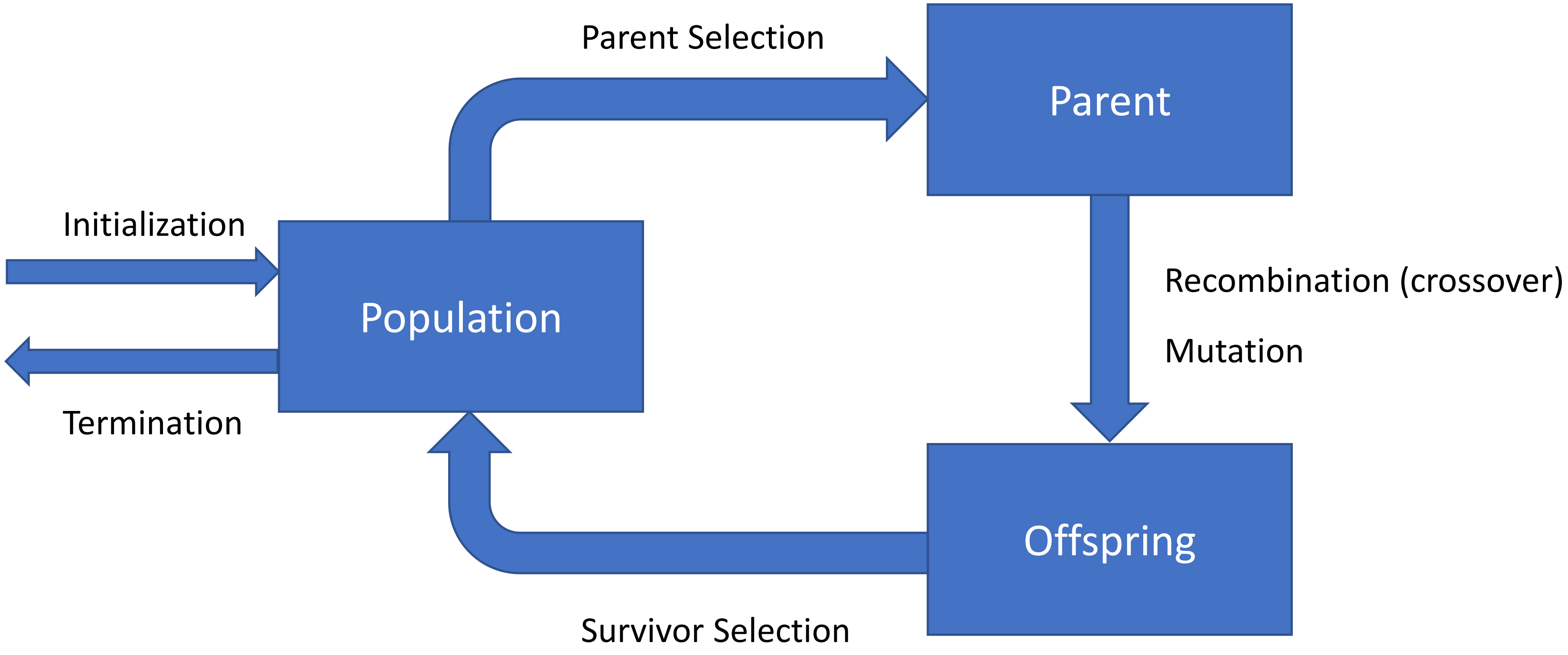
- Context-free grammar (CFG) rules[1] are:

1. Any subexpression $\mathcal{S}$ can be replaced with $\mathcal{S} + \mathcal{B}$, where $\mathcal{B}$ is any base kernel family.
2. Any subexpression $\mathcal{S}$ can be replaced with $\mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ is any base kernel family.
3. Any base kernel $\mathcal{B}$ may be replaced with any other base kernel family $\mathcal{B}'$.

