Feature Selection with Monte-Carlo Tree Search

Robert Pinsler
20.01.2015

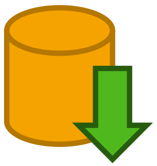
TECHNISCHE UNIVERSITÄT DARMSTADT

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

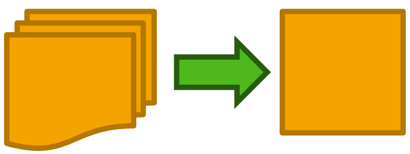
1. Feature Selection
2. Feature Selection as a Markov Decision Process
3. Feature UCT Selection
4. Experimental Validation
5. Summary and Outlook

Motivation



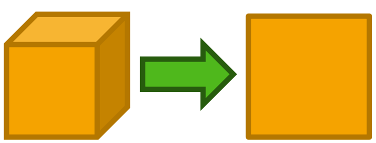
Less data

- less to store and collect
- faster to process



Reduced generalization error

- less noise (less irrelevant features)
- simpler hypothesis spaces (less redundant features)



Better understanding

- easier to understand
- easier to visualize

TECHNISCHE UNIVERSITÄT DARMSTADT

Supervised Approaches

Filter

- independently rank features with score function, select top n

no correlations *or* redundancy

Wrapper

- explore superset of feature, measure generalization error of all subsets
- whole combinatorial optimization problem

exploration vs. exploitation dilemma

Embedded

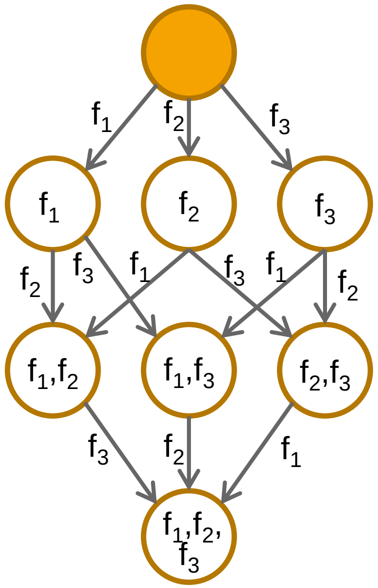
- combine feature selection and learning

no correlations *or* redundancy

FS as a Markov Decision Process

$$\mathcal{M} = (S, A, P, R)$$

$\mathcal{F}$	set of features plus *stopping* feature f_s
	final states: all states $F \subseteq \mathcal{F}$ containing f_s
$\mathcal{S} = 2^{\mathcal{F}}$	state space
$A = \{\text{add } f, f \in \mathcal{F}\}$	action space
$P : \mathcal{S} \times \mathcal{F} \times \mathcal{S} \mapsto \mathbb{R}^+$	transition function
	$P(F, f, F')$ is nonzero if $F' = F \cup \{f\}$
$V : \mathcal{S} \mapsto [0, 1]$	reward function (also denoted as R)
$\pi : \mathcal{S} \mapsto \mathbf{A}$	**policy**



Goal: find *optimal* policy

$$\pi^\star = \underset{\pi}{\operatorname{argmin}}\ \mathbf{Err}\left(\mathcal{A}\left(F_\pi\right)\right)$$

$\mathcal{A}(F \setminus \{f_s\})$	learned hypothesis based on F
$\mathbf{Err}(\mathcal{A}(F))$	generalization error of learned hypothesis

Finding an Optimal Policy

$$\pi^\star = \underset{\pi}{\operatorname{argmin}}\ \mathbf{Err}\left(\mathcal{A}\left(F_\pi\right)\right)$$

Following *Bellman's optimality principle*

$$V^\star(F) = \begin{cases} \mathbf{Err}(\mathcal{A}(F)) & \text{if } F \text{ is final} \\ \underset{f \in \mathcal{F} \setminus F}{min}\ V^\star(F \cup \{f\}) & \text{otherwise} \end{cases}$$

$$\pi^\star(F) = \underset{f \in \mathcal{F} \setminus F}{\operatorname{argmin}}\ V^\star(F \cup \{f\})$$

optimal, but *intractable* (state space exponential in #features)

Why not cast problem into 1-player game and use MCTS with UCT?

