AutoML in optimizacija z nadomestki

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

Marec 2019

AutoML: optimalna konfiguracija algoritma A

$$\max_{\theta \in \Theta} p_A(\phi, S)$$

Poseben problem numerične optimizacije

- Časovno (pre)zahtevna ciljna funkcija
- Imamo na voljo le omejeno število izračunov ciljne funkcije

- Ciljno funkcijo nadomestimo z napovednim modelom
- Ideja nadomestka (*surrogate*)

Pregled vsebine

AutoML

- Na modelih temelječa optimizacija (SMBO)
- Izboljšava in pričakovana izboljšava
- Nadgradnje SMBO

Nadomestki

- Različni pristopi optimizacije z nadomestki
- Meta model za nadomestke
- Empirična raziskava uporabnosti meta modela

Osnovni algoritem SMBO: vhodi in izhodi

SMBO: Sequential Model-Based Optimization

Zaporedna, na modelih temelječa optimizacija

Vhodi

- algoritem A in podatkovna množica S za katere iščemo optimalne vrednosti parametrov
- prostor možnih vrednosti parametrov Θ
- metoda za merjenje zmogljivosti $p_A : \Theta \times \mathcal{S} \rightarrow \mathbb{R}$
- algoritem A_s za učenje nadomestnega modela za p_A

Izhod

Optimalne vrednosti parametrov $\theta^* \in \Theta$

Optimizacijski problem in ciljna funkcija

$$\min_{\theta \in \Theta} p_A(\theta, S)$$

- Optimizacijski problem s časovno zahtevno optimizacijsko funkcijo
- p_A vključuje izvajanje 10-kratno prečno preverjanje na S

- Ne moremo si privoščiti veliko izračunov ciljne funkcije
- Zato uporabljamo nadomestno funkcijo

Osnovni algoritem in nadomestna funkcija

$\boldsymbol{\theta} = Sample(\Theta)$
$Evs = \emptyset$
repeat
    for θ **in** $\boldsymbol{\theta}$ **do**
        $p_\theta = p_A(\theta, S)$
        $Evs = Evs \cup \{(\theta, p_\theta)\}$
    $\theta^* = \arg\min_\theta p_\theta : (\theta, p_\theta) \in Evs$
    $m = A_s(Evs)$
    $\boldsymbol{\theta} = Select(m, \Theta, \theta^*, Evs)$
until $Terminate()$
return θ^*

Ključne komponente

1. *Sample*: začetni vzorec točk za izračun ciljne funkcije
2. *Learn*: učenje modela z algoritmom A_s
3. *Select*: izbira naslednjega vzorca točk za izračun
4. *Terminate*: ustavitveni kriterij

Komponenta *Sample*

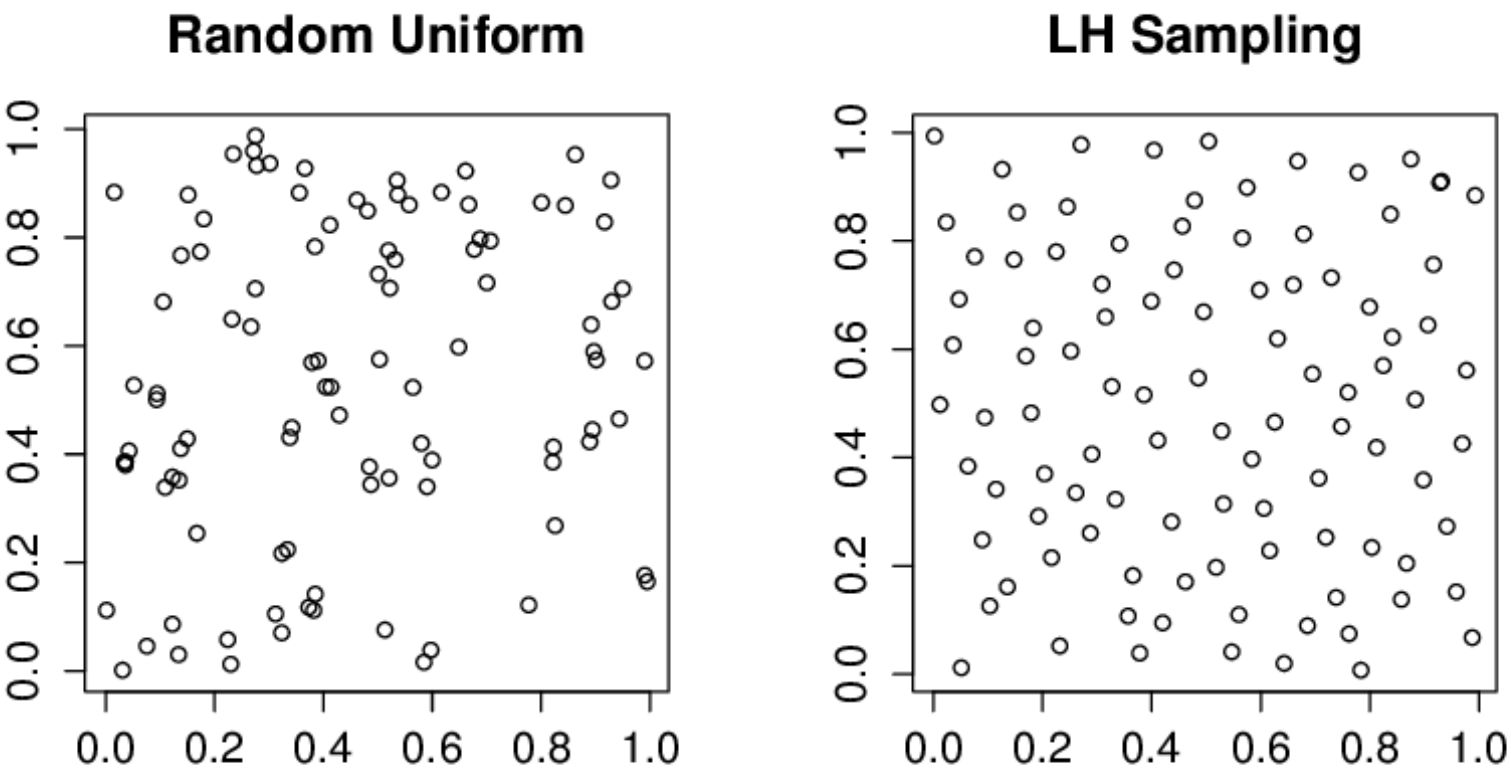
Pogosto poimenovana začetni dizajn (*initial design*)

