Seminar: različne vrste nadomestkov

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za matematiko in fiziko
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

Marec 2023

Pregled vsebine

Optimizacija TPE

TPE = Tree-Structured Parzen Estimators

Temelji na ideji SMBO z dvema spremembama

- Drugačni, Parzenovi nadomestki
- Drevesna struktura prostora parametrov

Primerjava z nadomestki SMBO

premikajo se po celotnem prostoru

Nadomestki SMBO

- Za podano konfiguracijo $\boldsymbol{\theta}$, vektor vrednosti vseh nad-parametrov
- Napovedujejo pričakovano vrednost standardni odklon ciljne funkcije p
- En napovedni model oblike $P(p|\boldsymbol{\theta})$

nadomestek mora izračunat pričakovano vrednost funkcije p $= \mu_\theta$ in σ_θ (odklon)

$s(\theta)$

Nadomestki TPE

model verjetnostne porazdelitve

- Napovedni modeli so oblike $P(\theta|p)$
- En nadomestni model za vsak nad-parameter θ iz vektorja $\boldsymbol{\theta}$

grejo ubratno kot SMBO

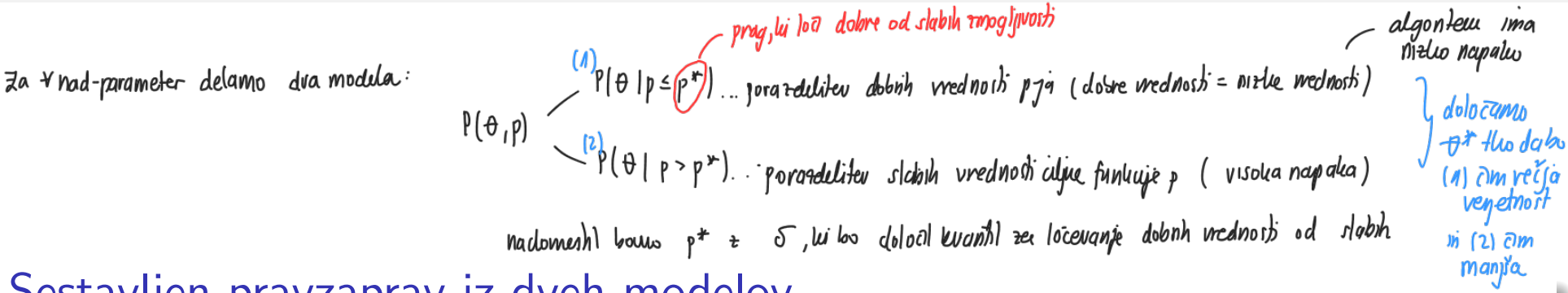
spnjazrijo se, da imajo model $P(\theta|p)$ (prve nam kališi θ rabmo)

Zakaj poseben nadomestek za vsak parameter?

Zato, da dosežemo večjo fleksibilnost

- Nadomestni modeli so bolj enostavni, Parzenovi nadomestki
- Vsak nad-parameter lahko obravnavamo ločeno od ostalih
- Določimo drevesno strukturo odvisnosti med nad-parametri

Nadomestni model TPE za nad-parameter θ



Sestavljen pravzaprav iz dveh modelov

1. Porazdelitev slabih vrednosti parametra $P(\theta|p > p^*) = P(\theta|\text{slabo})$
2. Porazdelitev dobrih vrednosti parametra $P(\theta|p \leq p^*) = P(\theta|\text{dobro})$

3. Vrednost p^* je prag, ki loči dobre od slabih zmogljivosti
4. Določimo ga s parametrom γ, ki določa delež dobrih vrednosti
5. Npr. $\gamma = 0.1$ določa prag p^* tako, da je 10% najnižjih (najboljših) opazovanih vrednosti ciljne funkcije p dobrih, ostale so slabe

Kako izberemo naslednjo vrednost θ?

določimo tako, da (1) čim večja in (2) čim manjša

$$\theta^* = \arg\max \frac{P(\theta^*|\text{dobro})}{P(\theta^*|\text{slabo})}$$

(1) v števcu, (1) v imenovalcu

[tako izbira θ^* je ekvivalentna izbiri pričakovane izboljšave.] max tega $\Leftrightarrow$ max pričakovane izboljšave

Ideja najbolj obetavne vrednosti θ^*

- Verjetnost $P(\theta^*|\text{dobro})$ bi morala biti čim večja
- Verjetnost $P(\theta^*|\text{slabo})$ bi morala biti čim manjša

Taka izbira je ekvivalentna izbiri pričakovane izboljšave (1)

Vpeljimo najprej novi oznaki porazdelitev

$$P(\theta|p) = \begin{cases} P_d(\theta) & ; p \leq p^* \\ P_s(\theta) & ; p > p^* \end{cases}$$

porazdelitev dobrih vrednosti

Poglejmo zdaj čemu je enaka porazdelitev $P(\theta)$

$$\begin{aligned} P(\theta) &= P(\theta|p \leq p^*)\, P(p \leq p^*) + P(\theta|p > p^*)\, P(p > p^*) \\ &= \gamma\, P_d(\theta) + (1-\gamma)\, P_s(\theta) \end{aligned}$$

formula za popolno verj. (pod različnimi pogoji)

porazdelitev dobrih vrednosti

po def. (tako smo γ definirali)

Spomnimo se od prej: $\gamma = P(p \leq p^*)$

Taka izbira je ekvivalentna izbiri pričakovane izboljšave (2)

expected improvement

$$EI(\theta) = \int_{-\infty}^{p^*} (p^* - p)\, P(p|\theta)\, dp$$

Bayes

$$= \int_{-\infty}^{p^*} (p^* - p)\, \frac{P(\theta|p)\, P(p)}{P(\theta)}\, dp$$

$$= \frac{1}{\gamma\, P_d(\theta) + (1-\gamma)\, P_s(\theta)} \int_{-\infty}^{p^*} (p^* - p)\, P(\theta|p)\, P(p)\, dp$$

$$= \frac{P_d(\theta)}{\gamma\, P_d(\theta) + (1-\gamma)\, P_s(\theta)} \int_{-\infty}^{p^*} (p^* - p)\, P(p)\, dp$$

$$E_1(\theta) = \int_{-\infty}^{p^*} (p^* - p)\, P(p|\theta)\, dp = \int_{-\infty}^{p^*} (p^* - p)\, \frac{P(\theta|p)\cdot P(p)}{P(\theta)}\, dp = \frac{P_d(\theta)}{P(\theta)} \int_{-\infty}^{p^*} \underbrace{p^* \cdot P(p)}_{p^* \cdot \gamma}\, dp - \frac{P_d(\theta)}{P(\theta)} \int_{-\infty}^{p^*} p \cdot P(p)\, dp$$

$$P(\theta) = P(p \le p^*)\, P(\theta | p \le p^*) + P(p > p^*)\, P(\theta | p > p^*) =$$

$$= \gamma \cdot P_d(\theta) + (1-\gamma)\, P_s(\theta)$$

$$P(\theta|p)_{p \le p^*} = P_d(\theta)$$

Taka izbira je ekvivalentna izbiri pričakovane izboljšave (3)