Feature Selection as a 1-Player Game

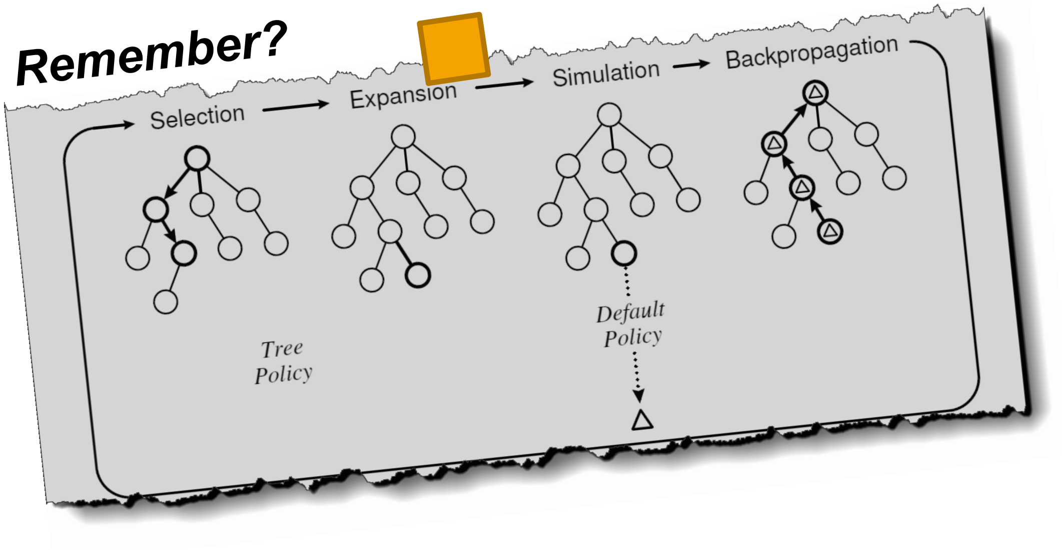
TECHNISCHE UNIVERSITÄT DARMSTADT

Formalize FS as
Markov Decision Process

MDP can be solved with
Reinforcement Learning

Cast problem as
1-player game

Use **MCTS** with **UCT**!



Restrict number of arms

TECHNISCHE UNIVERSITÄT DARMSTADT

UCB1-tuned instead of UCB1

limit exploration term by including *empirical variance of rewards*

T_F	no. of visits in node F
$t_{F,a}$	no. of times action a has been selected in F
c_e	exploration parameter
$\hat{\mu}_{F,a}$	average reward of a from F
$\hat{\sigma}^2_{F,a}$	empirical variance of rewards

$$a^* = \arg\max_{a \in A} \left\{ \hat{\mu}_{F,a} + \sqrt{\frac{c_e \ln(T_F)}{t_{F,a}} \min\left(\frac{1}{4}, \hat{\sigma}^2_{F,a} + \sqrt{\frac{2\ln(T_F)}{t_{F,a}}}\right)} \right\}$$

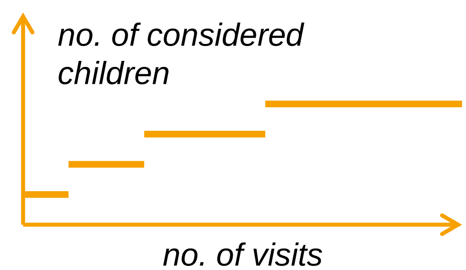
Continuous heuristic

set c_e to very small value

Discrete heuristic

consider only $\lfloor T_F^{\,b} \rfloor$ children (b < 1)

→ *progressive widening*



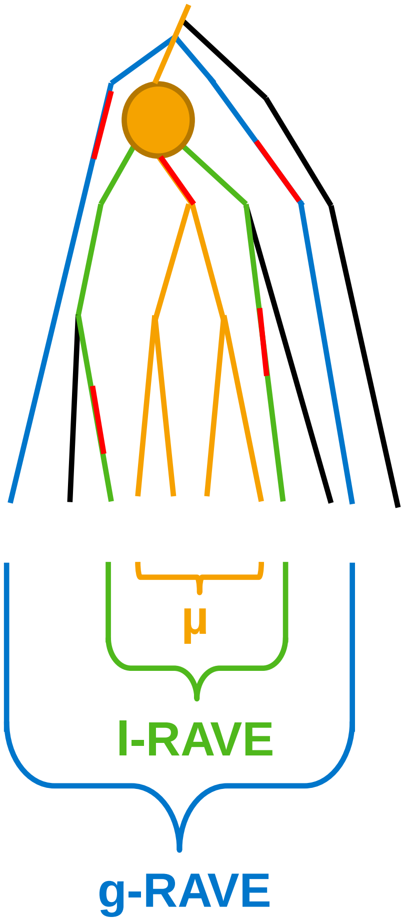
Rapid Action Value Estimation (RAVE)