Naključni vzorec točk $\theta \in \Theta$

- Naključno vzorčenje
- Vzorčenje po latinskem kvadratu (latinski hiper-kocki)

Naključno vzorčenje in latinska hiper-kocka

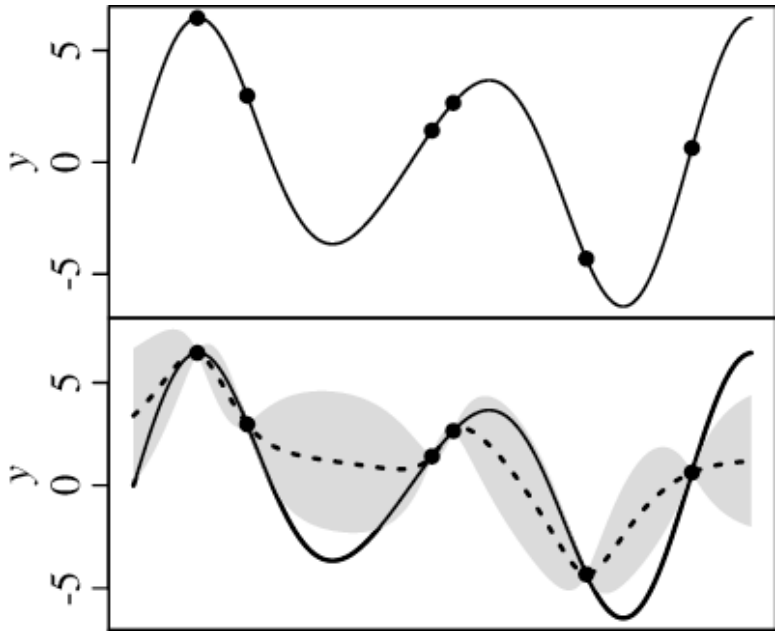
Prekletstvo dimenzionalnosti (prva predavanja pri osnovnem predmetu).



Komponenta *Learn*: nadomestni verjetnostni modela m

Verjetnostni model za regresijo

- Napoveduje pričakovano vrednost ciljne spremenljivke μ_θ
- IN zanesljivost napovedi: interval zaupanja ali odklon σ_θ
- Najbolj pogosto uporabljen A_s so Gaussovi procesi



Komponenta *Select*: Pričakovana izboljšava

Izboljšava

$$I(\theta) = \max(p_{\theta^*} - p_\theta, 0)$$

- θ^* je trenutna minimalna vrednost
- p_θ ne poznamo, zato ga ocenimo z verjetnostnim modelom m
- m vrne oceno napovedi μ_θ in odklon σ_θ

Pričakovana izboljšava (*Expected Improvement*)

$$EI(\theta) = E[W = \max(p_{\theta^*} - Y, 0)]$$

Y je naključna spremenljivka s povprečjem μ_θ in varianco σ_θ

Komponenta *Select*: Izračun pričakovane izboljšave

Predpostavimo, da je $Y \approx N(\mu_\theta, {\sigma_\theta}^2)$

$$EI(\theta) = (p_{\theta^*} - \mu_\theta)\,\Phi_Y(p_{\theta^*}) + {\sigma_\theta}^2 \phi_Y(p_{\theta^*})$$

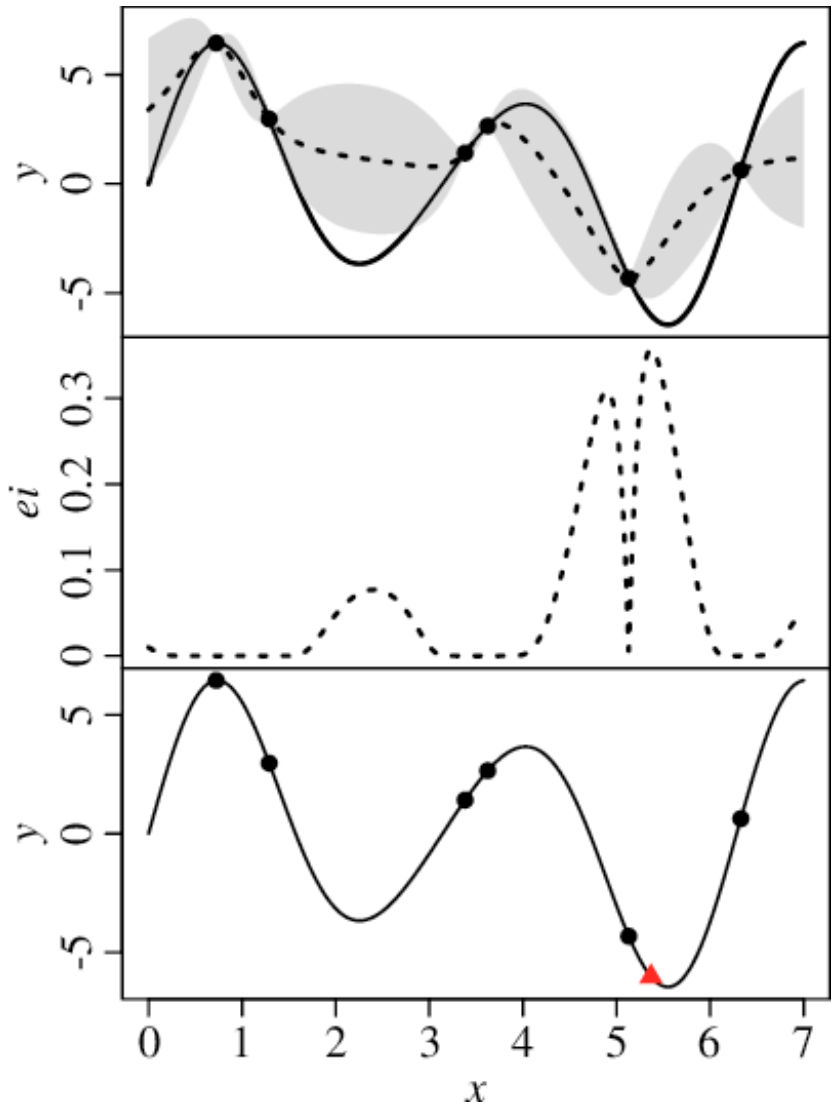
Φ_Y, ϕ_Y: porazdelitvena funkcija in funkcija gostote za $N(\mu_\theta, {\sigma_\theta}^2)$