Ker

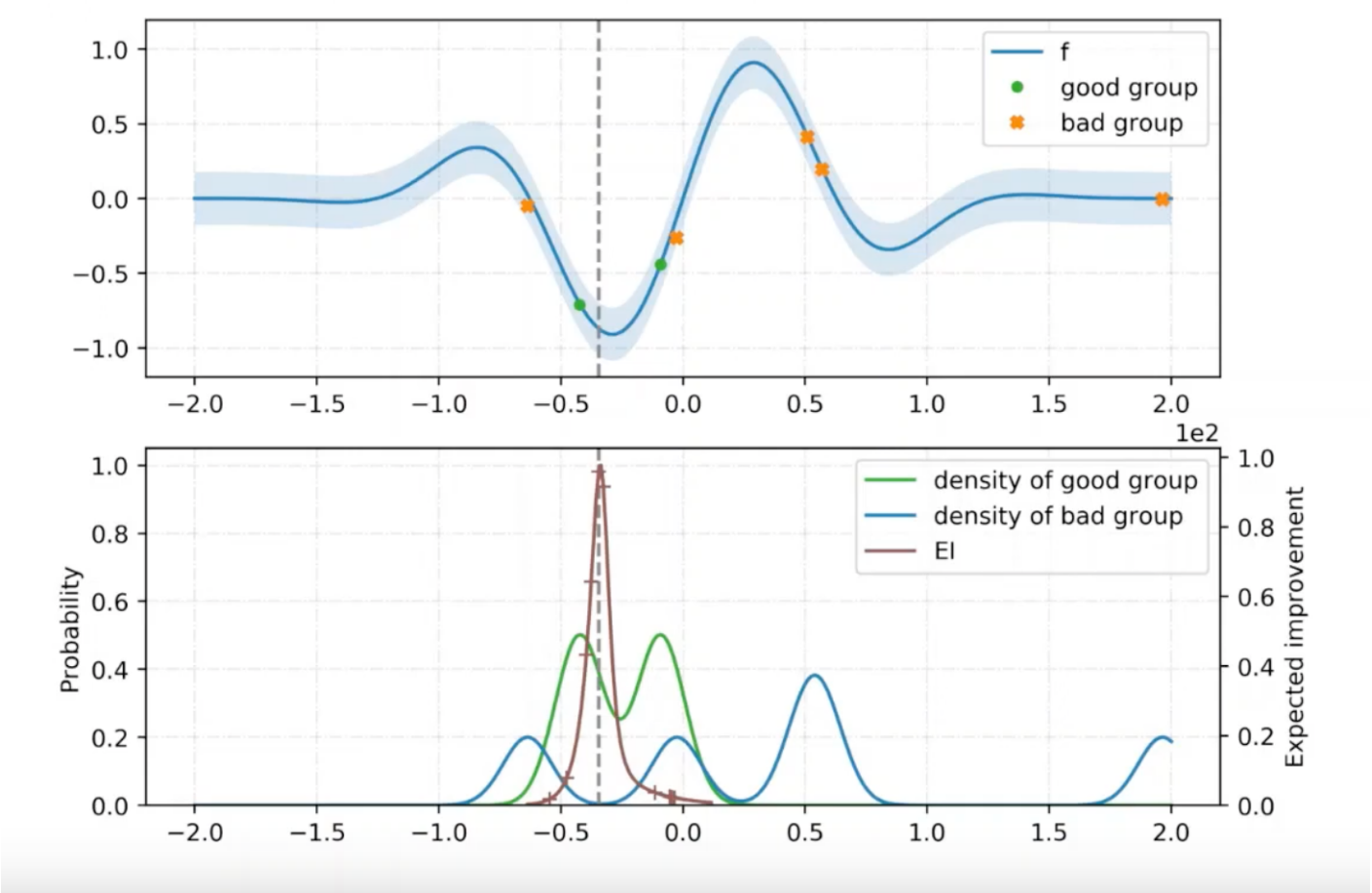
$$\int_{-\infty}^{p^*} p^*\, P(p)dp = p^*\,\gamma$$