AMAF heuristic
incorporate additional knowledge gained within search

$$\text{g-RAVE}_f = average\{V(F_t), f \in F_t\}$$
$$\ell\text{-RAVE}_{F,f} = average\{V(F_t), F \rightsquigarrow F_t, f \in F_t\}$$

associate *RAVE score* to each size of feature set:

$$\text{g-RAVE}_{f_s^{(d)}} = average\{V(F_t), |F_t| = d+1\}$$



Selection of New Nodes

Discrete heuristic
select top-ranked feature after RAVE whenever integer part of T_F^b is incremented

Continuous heuristic
replace *UCB1-tuned* formula by

$$(1-\alpha)\cdot\hat{\mu}_{F,f}+\alpha\left((1-\beta)\cdot\ell\text{-RAVE}_{F,f}+\beta\cdot\text{g-RAVE}_{f}\right)$$
$$+\sqrt{\frac{c_e\ln(T_F)}{t_{F,f}}\min\left(\frac{1}{4},\hat{\sigma}^2_{F,f}+\sqrt{\frac{2\ln(T_F)}{t_{F,f}}}\right)}$$

$\alpha=\frac{c}{c+t_{F,f}}$	impact of ℓ-RAVE
$\beta=\frac{c_l}{c_l+t_l}$	impact of g-RAVE
t_l	no. of iterations involved in ℓ-RAVE computation
$t_{F,f}$	no. of times feature f has been selected in F
c, c_l	parameter

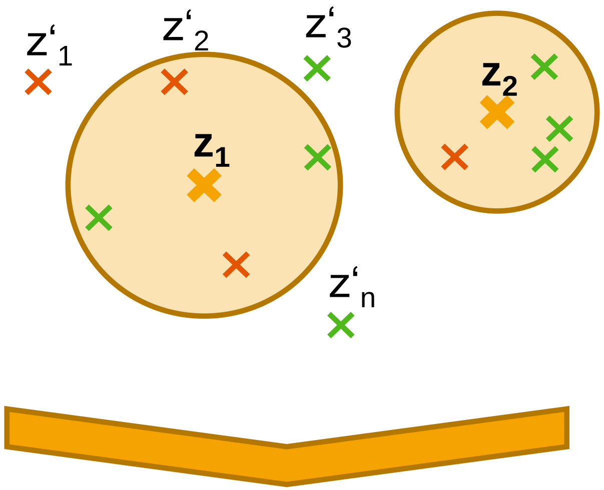
TECHNISCHE UNIVERSITÄT DARMSTADT

Instant Reward Function

k-nearest neighbor (k-NN)

$$s_F(z) = |\{z' \in \mathcal{N}_{F,k}(x),\ y' > 0\}|$$

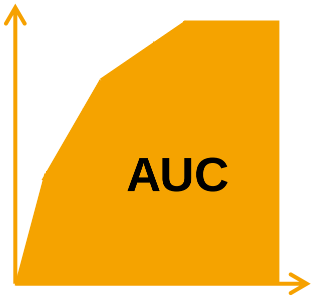
d_F	Euclidean distance based on features in F
$\mathcal{L}$	training set
$\mathcal{V}$	aggressive subsample of $\mathcal{L}$
$z = (x, y)$	labeled example in $\mathcal{V}$
$\mathcal{N}_{F,k}(x)$	set of k-NN of x in $\mathcal{L}$ after d_F
$s_F(z)$	number of positive examples among $\mathcal{N}_{F,k}(x)$



Area under the ROC curve (AUC) *

aka Mann Whitney Wilcoxon sum of ranks test

$$V(F) = \frac{|\{(z, z') \in \mathcal{V}^2,\ s_F(x) < s_F(x'),\ y < y'\}|}{|\{(z, z') \in \mathcal{V}^2,\ y < y'\}|}$$



* Note that 0 really is the minimum as we do not simply predict a class which we could change. Instead we want to find a feature set with minimum generalization error

Feature UCT Selection (FUSE)