Kaj pa Φ_Y in ϕ_Y?

$$u = \frac{y - \mu_\theta}{\sigma_\theta}$$

- $\Phi_Y(y) = \Phi(u)$, kjer je Φ porazdelitvena funkcija za $N(0,1)$
- $\phi_Y(y) = \phi(u)/\sigma$, kjer je ϕ funkcija gostote za $N(0,1)$

Komponenta *Select*: Največja pričakovana izboljšava



Komponenta *Select*: Optimizacijski problem

$$\max_{\theta \in \Theta} EI(\theta)$$

- Ciljna funkcija *EI* nezahtevna za izračun (nadomestni model m)
- Zato lahko uporabimo poljubno optimizacijsko metodo
- Pogosto uporabljena metoda: sistematični vzorec (rešetka, *grid*)

Komponenta *Terminate*

Ustavitveni kriterij

- Skoraj vedno je to omejitev uporabljenega časa
- Pravzaprav omejitev števila izračunov ciljne funkcije

Smeri nadgradnje

1. Stohastične ciljne funkcije: večkratni izračuni za podan θ
2. Več podatkovnih množic namesto ene
3. Diskretni parametri

Stohastične ciljne funkcije: SKO

SKO: Sequential Kriging Optimization

Zaporedna optimizacija z regresijo zasnovano na Gaussovimi procesi

Uporabljene komponente

- *Sample*: za *izbrane* začetne točke opravi večkratne izračune
- *Learn*: nadomestni model GP predpostavi *šum* v ciljni spremenljivki
- *Select*: optimizacija z Nelder-Mead metodo
- *Select*: pri izbiri se upošteva napoved plus ena standardni odklon
- *Select*: pričakovana izboljšava prilagojena tako, da je pristranska do točk z visoko varianco

Stohastične ciljne funkcije: SPO

SPO: Sequential Parameter Optimization

Zaporedna optimizacija parametrov

Uporabljene komponente

- *Sample*: za *vse* začetne točke opravi večkratne izračune
- A_s: nadomestni model GP predpostavi *odsotnost šuma*
- *Select*: sistematični vzorec za iskanje največje pričakovane izboljšave, izbira vnaprej določenega števila točk za izračun ciljne funkcije
- *Select*: skrbi tudi za izbiro točk za ponovne/večkratne izračune

Več podatkovnih množic: SMAC in ROAR

SMAC: Seqential Model-based Algorithm Configuration

- *Sample*: naključno vzorčenje brez latinskih kvadratov
- *Learn*: naključni gozd kot verjetnosti model: napovedi posameznih dreves uporabljeni za oceno μ_θ in σ_θ
- *Select*: izbira podatkovnih množic za vrednotenje p_A ob upoštevanju števila prejšnjih izračunov

ROAR: Random Online Aggressive Racing

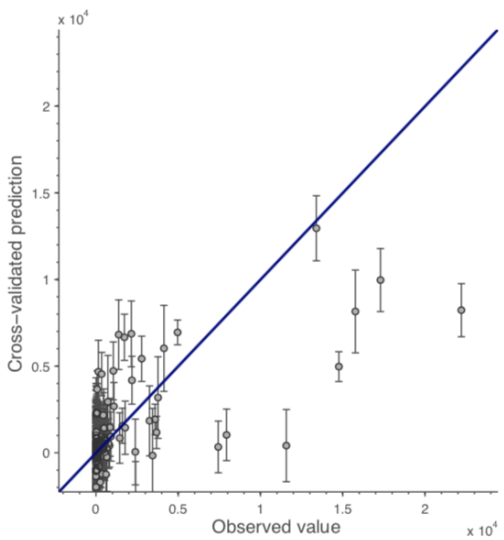
- Metoda brez modela, naključna izbira točk za izračun ciljne fukcije
- *Sample*: Izbira le ene naključne točke $\theta \in \Theta$
- *Select*: Naključna izbira ene točke
- Presenetljivo dobri rezultati

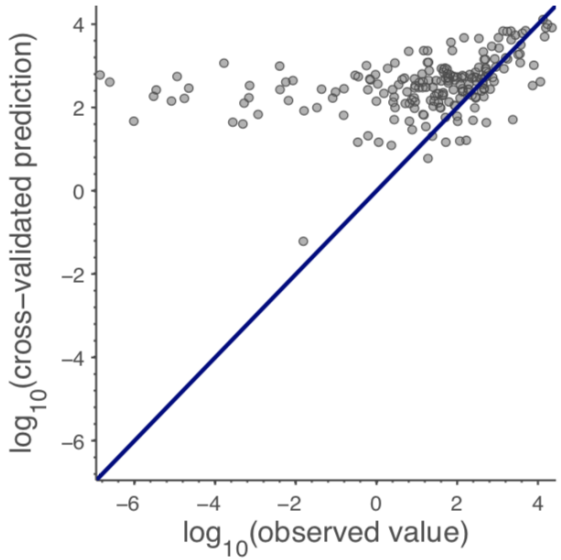
Različni tipi parametrov: Naključni gozd

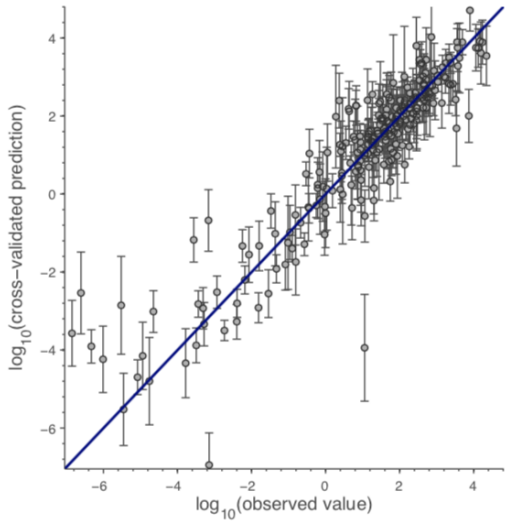
Modeli GP omejeni na numerične parametre

Learn: uporaba naključnega gozda odpravi to omejitev

Transformacija vrednosti ciljne funkcije (log)







Kaj pa meta učenje in meta modeli?