Velja

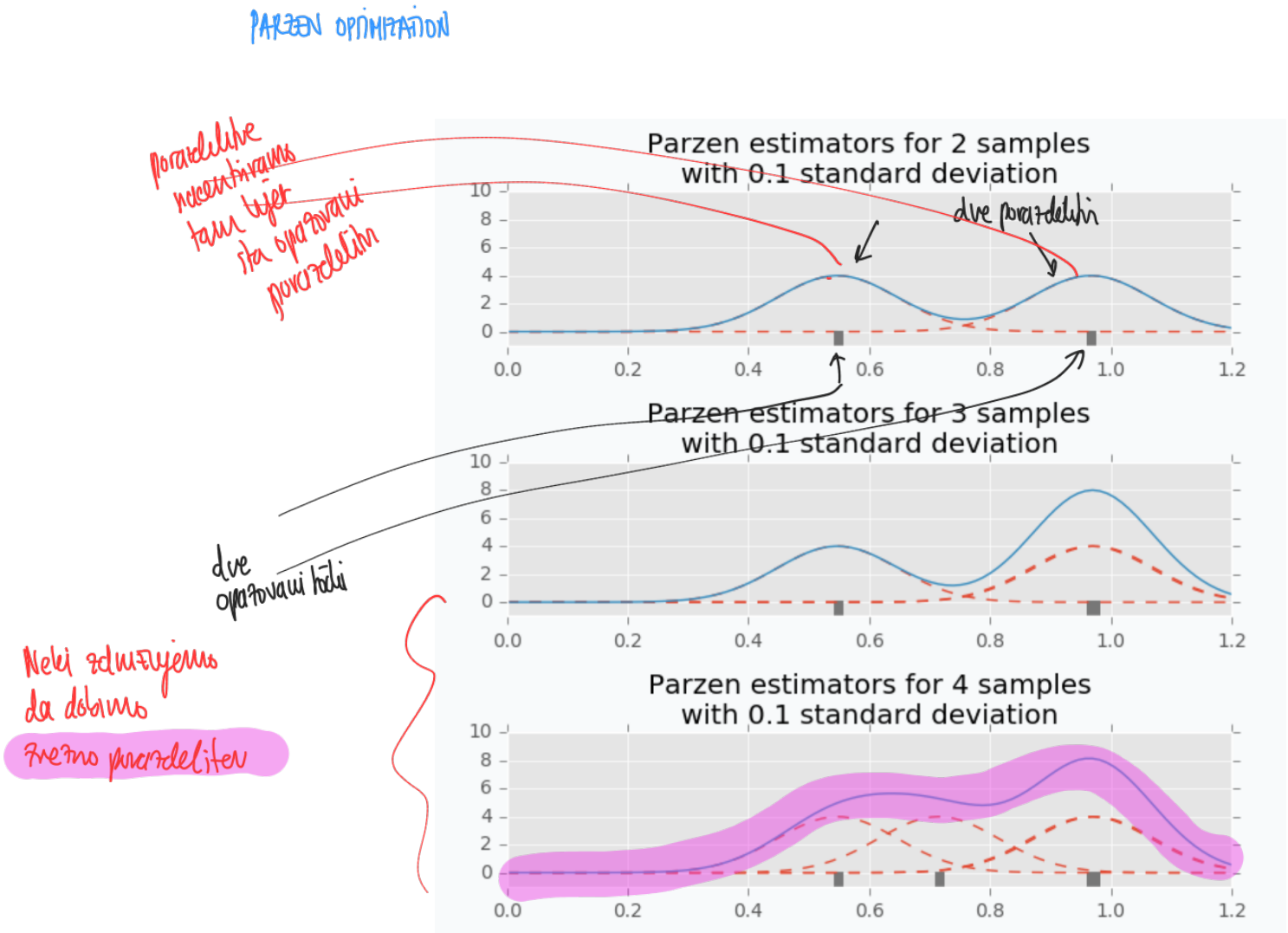
$$\begin{aligned} EI(\theta) &= \frac{\gamma p^* P_d(\theta) - P_d(\theta)\int_{\infty}^{p^*} p\, P(p)\, dp}{\gamma\, P_d(\theta) + (1-\gamma)\, P_s(\theta)} \\ &= \frac{\gamma p^* - \int_{\infty}^{p^*} p\, P(p)\, dp}{\gamma + (1-\gamma)\,\frac{P_s(\theta)}{P_d(\theta)}} \\ &\propto \left(\gamma + (1-\gamma)\,\frac{P_s(\theta)}{P_d(\theta)}\right)^{-1} \end{aligned}$$

to bo nelej popravlu

Izbira najbolj obetavne vrednosti θ



Nadomestni modeli so enostavne Gaussove kombinacije



Določanje apriornih porazdelitev

Za različne tipe nad-parametrov

- Izbira diskretne vrednosti iz seznama dopustnih vrednosti
- Izbira celoštevilčne vrednosti iz intervala $[0, max]$

- Izbira numerične vrednosti iz podane porazdelitve
- Porazdelitev je lahko enakomerna, normalna, logaritmična

Drevesna struktura soodvisnosti parametrov


```
hp.choice('classifier_type', [
  {
    'type': 'naive_bayes',
  },
  {
    'type': 'svm',
    'C': hp.lognormal('svm_C', 0, 1),
    'kernel': hp.choice('svm_kernel', [
      {'ktype': 'linear'},
      {
        'ktype': 'RBF',
        'width': hp.lognormal('svm_rbf_width', 0, 1)
      }
    ])
  },
  {
    'type': 'dtree',
    'criterion': hp.choice('dtree_criterion', ['gini', 'entropy']),
    'max_depth': hp.choice('dtree_max_depth', [None, hp.qlognormal('dtree_max_depth_int', 3, 1, 1)]),
    'min_samples_split': hp.qlognormal('dtree_min_samples_split', 2, 1, 1)
  }
])
```

Optimization

$$x^* = \arg\min_{x \in \mathcal{X}} F(x)$$

Assumptions on the objective function $F : \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^k$

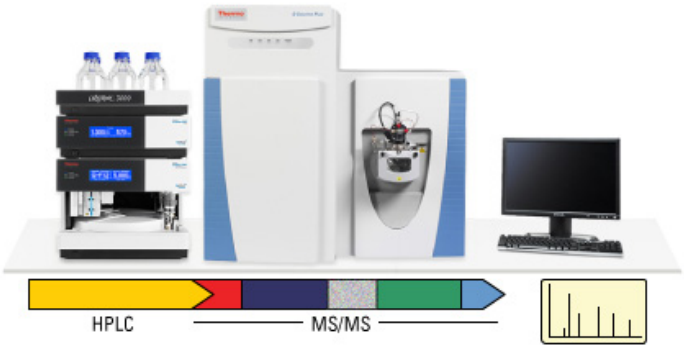
- Can be evaluated at arbitrary query point $x \in \mathcal{X}$
- Black-box function with no (simple) closed form

Problem

Limited resources for evaluating the objective F.

Limited Resources for Objective Evaluation

Expensive evaluation



- Ks of $$ for each data point
- Limited number of evaluations

Computationally complex evaluation

- Hours/days of CPU time
- Unlimited number of evaluations

E.g. tuning parameters/structure of a neural network.

