Value Representation in Large Factored State Spaces

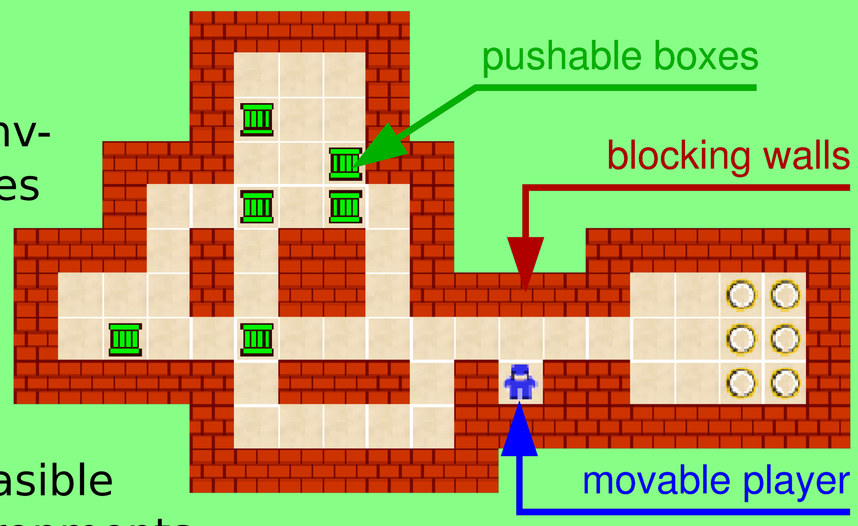
Wendelin Böhmer <wendelin@ni.tu-berlin.de>

Klaus Obermayer <oby@ni.tu-berlin.de>

Neural Information Processing Group, Technische Universität Berlin

Exploiting Factorization in Large Metric State Spaces

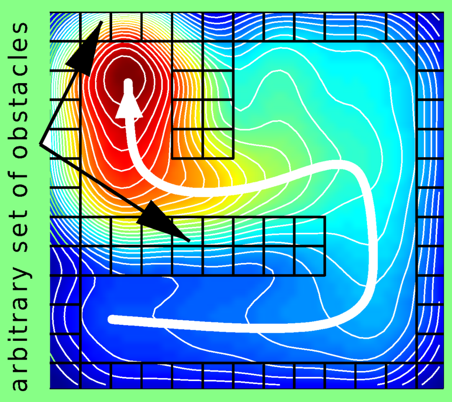
Overview:
- Autonomous agents in dynamic environments, e.g. with pushable boxes
- Stationary planning induces high dimensional state spaces ($\mathbb{O}(2^n)$)
- Example: puzzle game *Sokoban*



Problem A: sampling all states infeasible
Problem B: no transfer to new environments

Factored structure of induced state spaces:
- Often state variables describe "objects", e.g. detected from vision
- Reward functions often factorize: reward depends only on few variables
- *Factored MDP* can be exploited, but transition models rarely factorize

Divide & Conquer models:
- Relational desciption of transition rules
- Metric transition effect models
- Decomposable factorization
- Adapts to new situations



Representation with "Linear Factored Functions"