Uporaba meta modelov kot

- *Sample*: kriterij za izbiro začetnih točk
- *Select*: dodaten kriterij za izbiro nadaljnjih točk

V nadaljevanju meta model za nadomestne funkcije.

Literatura in praktični napotki

Priporočena literatura

- (Jones in ost. 1998): SMBO
- (Hutter in ost. 2009): SPO in SKO
- (Hutter in ost. 2011): SMAC in ROAR

Programska oprema in spletni viri

- R-paket *mlrMBO*, ki implementira orodja za SMBO
- Spletna stran *AutoML.org*

Optimization

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

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

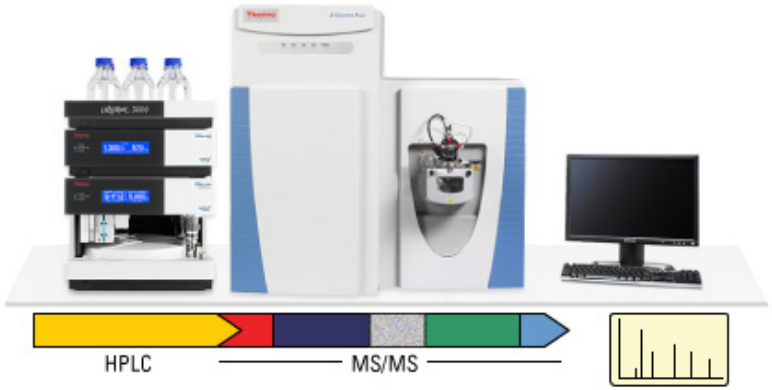
- Can be evaluated at arbitrary query point $x \in \mathcal{X}$
- Black-box function with no (simple) closed form

Problem

Limited resources for evaluating the objective F

Limited Resources for Objective Evaluation

Expensive evaluation



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

Computationally complex evaluation

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

Example

Tuning parameters/structure of a neural network

Idea: Replace the Objective F with a Surrogate P

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

Desired properties of P

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

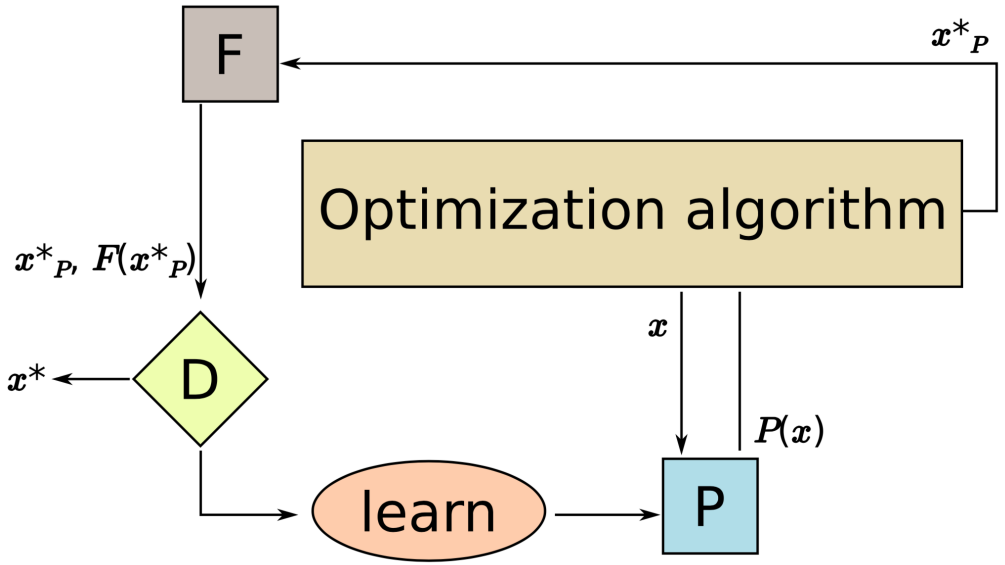
Issued addressed in this talk

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

Dva razreda pristopov

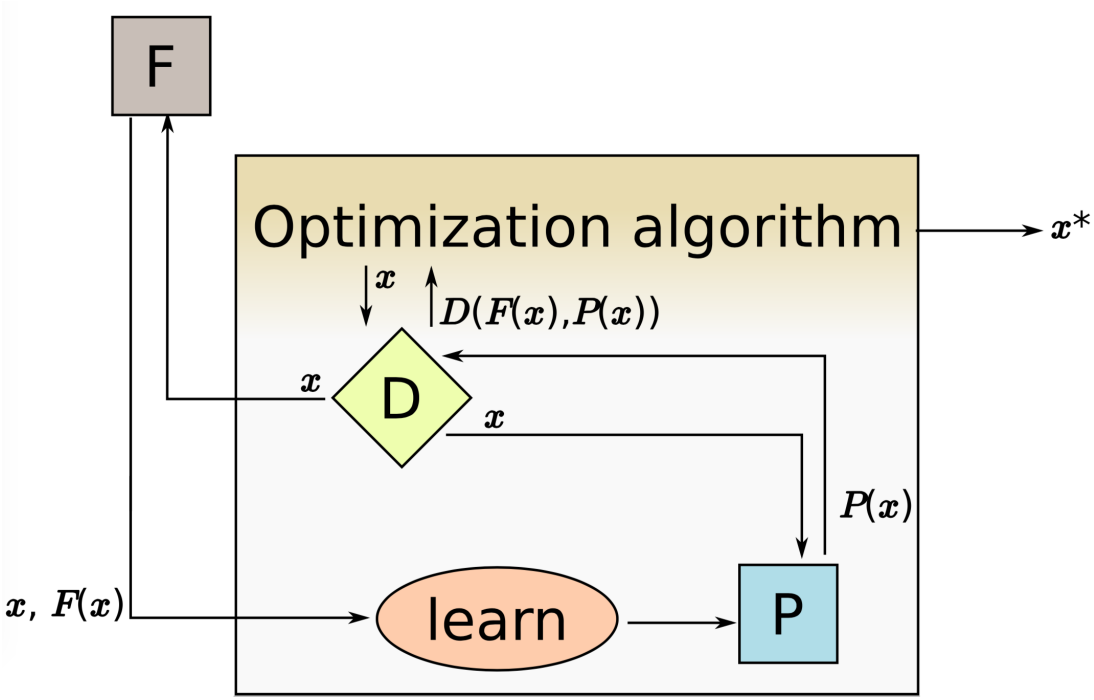
- Ovojnica (*Wrapper*)
- Ugnezdenje (*Embedded*)