[1] Duvenaud et al., 2013

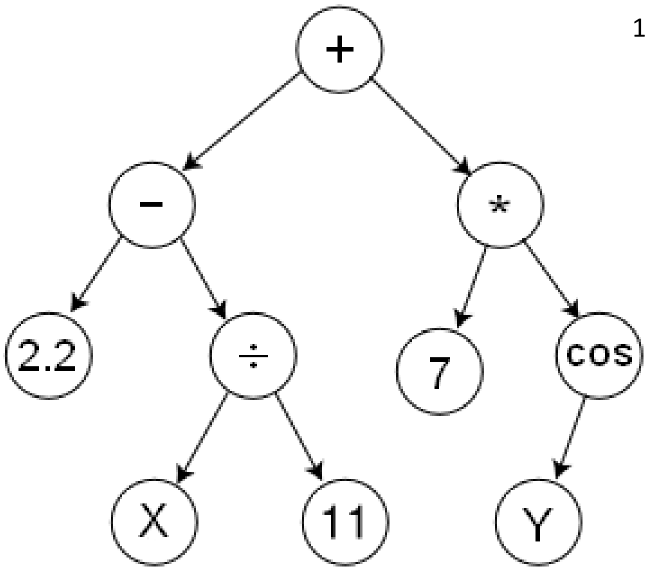

SE+PER
SE×SE
SE+LIN
SE×RQ
SE+RQ
SE×LIN
SE+SE
SE×LIN
SE
PER×LIN
RQ+SE
PER×LIN
RQ+RQ
PER×RQ
RQ+LIN
PER×SE
RQ+PER
Empty kernel
PER
RQ
PER+PER
RQ×SE
PER+LIN
RQ×RQ
PER+RQ
RQ×LIN
PER+SE
RQ×LIN
LIN
LIN×LIN
LIN+SE
LIN×LIN
LIN+RQ
LIN×RQ
LIN+LIN
LIN×SE
LIN+PER

Evolutionary Algorithms



Genetic Programming

- Terminal set T
- Function set F
- Expression Trees
  - Composed of primitives from $T \cup F$



$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$

[1] https://en.wikipedia.org/wiki/Binary_expression_tree

Related Work

- Greedy Search:
  - Structure Discovery in Nonparametric Regression through Compositional Kernel Search (Duvenaud et al., 2013)

- Bayesian Optimization:
  - Bayesian optimization for automated model selection (Malkomes et al., 2016)

- Grammatical Evolution:
  - Evolution of covariance functions for gaussian process regression using genetic programming (Kronberger & Kommenda, 2013)

Challenges

- Model evidence is expensive to estimate $\mathcal{O}(N^3)$
- Relationship between covariance functions and model evidence is complex
- Depends on structure of $\mathcal{D}$
- No gradient information

Motivation

- A sum of kernels is an OR-like operation
- A product of kernels is an AND-like operation[1]
- Locality is critical for the success of evolutionary algorithms, otherwise they will degenerate to random search[2]
- That is, genotypic neighbors must correspond to phenotypic neighbors

[1] Duvenaud et al., 2013
[2] Rothlauf et al., 2006

Why Evolutionary Algorithms?

- Crossover operator can reduce dimensionality of search space if it’s possible to search for global maximum by searching for maximum in each dimension independently

- Goal is to look at the feasibility of using genetic programming in GP kernel search

Evolutionary Kernel Construction

Evolutionary Kernel Search (EKS)

- Search for derivations of CKS grammar
- Here, $T = \{SE, RQ, PER, LIN\}$
- $F = \{+,\times\}$
- Kernels are closed under $+$ and $\times$
- Therefore, we can grammatically evolve kernels
- This primitive set implies a kernel encoding