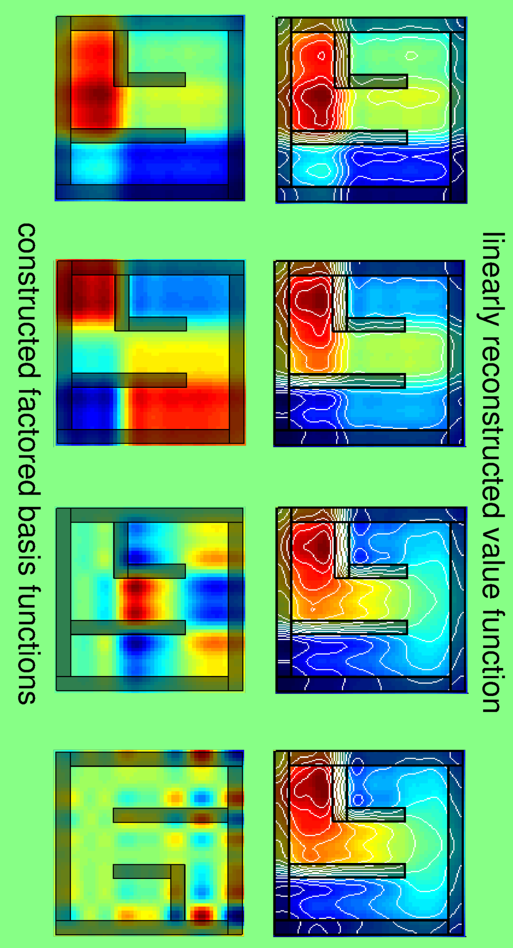
LFF: $f(\vec{x}) = \vec{a}^\top \vec{\varphi}(\vec{x}) = \vec{a}^\top \Big[\prod_\alpha \vec{\varphi}^\alpha(x_\alpha) \Big] = \vec{a}^\top \Big[\prod_\alpha \underbrace{\mathbf{B}^\alpha \vec{\phi}^\alpha(x_\alpha)}_{\vec{\varphi}^\alpha(x_\alpha)} \Big]$

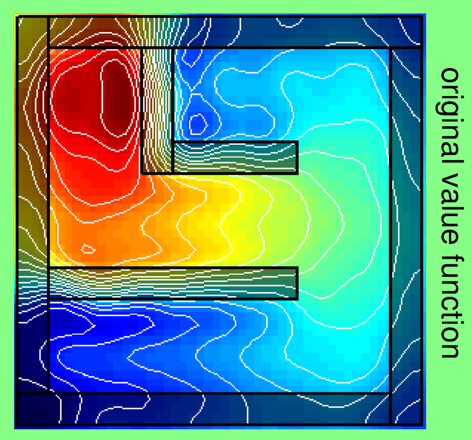
- *Inner products & marginalization* can be computed analytically as well as *point-wise products* (increases number of bases)
- *Partial derivatives* factorize in related LFF
- No analytical *nonlinearities* like *max* or *inverse*

Developed greedy LFF algorithms:
- *Compression*, e.g. after multiplication
- *Density estimation* from data $\{\vec{x}_t\}_{t=1}^n \sim \xi$
- *Regression* from labeles $\{y(\vec{x}_t)+\delta_t\}_{t=1}^n$
- *Sparse regression* that adjusts uncertanty prior

$$\begin{array}{rl} \inf\limits_{f,\vec{\mu},\eta} & \|y - f\|^2_{\xi\chi} + \frac{1}{\eta} D_{\mathrm{KL}}(\vec{\mu}\|\tfrac{1}{d}) \\ \text{s.t.} & \|\varphi_i^\alpha\|_{\nu^\alpha} = 1\,,\ \vec{x} \leftarrow \vec{x} + \vec{\epsilon}(\vec{x}) \\ \text{(priors)} & \epsilon_\alpha(\vec{x}) \sim \mathcal{N}\big(0,\ \mu_\alpha \tfrac{d\nu}{d\xi}(\vec{x})\big) \\ \text{(sparse)} & \mu_\alpha \geq 0\,,\ \sum\limits_\alpha \mu_\alpha = 1 \end{array}$$

Current research focus:
- Increase stability under multiplication with priors
- Learn factorizing transition operators
- Nonlinear policy improvement by symbolic reasoning





Q-Value Function
$Q^\pi(\vec{x},\vec{a}) = \sum_i a_i \psi_i(\vec{z}) = \sum_i a_i \prod_\alpha \psi_i^\alpha(z_\alpha)$

construct factored basis

relational policy iteration with metric evaluation

Factored Metric Models
$\hat{P}_k[\varphi](\vec{z}) = \prod_\alpha \hat{P}_k[\varphi^\alpha](\vec{z}_k, z_\alpha)$

Divide & Conquer Models
$\hat{P}[\varphi] = \varphi + \sum_k p_k \big(\hat{P}_k[\varphi] - \varphi\big)$

$p_k(\vec{z}) = \prod_\alpha p_k^\alpha(\vec{z}_k, z_\alpha)$
Grounded Relational Conditions