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 (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 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 tuples $(x_r, m_r = MetaModel(x_r), d_r = D(x_r))$

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

Training the Surrogate

Train set

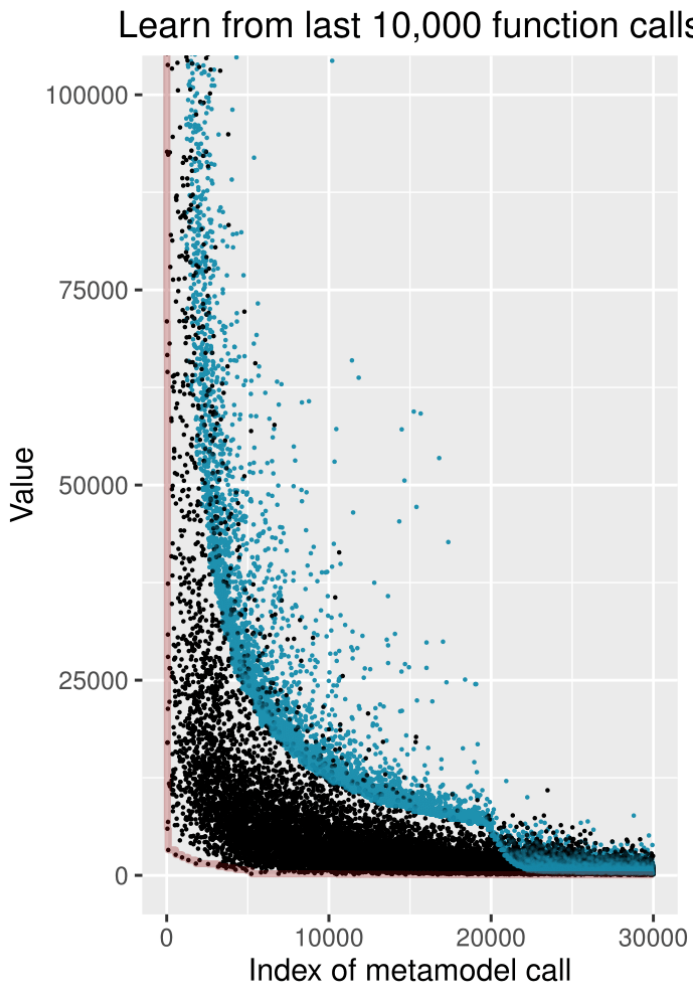
- Examples: based on the history of evaluations $(x_r, m_r, d_r) \in h : d_r = 1$
- Input values (k): x_r
- Target value: $MetaModel(x_r) = F(x_r)$

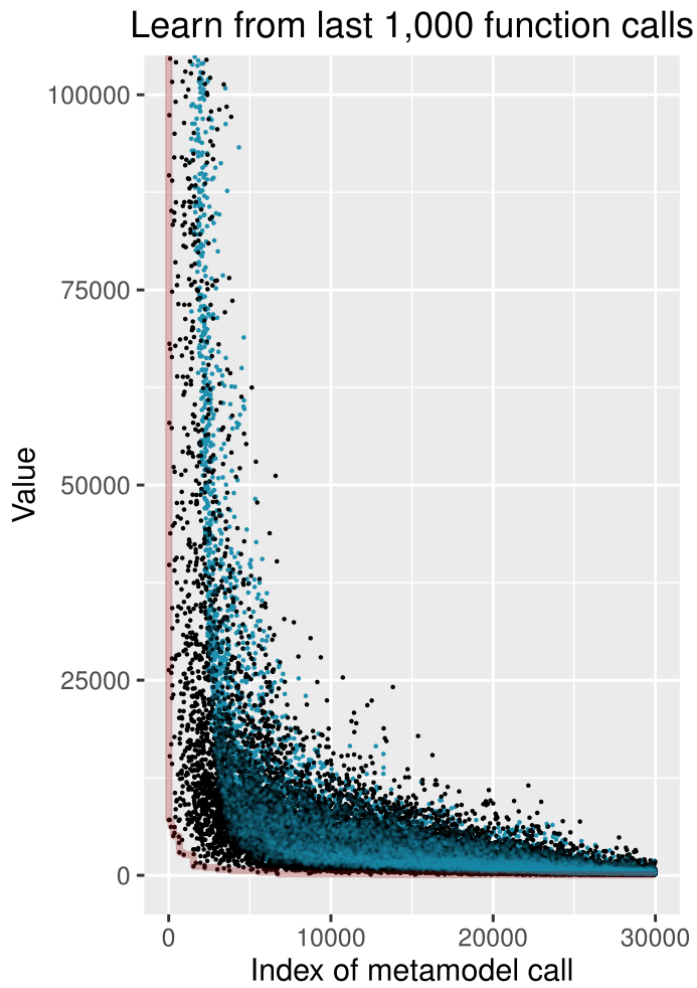
Learning algorithm and output

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

Fading Memory Surrogate

Learning on recent examples improve time and *predictive* performance.