Compositional Kernel Tree

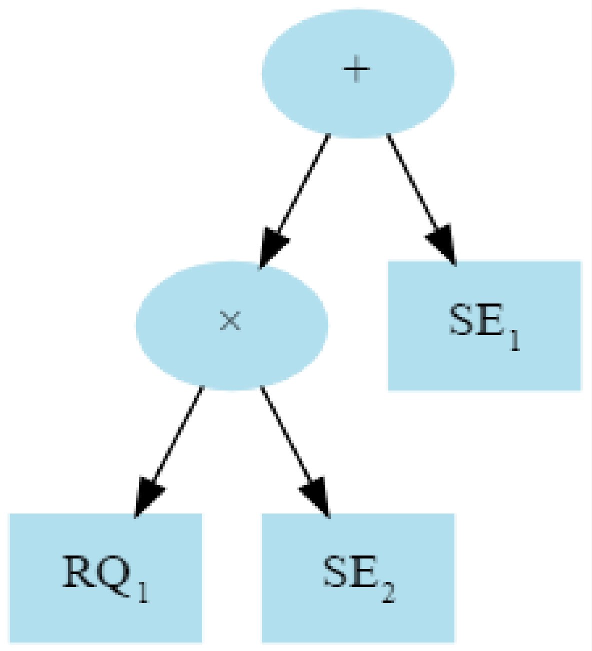
- Need to map from covariance functions to trees
- *Full* binary expression tree with N total nodes

$I = \frac{N-1}{2}$ internal nodes (operators)

$L = \frac{N+1}{2}$ leaf nodes (base kernels),

Where $N \in \{1,3,5 \ldots\}$

$(RQ_1 \times SE_2) + SE_1$



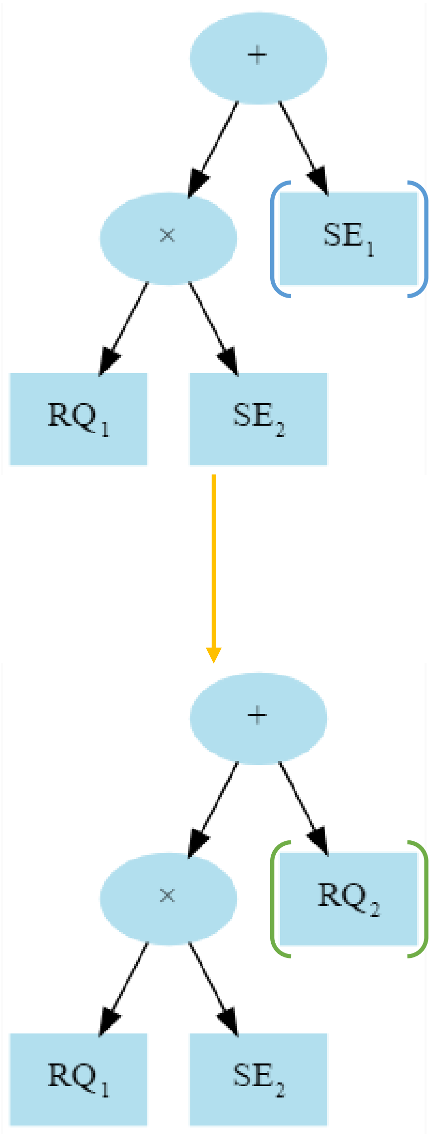
Kernel Crossover

- Here, crossover is a mapping $\mathbb{M}^2 \mapsto \mathbb{M}^2$

- A modification of the standard sub-tree exchange crossover is used:
  - Leaf-biased **sub-tree exchange crossover**