Idea: Replace the Objective F with a Surrogate P

Use machine learning to train $P : \mathcal{X} \to \mathbb{R}$

kaj mora veljat za P? / dobra aproksimacija funkcije F
\ hitro računljiva funkcija

Desired properties of P

- Good approximation of F
- Much (orders of magnitude) more computationally efficient

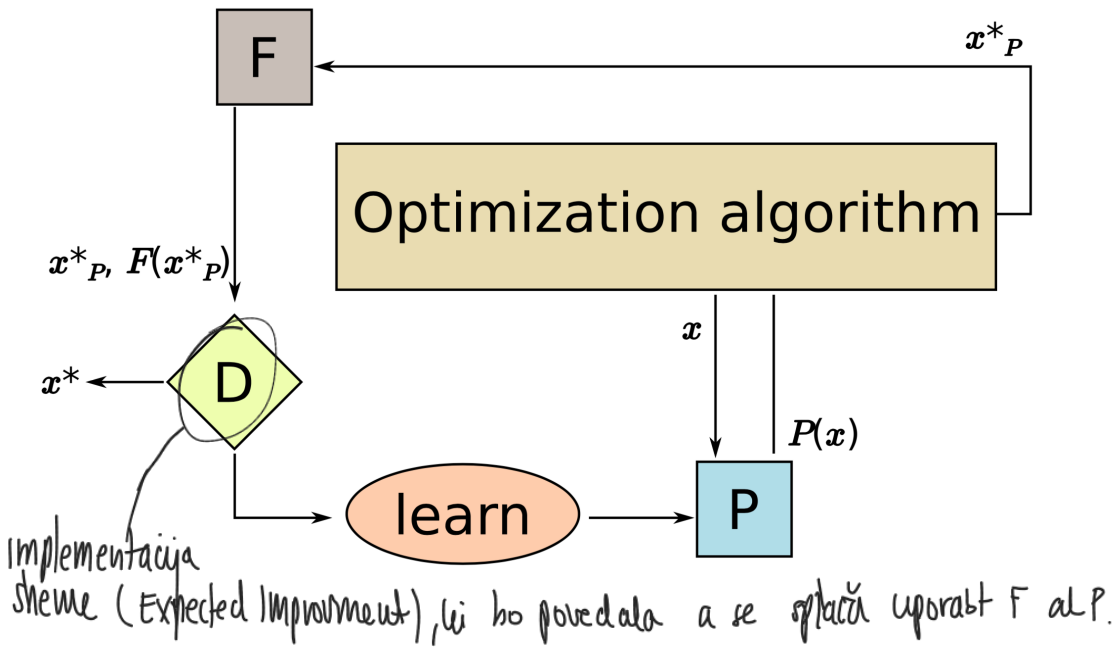
Issues addressed in this talk

- Surrogate training: how to learn and maintain efficient P?
- Substitution strategy: when to substitute F with P?

Dva razreda pristopov

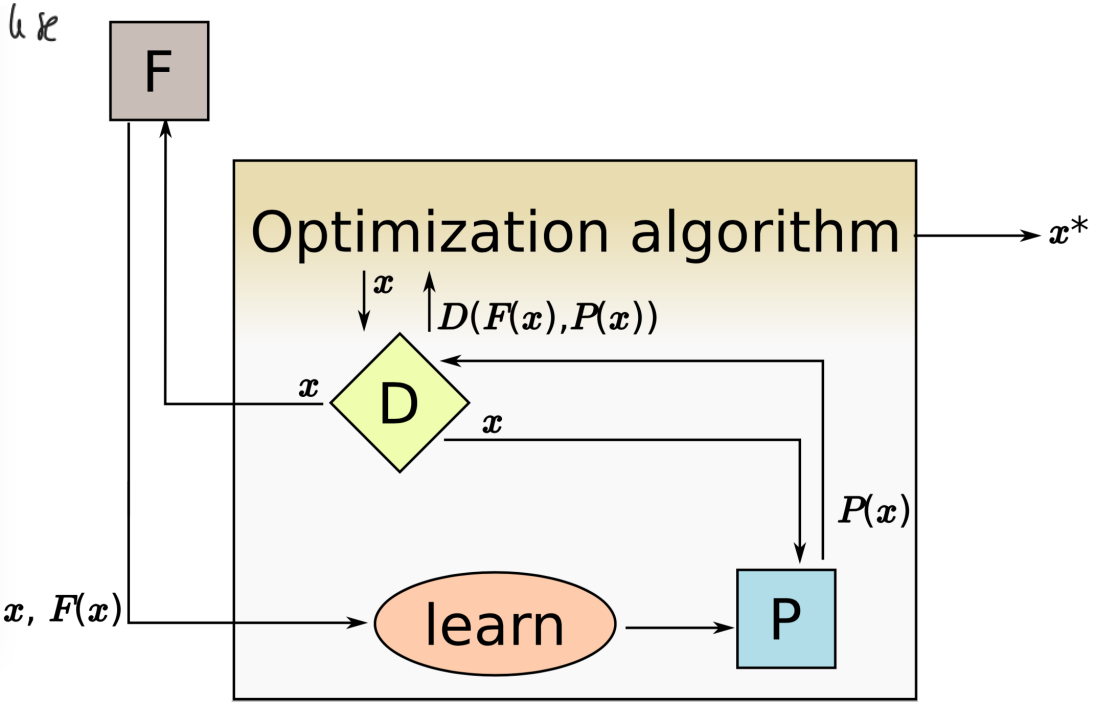
- Ovojnica, *Wrapper*
- Gnezdenje *Embedded*

Wrapper Approaches



- Surrogate training part of the wrapper
- Substitution strategy D fixed: only wrapper can evaluate F, the optimization algorithm evaluates only the surrogate P
- Sequential Model-Based Optimization, SMBO (Jones et al. 1998) and its variants, COBRA (Regis 2013; Bagheria et al. 2015)

Embedded Approaches



- Surrogate training embedded in the optimization algorithm
- Substitution strategy D fixed and also embedded
- Surrogate variants of the optimization algorithms (Das et al. 2016)

Pros and Cons Summary

Wrapper approaches

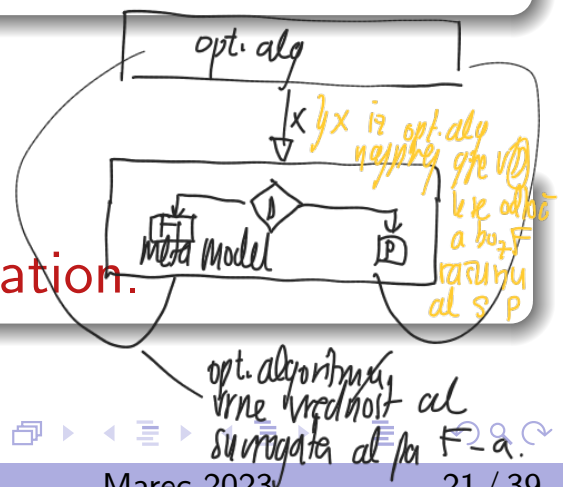
- ⊕ Can be coupled with an arbitrary optimization algorithm
- ⊖ Have inflexible substitution strategy

Embedded approaches

- ⊕ Have flexible substitution strategy
- ⊖ Require reimplementation of the optimization algorithm

Our Approach: Meta-Model Framework

Flexible substitution strategy, no need for reimplementation.