Training the Substitution Strategy (Relevator)

Assumption

For query points close to the optimum: evaluate F and not P.

Thus, the relevance of the query point

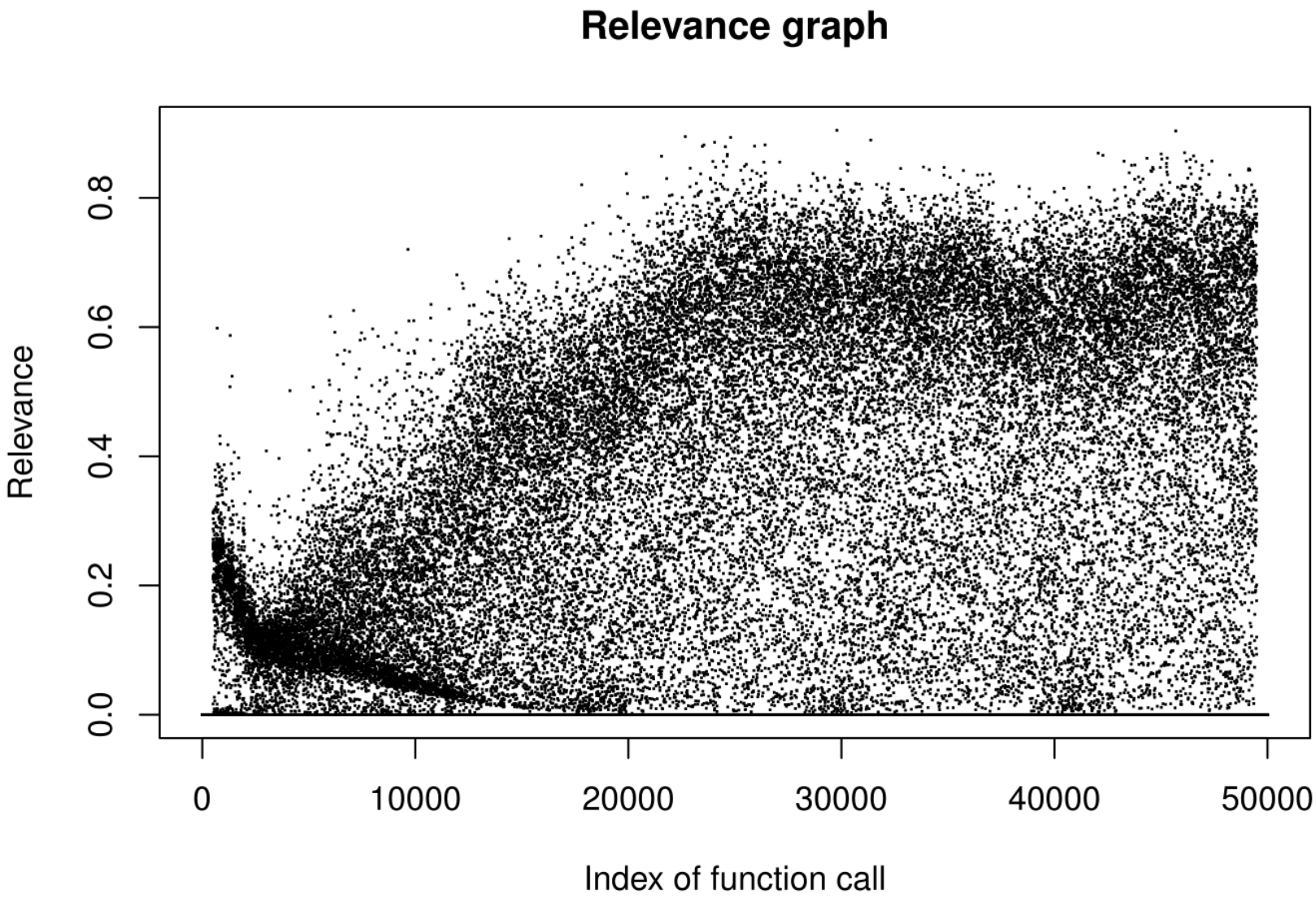
$$Relevance(x) = \begin{cases} \left(1 + \frac{F(x) - f_{min}}{f_{avg} - f_{min}}\right)^{-1} & ; F(x) \geq f_{min} \\ 1 & ; F(x) < f_{min} \end{cases}$$

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

Train set and output

- Examples: based on the history of evaluations of F
- Input values x_r, output values $Relevance(x_r)$
- Model predicting $Relevance(x)$ for a given x