Value Function
$V^\pi(\vec{x}) = \sum_i w_i \varphi_i(\vec{x}) = \sum_i w_i \prod_\alpha \varphi_i^\alpha(x_\alpha)$

Metric Action Policies
$\hat{\Gamma}_k[\psi](\vec{x}) = \int \pi_k(d\vec{a})\ \psi(\vec{x},\vec{a}) = \psi^x(\vec{x}) \cdot \hat{\Gamma}_k[\psi^a](\vec{x})$

Divide & Conquer Policies
$\hat{\Gamma}_\pi[\psi] = \hat{\Gamma}_r[\psi] + \sum_k p_k \Big(\hat{\Gamma}_k[\psi] - \hat{\Gamma}_r[\psi]\Big)$

$p_k(\vec{x}) = \prod_\alpha p_k^\alpha(\vec{x}_k, x_\alpha)$
Grounded Relational Conditions

ground relational rules

provide metric policies for action symbols

Relational Rules
"collosion predicate" $c()$
- Sokoban Rule 1: $\forall(k \in B \cup W).\neg c(x,k,a)$
- Sokoban Rule 2: $\exists(b \in B).[c(x,b,a) \wedge \forall(k \in B \cup W\ b).\ \neg c(b,k,a)\ ]$

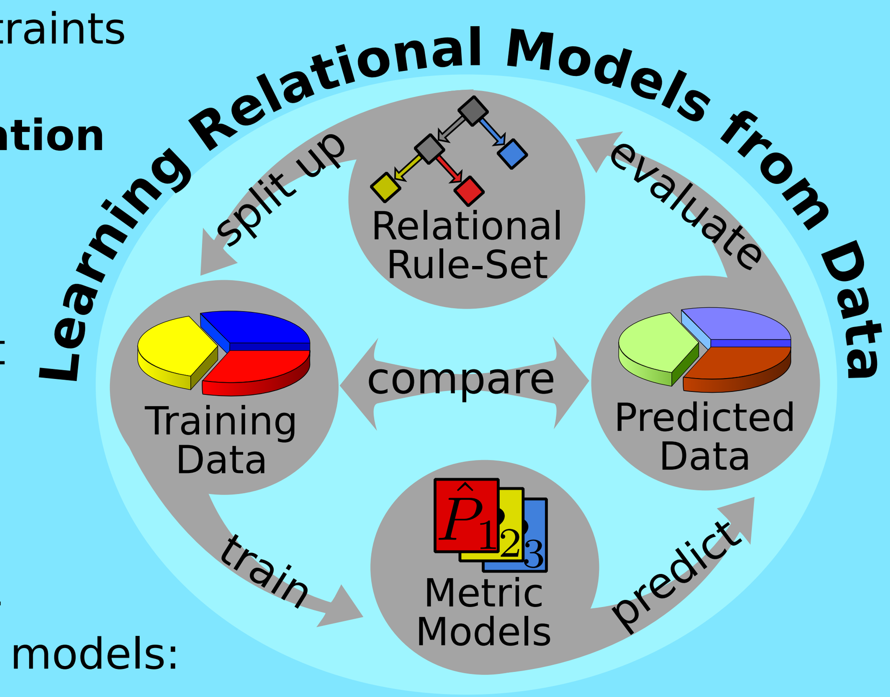
State and Action Symbols

Project Goal A

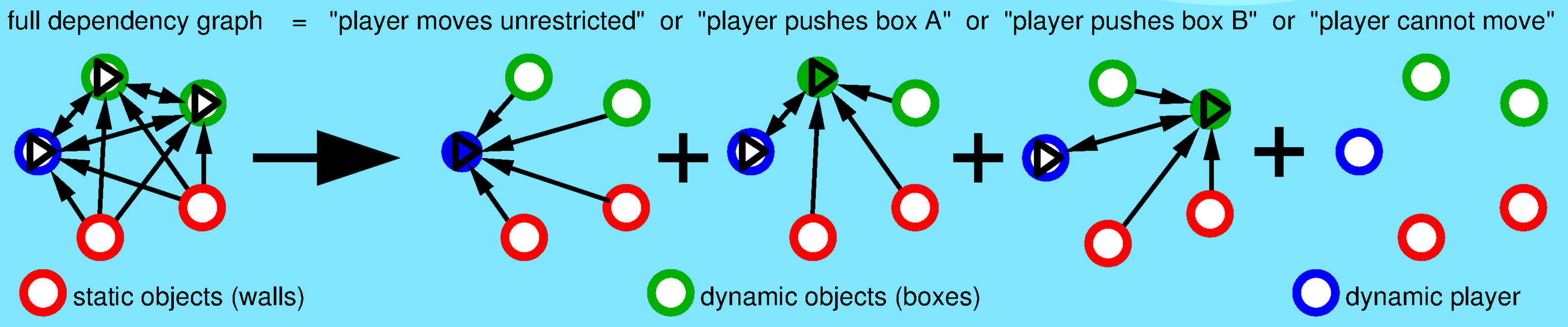
Bayesian Learning of Relational Models for Transitions and Reward from Data

Work Package A1: lean models
- Leaning relational models is greedy
- Bayesian distribution intractable in practice
- Sparse coding allows tractable approximations
- Learn rule-sets that split training data
- Include structural (factored) constraints

Work Package A2: active explotation
- Training data is finite but may be extended by active exploration