The Idea

- Encapsulate F, P and D into a single entity (meta model)
- Learn both the surrogate P and the substitution strategy D

- The optimization algorithm interacts with the meta model only
- The meta model autonomously decides whether to use F or P

The Meta-Model Structure

- $F : \mathcal{X} \to \mathbb{R}$ the objective function, $\mathcal{X} \subseteq \mathbb{R}^k$
- $P : \mathcal{X} \to \mathbb{R}$ the surrogate with training procedures
- $D : \mathcal{X} \to \{0, 1\}$ the substitute strategy with training procedures
- h: history of evaluations of F and P, sequence of triplets $(x_r \in \mathcal{X}, m_r = MetaModel(x_r), d_r = D(x_r))$

$$MetaModel(x) = \begin{cases} F(x) & ; D(x) = 1 \\ P(x) & ; D(x) = 0 \end{cases}$$

Training the Surrogate

Train set

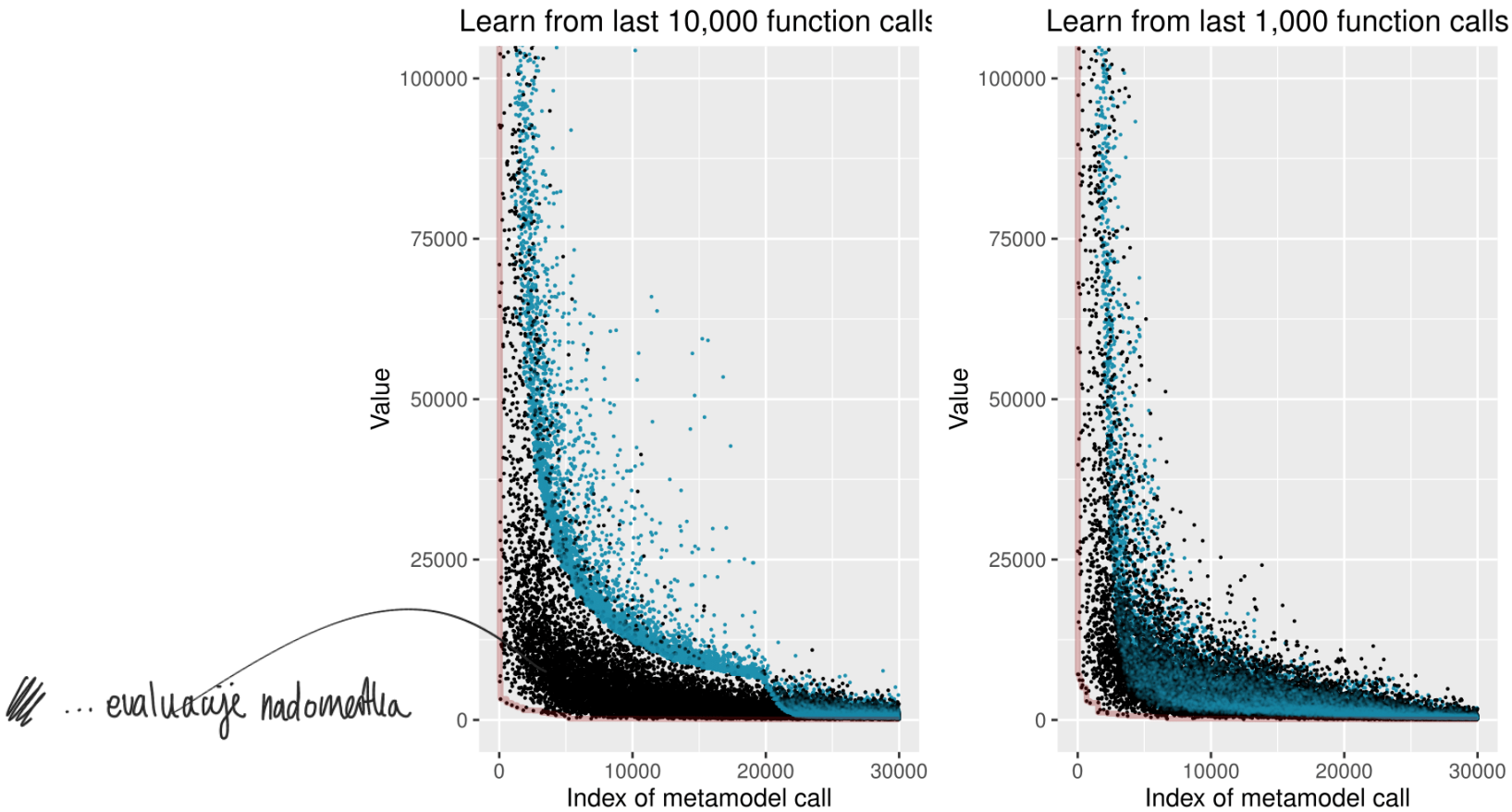
- Examples: based on the history of evaluations $(x_r, m_r, d_r) \in h : d_r = 1$
- The values of the k input variables: x_r
- The value of the target var: $MetaModel(x_r) = F(x_r)$, since $d_r = 1$

Learning algorithm and its output

- Any regression algorithm
- Model predicting $F(x)$ for a given x

Fading Memory Surrogate

Learning on recent examples improves predictions and saves time.



Training the Substitution Strategy (Relevator)

↳ uči se katero točko nej ovrednoti s pravo ciljno funkcijo (F) al samo z nadomestitvijo

Design decision

If the query point x is close to the optimum, evaluate F (and not P).