Relevance Predictions



From Predicted Relevance to Decision Function

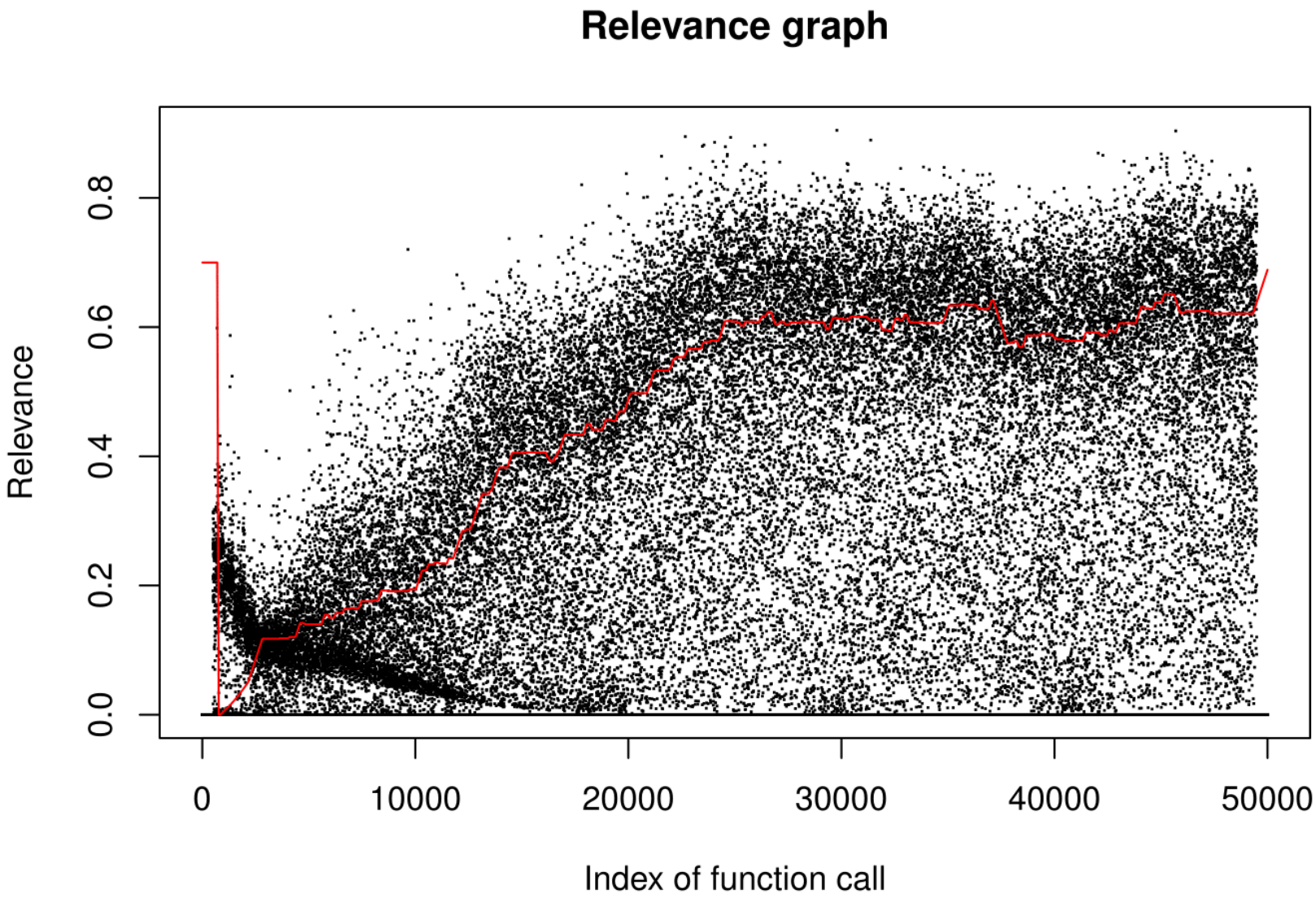
$$D(x) = \begin{cases} 1 & ; Relevance(x) \geq T(h) \\ 0 & ; Relevance(x) < T(h) \end{cases}$$

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 algorithms (Hansen et al 2016)

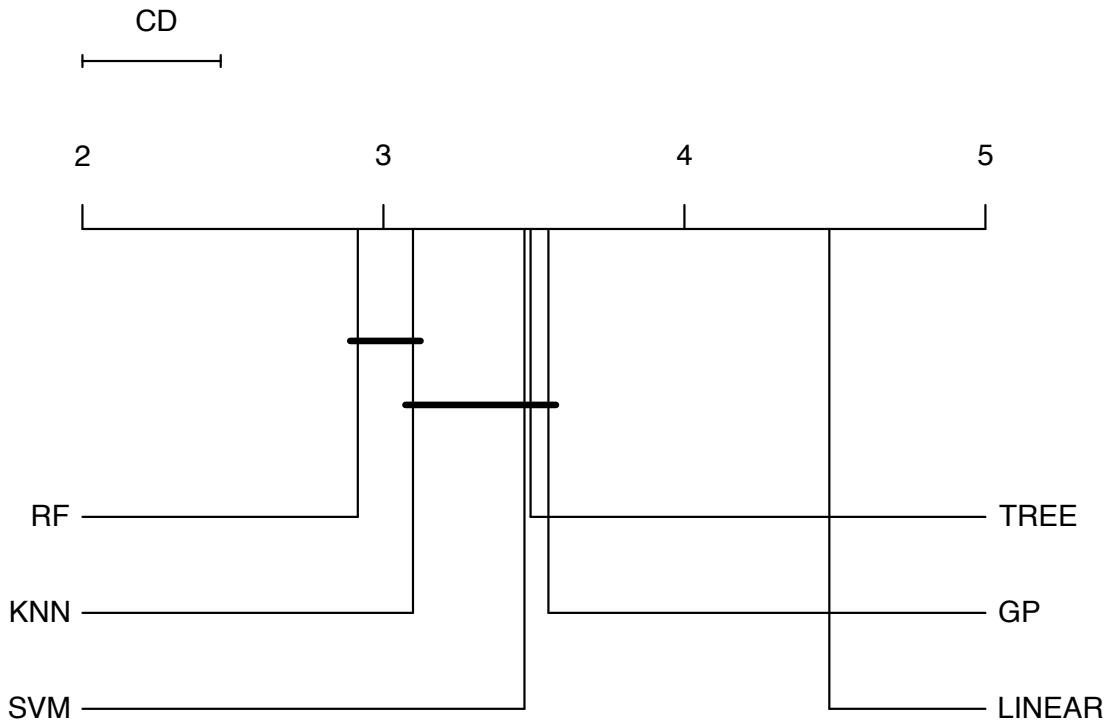
6 learning methods, 36 meta-model variants

- linear regression (LINEAR)
- 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