- Find the most informative states, i.e. where likely models contradict
- Explore those states actively, i.e. scientific hypothesis testing



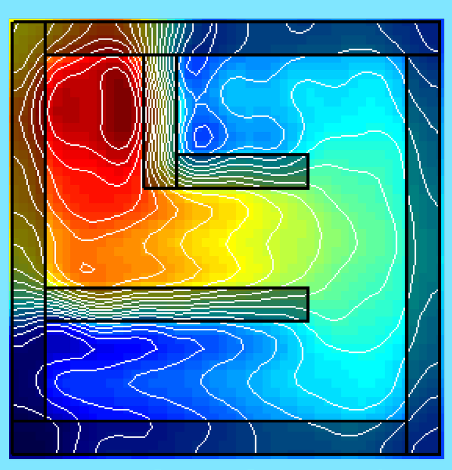
Exploit factored structure by decomposition in multiple metric transition models:



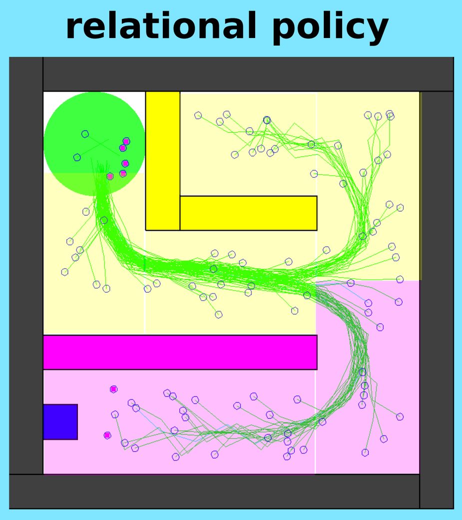
Project Goal B

Utilize Metric Information for Relational Policy Improvement, and to Adapt Symbols

Work Package B1: metric evaluation
- Merge relational planning and metric values
- Relational RL yields action sequences, no policies
- Evaluate candidate AS with their metric value



metric value and **relational policy**



Work Package B2: decision tree policies
- Grounded relational policies are *decision trees*
- Construct decision trees that maximize Q-value
- Greedy predicate selection vs. Bayesian methods

Work Package B3: adjust relational symbols
- Relational actions a_k have metric policies π_k
- Most actions a_k depend on vew variables
- Interprete Q-values as rewards for π_k, i.e. adjust π_k with standard RL in reduced space
- New actions: fequent local optimizations

Example relational policy on the left:
- Conditions select yellow and mangenta areas
- Action a_m: keep mangenta wall to your *left*
- Action a_y: keep yellow wall to your *right*

Linking Metric and Symbolic Levels in Autonomous Reinforcement Learning

DFG Schwerpunktprogramm 1527
Autonomous Learning