FUSE

Input: number of iterations T and many-armed behavior MA
Output: search tree $\mathcal{T}$ and g-RAVE score
Initialize $\mathcal{T} \leftarrow \emptyset, \forall f$, g-RAVE(f) $= 0$
for $t = 1$ **to** T **do**
  Iterate($\mathcal{T}$, g-RAVE, $\emptyset$)
end for

Iterate_random

Input: search tree $\mathcal{T}$, score g-RAVE, subset F
Output: reward V
while $rand() < q^{|F|}$ **do**
  $f^\star \leftarrow$ uniformly selected feature in $\mathcal{F} \setminus (F \cup \{f_s\})$
  $F \leftarrow F \cup \{f^\star\}$
end while
$V \leftarrow V(F)$; Update g-RAVE

Iterate

Input: search tree $\mathcal{T}$, score g-RAVE, subset F
Output: reward V
if F final **then**
  $V \leftarrow V(F \setminus \{f_s\})$; Update g-RAVE
else
  if $t(F) \neq 0$ **then**
    if MA = progressive widening **then**
      $f^\star \leftarrow \underset{f \in AllowedFeatures(F)}{\text{argmax}}$ UCB1-tuned(F, f)
    else
      $f^\star \leftarrow \underset{f \in \mathcal{F} \setminus F}{\text{argmax}}$ tradeoff UCB/RAVE(F, f)
    end if
    $V \leftarrow iterate(\mathcal{T}$, g-RAVE, $F \cup \{f^\star\})$
  else
    $V \leftarrow iterate\_random(\mathcal{T}$, g-RAVE, $F)$
  end if
  Update T_F, t_f, $\hat{\mu}_{F,f}$, $\hat{\sigma}^2_{F,f}$ and ℓ-RAVE$_{F,.}$
end if

TECHNISCHE UNIVERSITÄT DARMSTADT

FUSE and FUSE^R

Output	Search tree (most visited path)	RAVE score
Algorithm	**FUSE** RAVE score guides FUSE exploration	**FUSE^R** FUSE helps build RAVE score, indicating feature relevance
FS approach	**Wrapper**	**Filter**

Experimental Validation

Data set	Samples	Features	Properties
Madelon	2,600	500	*XOR*-like
Arcene	200	10,000*	disjunction of overlapping sub concepts
Colon	62	2,000	„easy“