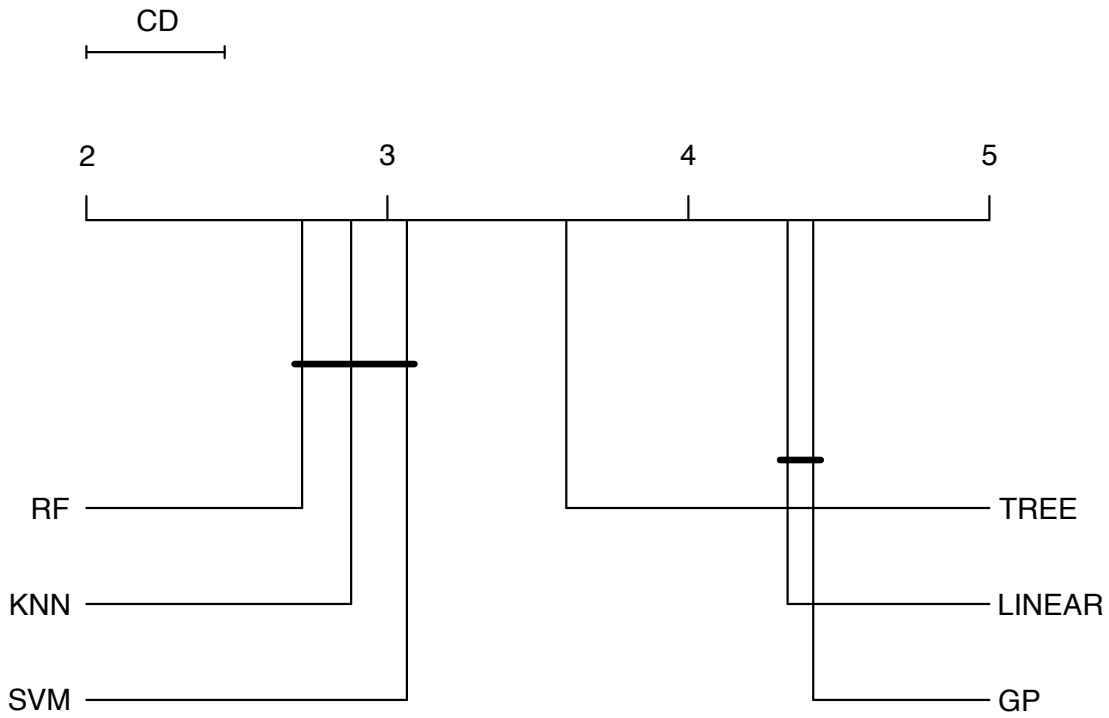
The Impact of the Surrogate



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

3 real problems

Estimating parameters of three models of biological dynamic systems from observation data.

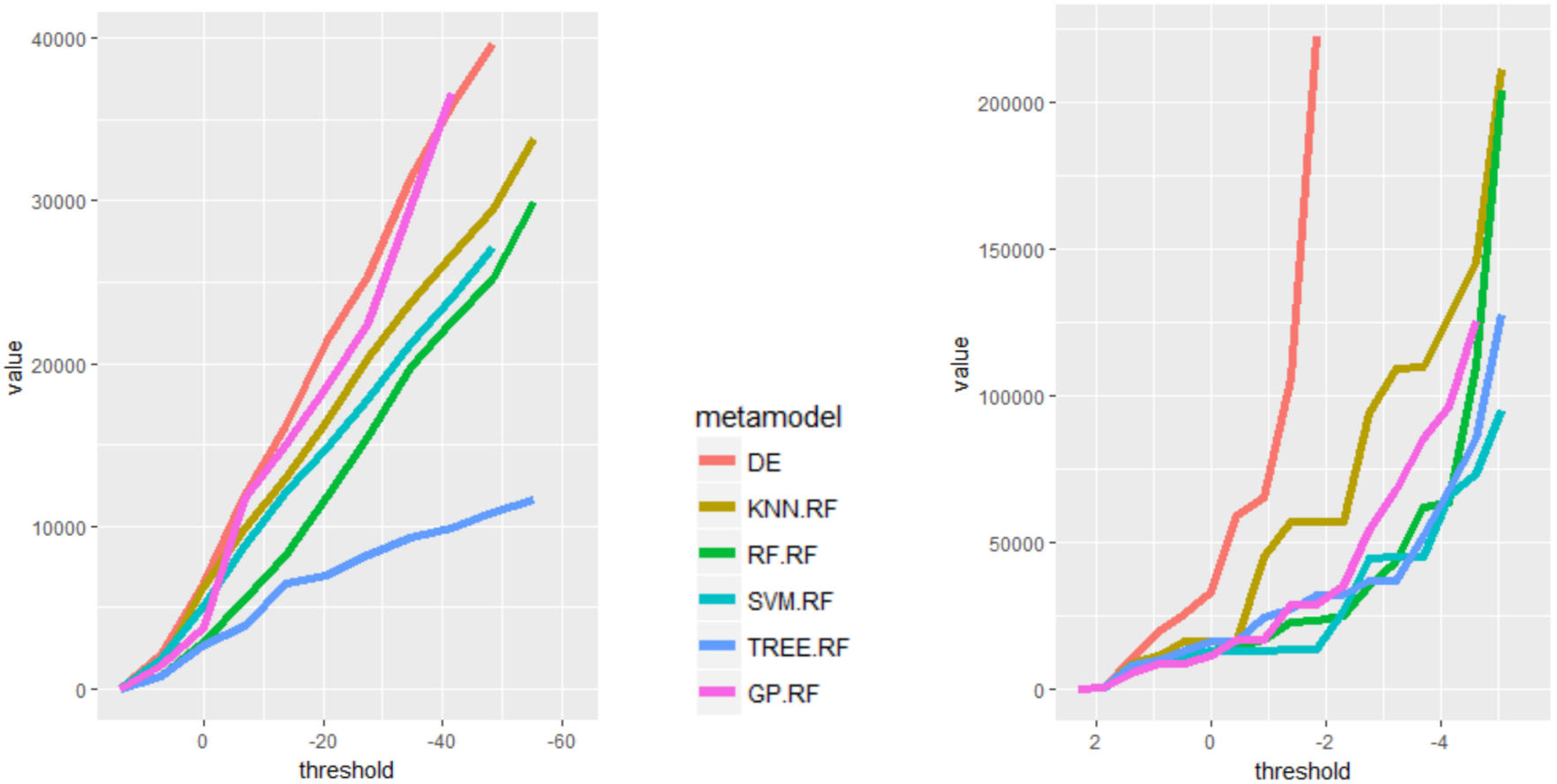
15 meta-model variants

- 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



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. 2017): konferenčni članek

Programska oprema

Žiga Lukšič: osebna komunikacija