Thus, the relevance of the query point

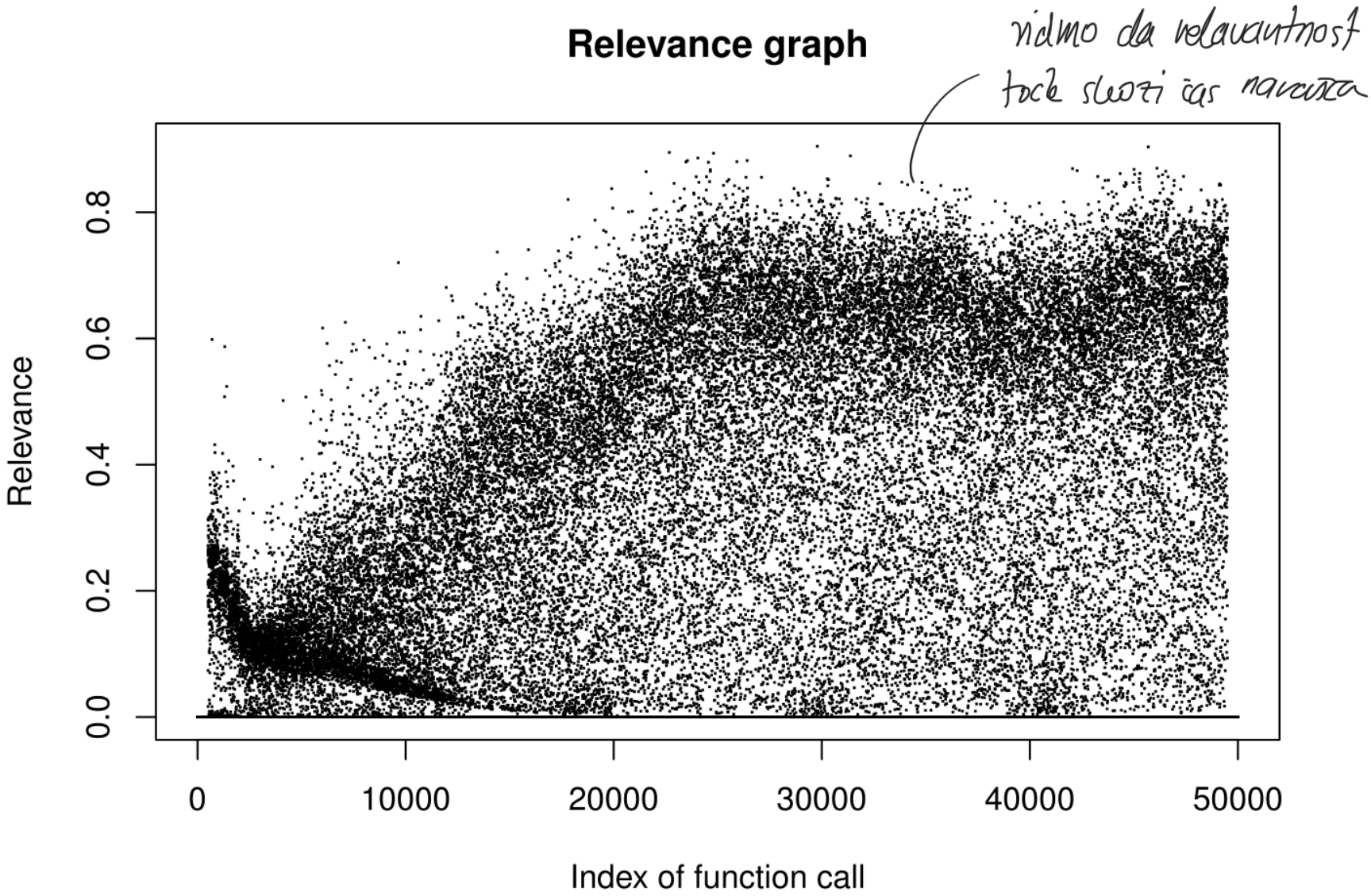
$$Relevance(x) = \begin{cases} \left(1 + \frac{F(x) - f_{min}}{f_{avg} - f_{min}}\right)^{-1} & ; F(x) \geq f_{min} \\ 1 & ; F(x) < f_{min} \end{cases}$$

where f_{min} and f_{avg} are the **current** minimum and average values of F in h

Train set and output

- Examples: based on the history of evaluations of F
- Values of input variables x_r, target values $Relevance(x_r)$
- Model predicting $Relevance(x)$ for a given x

Relevance Predictions



From Predicted Relevance to Decision Function

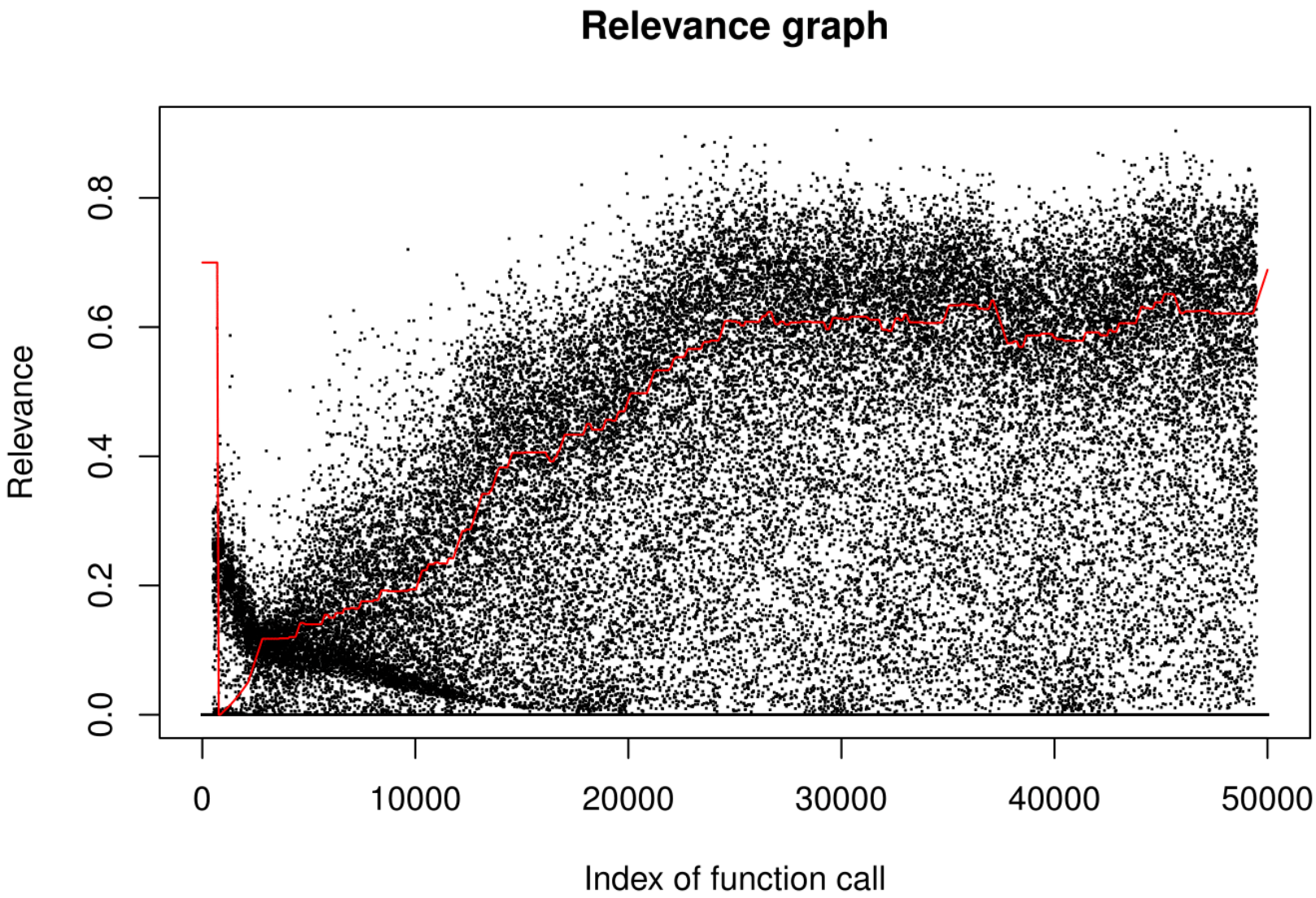
$$D(x) = \begin{cases} 1 & ; Relevance(x) \geq T(h) \\ 0 & ; Relevance(x) < T(h) \end{cases}$$