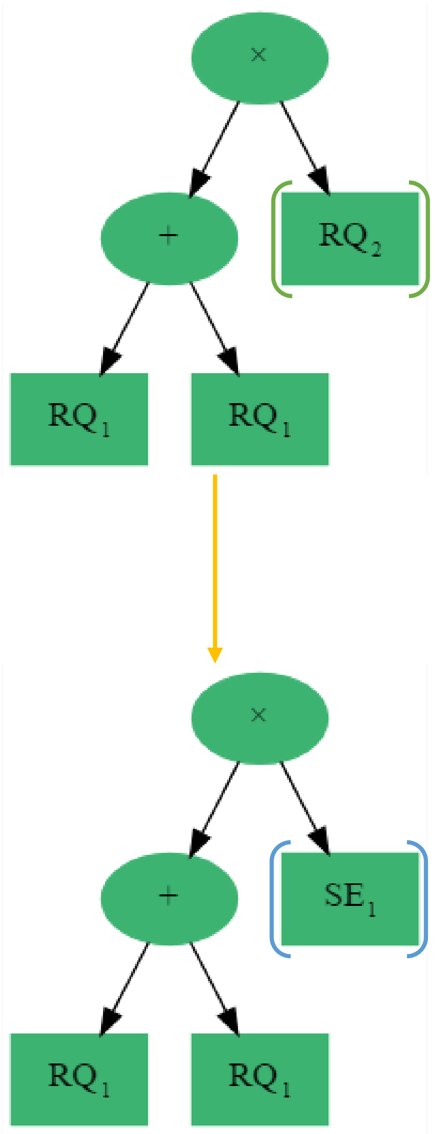
- Swapping sub-trees can be thought of as exchanging structural assumptions

$(RQ_1 \times SE_2) + SE_1$



$(RQ_1 \times SE_2) + \boldsymbol{RQ_2}$

$(RQ_1 + RQ_1) \times RQ_2$



$(RQ_1 + RQ_1) \times \boldsymbol{SE_1}$

Why Leaf-biased?

- Naïve uniform random selection of two crossover points:

$$p(\text{leaf}) \qquad = \frac{1}{2} + \frac{1}{2N}$$

$$p(\text{internal}) = \frac{1}{2} - \frac{1}{2N}$$

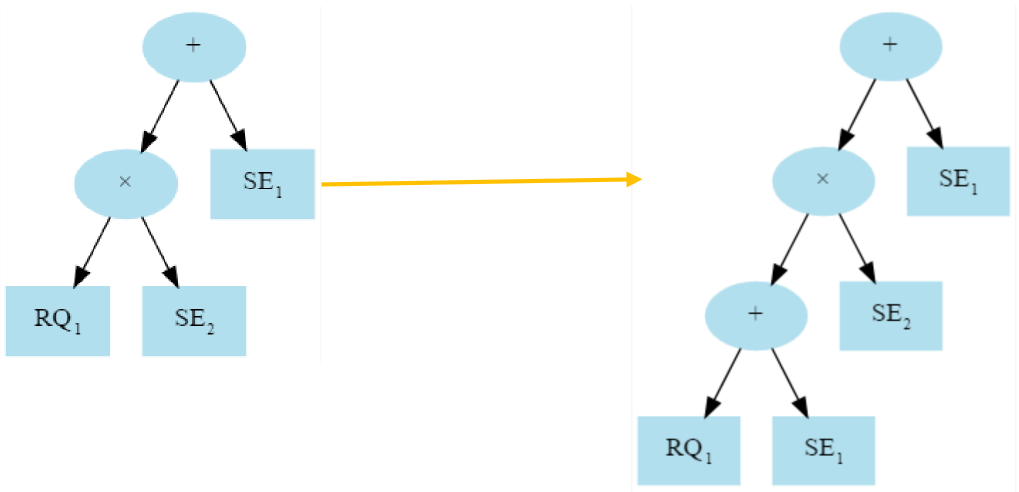
- Undesirable property of swapping mostly leaves, taking small steps in model space