* only top 2000 are considered for FUSE and CFS, ranked after their ANOVA score

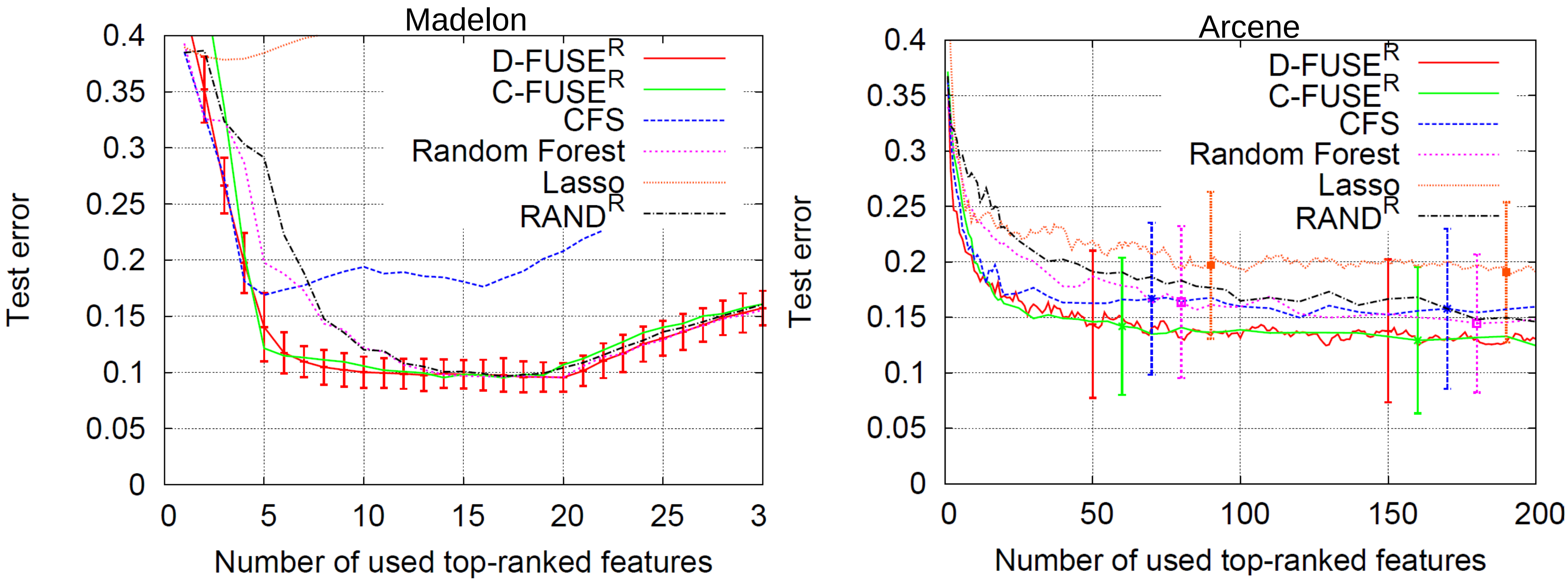
Baseline approaches

- Correlation-based Feature Selection (CFS)
- RandomForest-based Gini score (Gini-RF) *
- Lasso
- $RAND^R$ – average RAVE score built from random 20-feature subsets

* with 1,000 trees

- 200,000 iterations
- Gaussian SVM as end learner (5-fold CV optimized hyper-parameters)

Results



FUSE algorithms *"best of both worlds"*

- detect feature interdependencies (like Gini-RF, better with few features)
- filter out redundant features (like CFS, better with many features)

Results (contd.)

- all equal on colon
- **FUSE vs. FUSE^R:** FUSE does not control depth of search tree efficiently
  → FUSE^R better
- **discrete vs. continuous:** same performance with optimal parameters
  → discrete more robust due to less parameters

Performance on Madelon dataset

- FUSE^R converges more slowly than FUSE but improves after 10,000 iterations
- FUSE^R is faster by an order of magnitude than RAND^R
- runtime 45 minutes (Arcene: 5min, Colon: 4min) *

* on Intel Core 2×2.6GHz CPU with 2GB memory, only considering FS on the training set

Summary and Outlook

Contributions

- formalized FS task as a Reinforcement Learning problem
- proposed efficient approximation for optimal policy
- used UCT to define FUSE algorithm
- according to benchmark state of the art, but costly

Future directions

- extend to multi-class problems
- extend to mixed (continuous and discrete) search spaces
- combine FUSE with other end learners
- reconsider instant reward
- extend to feature construction

Critical Evaluation

- original approach for FS
- promising validation results

However…

- many degrees of freedom
  - interdependencies not fully understood
  - problem is simply shifted
- inherits problems from k-NN when working with
  - high dimensionality
  - skewed class distributions
- extensions probably further increase computational costs
- RF, Lasso as wrappers is fair for comparison,
  but unlike (usually) used in practice

Thank you!

Questions?

See next slide for sources

Sources

- Auer et. al.: *Finite-time Analysis of the Multiarmed Bandit Problem*. Machine Learning. 2002.
- **Gaudel, Romaric; Sebag, Michèle: *Feature Selection as a One-Player Game*.
In: Proceedings of the 27th International Conference on Machine Learning. 2010.**
- Gelly, Sylvain; Silver, David: *Combining Online and Offline Knowledge in UCT*.
In: Proceedings of the 24th International Conference on Machine Learning. 2007.
- Guyon, Isabelle; Elisseeff, André: *An Introduction to Feature Extraction*.
In: Guyon, Isabelle et. al. (editors): Feature Extraction. 2006.
- Guyon, Isabelle; Elisseeff, André: *An Introduction to Variable and Feature Selection*.
In: Journal of Machine Learning Research. 2003.
- Helmbold, David P.; Parker-Wood, Aleatha: *All-Moves-As-First Heuristics in Monte-Carlo Go*.
In: Proceedings of the 2009 International Conference on Artificial Intelligence. 2009.
- Kocsis, Levente et. al.: *Bandit based Monte-Carlo Planning*. In: Proceedings of the 17th European Conference on Machine Learning. 2006.
- **Sebag, Michele: *Monte Carlo Tree Search: From Playing Go to Feature Selection*.
Presentation. 2010.**

- http://theconnectivist-img.s3.amazonaws.com/wp-content/uploads/2013/10/Airplane-1300x724.jpg – *last accessed: 17.01.2015 10:00pm.*