Replacement Rate RR

$$RR = \frac{|\{(x_r, m_r, d_r) \in h : d_r = 0\}|}{|h|}$$

The value of T dynamically adjusted to maintain desired value of RR.

Dynamic Relevance Threshold



Experiments on Synthetic Benchmarks

45 Benchmarks

COCO platform for comparing optimization methods (Hansen et al. 2016)

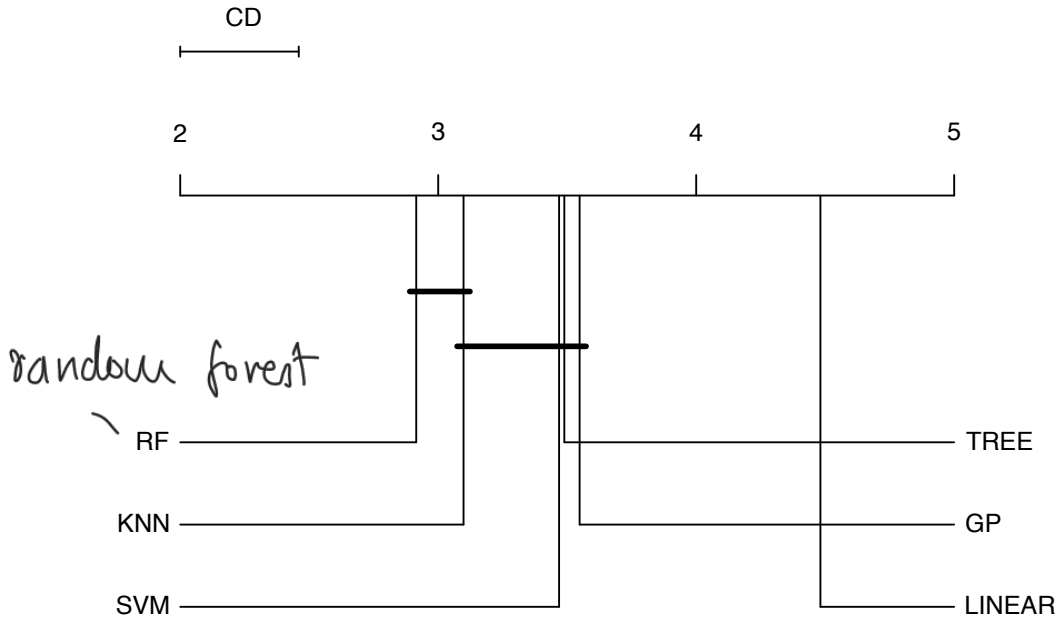
6 learning methods, 36 meta-model variants

- linear regression (LINEAR) and nearest neighbors (KNN)
- regression trees (TREE) and random forests (RF)
- Gaussian processes (GP) and support vector machines (SVM)

Optimization algorithm and performance measure

- Differential evolution
- Rate of substitution of the objective with the surrogate

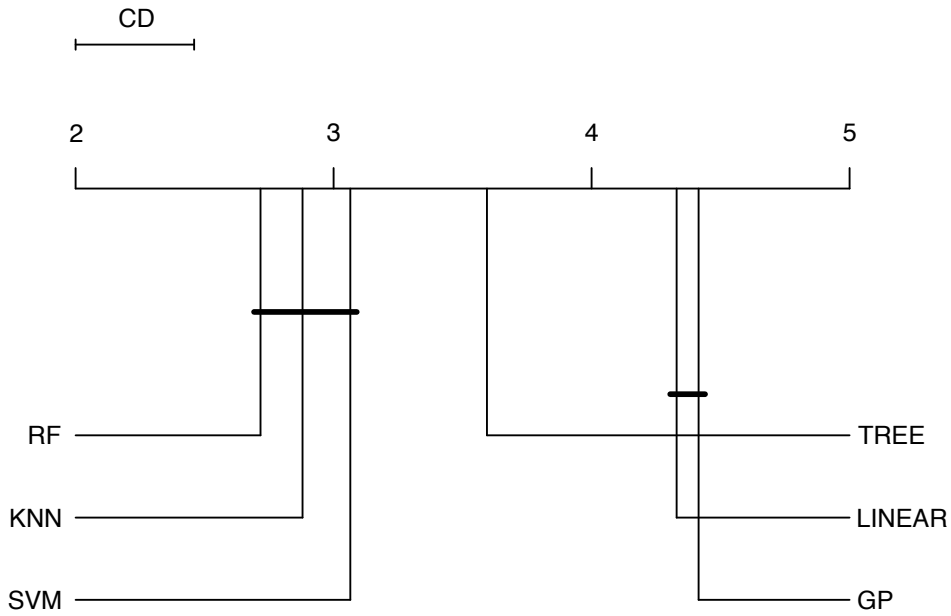
The Impact of the Surrogate



Meta model **robust** to the surrogate choice

Only the linear surrogate significantly worse than the alternatives.