- Empirically find leaf probability of 0.1 to work well

Kernel Mutation

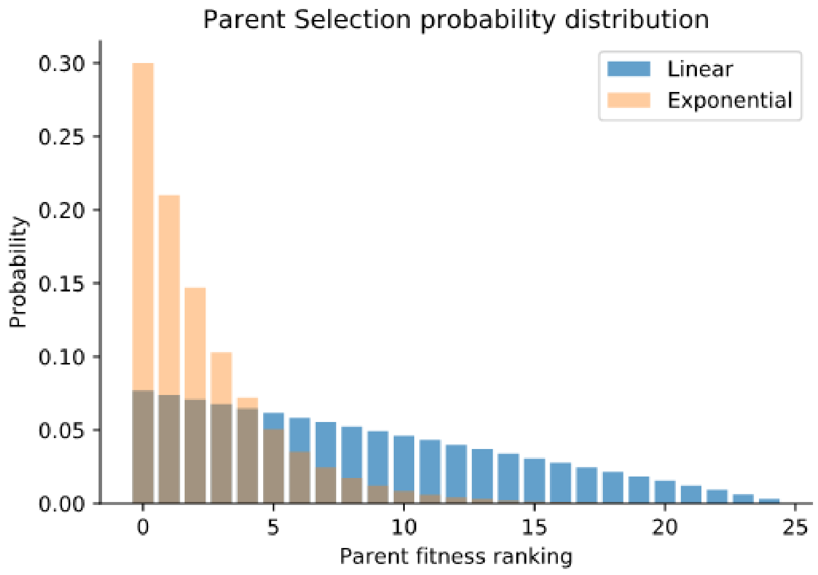
- Mutation is a mapping $\mathbb{M} \mapsto \mathbb{M}$
- **Subtree replacement mutation** with a *Ramped Half-n-Half* random tree generator is used up to height 2.

$(RQ_1 \times SE_2) + SE_1$ $\quad\quad$ $((\boldsymbol{RQ_1} + \boldsymbol{SE_1}) \times SE_2) + SE_1$



Implementation

- Initialization:
  - *Ramped Half-n-Half*
- Parent selection:
  - Exponential Ranking Selection
- Offspring selection:
  - Truncation Selection
- Duplicate removal:
  - By expanded composite kernel equivalence (and only evaluated once)



Experimental Parameters

- Population size:
  - 25
- Population-level crossover rate:
  - 0.6
- Population-level mutation rate:
  - 0.1
- Variation rate:
  - $(0.6 + 0.1)$