The Impact of the Relevator



Meta model **sensitive** to the relevator choice

Three relevators, RF, KNN and SVM, significantly better than the others.

Experiments on Real Problems

Three real problems

Estimating parameters of three models dynamical systems from data.

15 meta-model variants

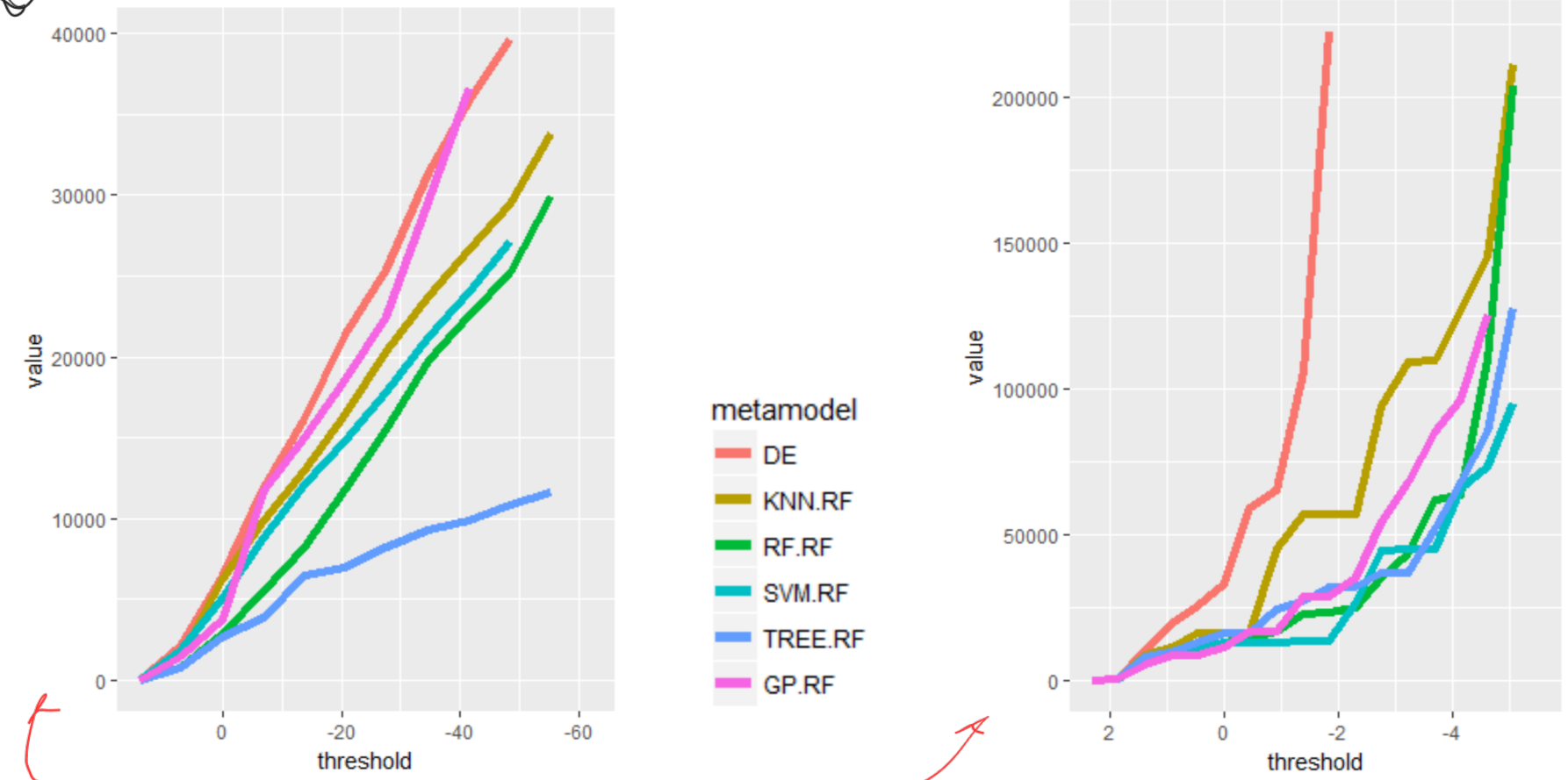
- 5 surrogate methods: all but linear regression
- 3 relevator methods: RF, KNN and SVM

Optimization algorithm and performance measure

- Differential evolution
- Convergence curves and substitution rate

Inverse Convergence Curves

koliko rabi vrednotenj funkcije, da dosežeš optimalnost?



vidimo da je osnovni algoritem vedno slabši

Convergence Curves: Significant Improvements over DE

Page's trend test of the convergence behavior

p-values indicate the significance of the increase of difference between the plain and surrogate convergence curves with the number of evaluations.

$\frac{P\rightarrow}{\downarrow D}$	TREE	KNN	GP	SVM	RF
KNN	**3.69e-3**	**3.04e-5**	0.504	0.372	**5.87e-6**
SVM	0.399	0.437	0.644	0.528	0.704
RF	**5.82e-6**	**4.09e-13**	**4.98e-3**	**1.26e-8**	**4.26e-9**

- Random Forest (RF) best relevator with arbitrary surrogate
- Surrogates based on GP and SVM inferior

Substitution Rates

Meta-model variant	P_1	P_2	P_3	Average
S = TREE, D = RF	0.73	0.72	0.86	0.77
S = RF, D = RF	0.36	0.77	0.89	0.67

- Up to 77% overall average substitution rate on the three problems
- Up to 89% substitution rate on individual problems

Central Contribution

New Paradigm:
Allows for a new, seamless method for coupling surrogates with an arbitrary state-of-the-art optimization method (stochastic or deterministic).

Further/ Ongoing Work

- Generality of results: other optimization algorithms
- Multi-objective optimization
- Constrained optimization
- Combinatorial optimization

Literatura in praktični napotki

Priporočena literatura

- (Lukšič 2017): magistrska naloga
- (Lukšič in ost. 2019): 10.1109/ACCESS.2019.2959846

Programska oprema

github.com/zigaLuksic/glitch-doctor