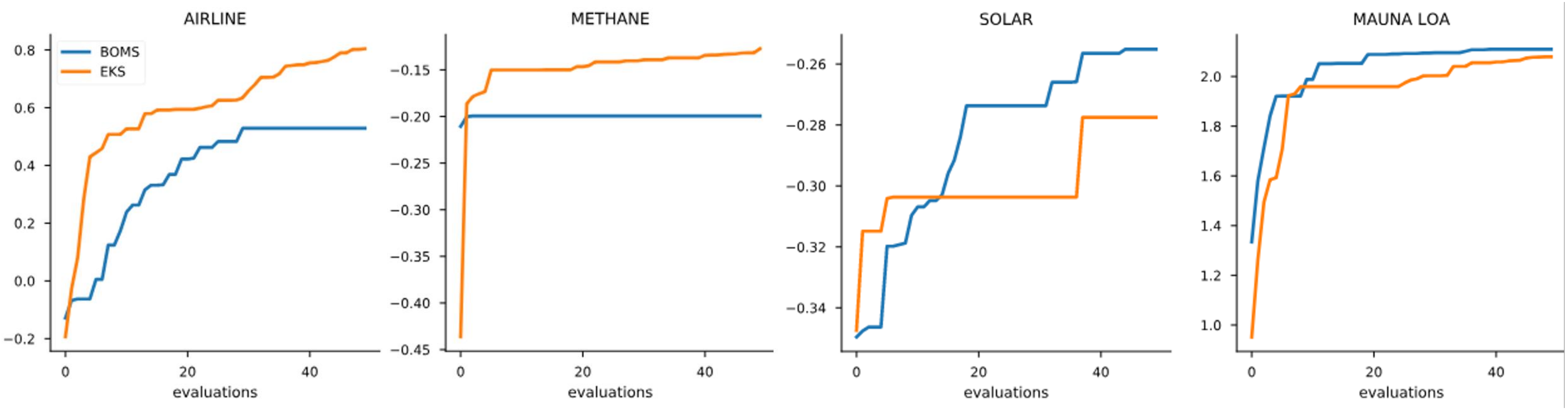
Experiments

Experimental Setup

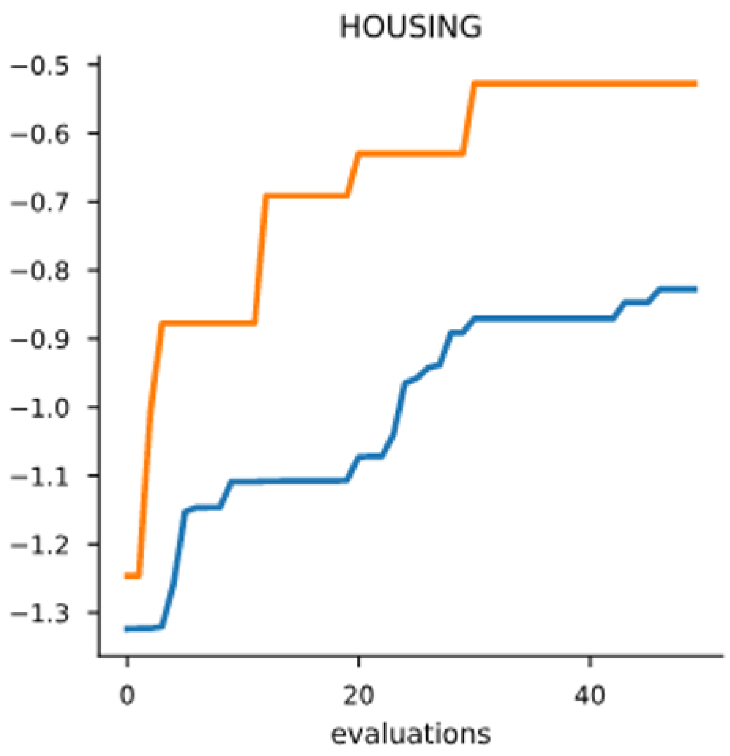
- This method evolutionary kernel search (EKS) is compared to Bayesian optimization for automated model selection (BOMS).[1]
- Same model space
- Log evidence divided by dataset size is reported
  - For BOMS, Laplace approximation is used
  - For EKS, MLE estimate is used

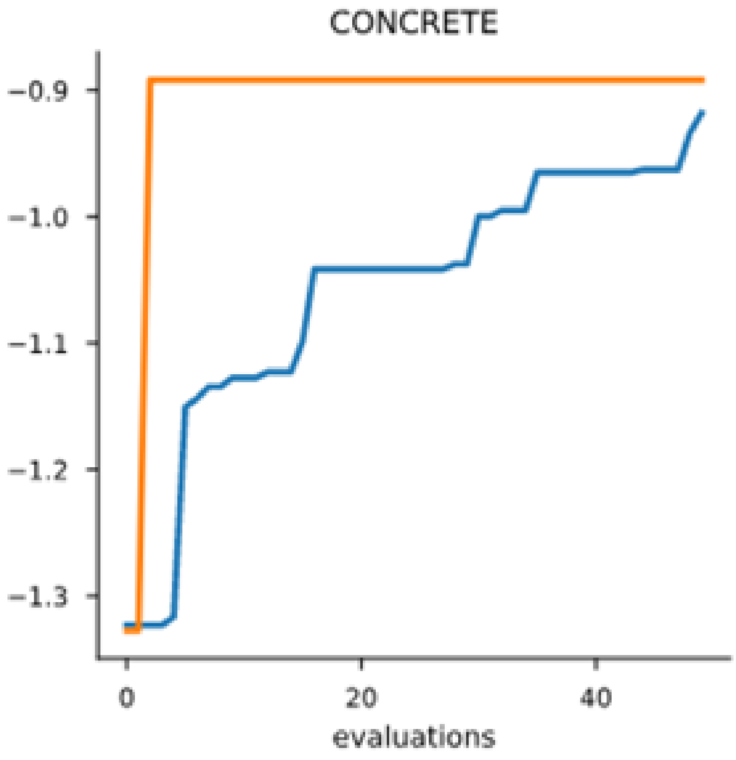
[1] Malkomes et al., 2016

One-dimensional timeseries

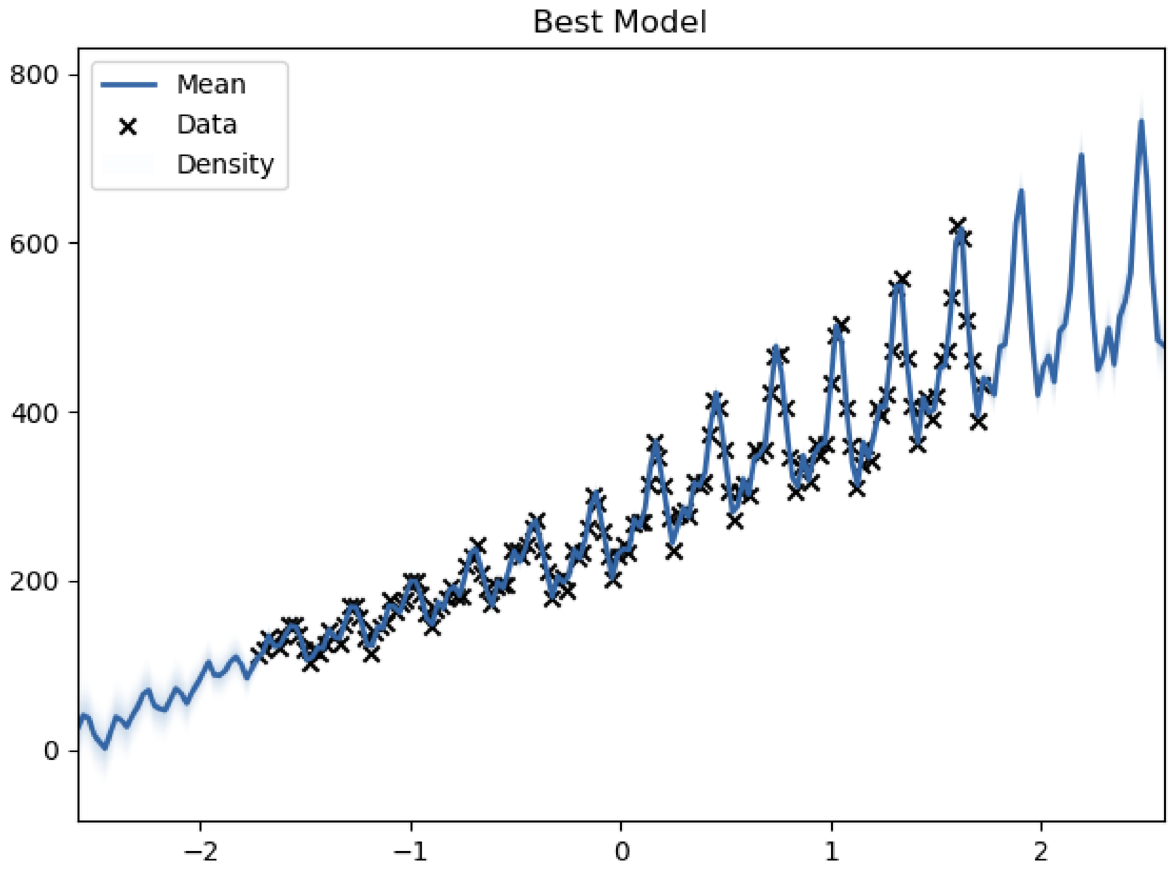


Multidimensional Datasets





Example: Airline Dataset



Summary

- Presented a computationally inexpensive method to propose candidate models using genetic programming
- Capable of recovering structure on a variety of datasets
- High selection pressure necessary; most models give poor explanations of data

Limitations

- Sample inefficient
- No convergence guarantees
- Population dynamics empirically found to be very unstable

Future Work

Future Work

- Exploring the effectiveness of the variation operators in a surrogate-based method for candidate proposal, possibly taking larger steps in model space
- This implicit distribution over candidates by the crossover operator and mutation operators as a proposal distribution
- Measuring locality of the representation proposed here

Thank you!

Questions?

Relevant Links

- Report
- Code