State Extraction by Dimensionality Reduction

Wendelin Böhmer[1], Steffen Grünewälder[2] and Klaus Obermayer[1]

[1]Neural Information Processing Group, Berlin Institute of Technology [2]Centre for Computational Statistics and Machine Learning, University College London

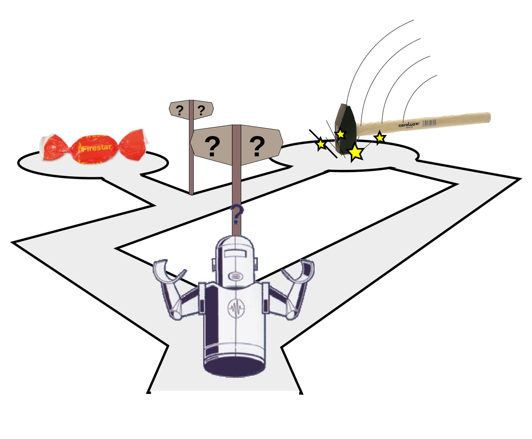
Abstract

Real world data (e.g. video images) are usually high dimensional and present information highly mixed and noise afflicted. Most applications therefore require a *dimensionality reduction* or *information filtering* beforehand. Such filters have to present the desired information in an applicable fashion and have to make sure few are lost. Several methods exist that *learn* these filters from the statistics of presented training data (e.g. PCA, CCA, PLS or SFA [1]).
We used the unsupervised method of *slow feature analysis* (SFA) [2] to extract the position of a robot from the images of its head-mounted camera. To catch higher order correlations in video images, we constructed a kernelized SFA algorithm analogous to *kernel PCA* [4], which outperformed its linear counterpart considerably. In order to deal with the huge number of training samples needed to catch the statistics of real images, we employed a sparse kernel matrix approximation method first introduced by Csató and Opper [5]. The resulting feature space (an approximation of the space of trigonometric polynomials [3]) is particularly well suited to approximate continuous functions with a linear model, e.g. *value functions* in *reinforcement learning*. We demonstrated this by learning the robots control in a simple navigation task.

Reinforcement Learning

Q-Learning approach:

- Execute *actions* a_t in *states* $x_t \in \mathcal{X}$
- Draw actions from *policy* $a_t \sim \pi(x_t)$
- Next state is drawn $x_{t+1} \sim P(x_t, a_t)$
- Receive *rewards* $r_t \sim R(x_t, a_t, x_{t+1})$
- Estimate the expected discounted future reward given a choice a in x:



$$Q^\pi(x,a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t\, r_t \big| x_0 = x, a_0 = a\Big] \quad \text{(Q-value)}$$

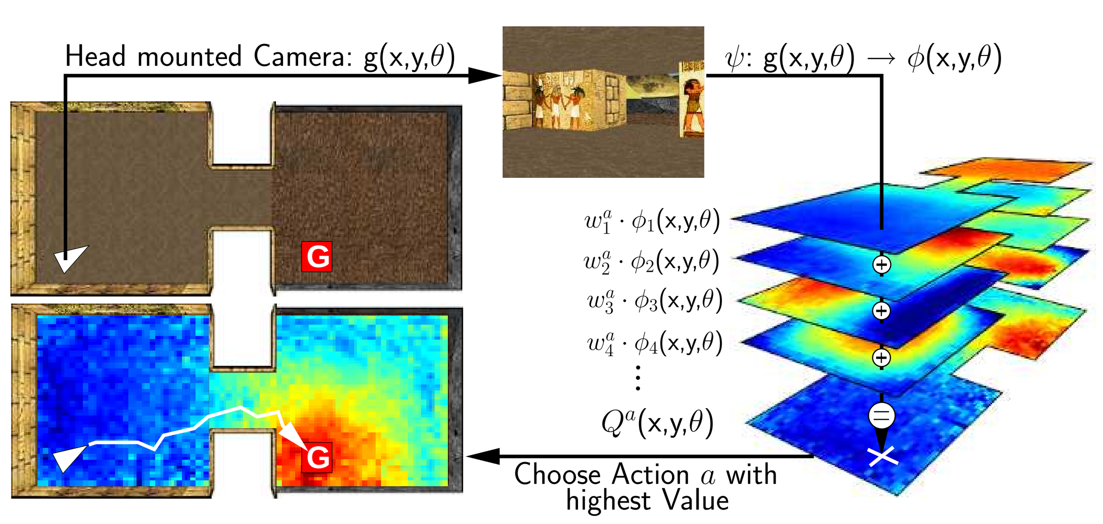
- Find the *optimal policy* $\pi^*(\cdot)$ that $\forall x \in \mathcal{X} : \max Q^{\pi^*}(x, \pi^*(x))$
- Define $\pi^*(x)$ by always choosing $a^* = \arg\max_a Q^{\pi^*}(x,a)$
- *Policy iteration* between Q-value estimation and policy improvement
- $\mathcal{X}$ is *continuous* $\Rightarrow Q^\pi(x,a)$ must be approximated, e.g. linear: $Q^\pi(x,a) \approx Q^a(x) = \boldsymbol{\varphi}(x)^\top \boldsymbol{w}^a$, with $\boldsymbol{\varphi} : \mathcal{X} \mapsto \Phi$

$\Rightarrow$ **Feature space Φ must facilitate function approximation!**

Robot Navigation

- Hidden state $(x, y, \theta) \in \mathcal{X}$ (position x, y and orientation θ)
- Observation $z = g(x, y, \theta) \in \mathcal{Z}$ (images from head mounted camera)

$\Rightarrow$ **Requires a mapping $\boldsymbol{\phi} : \mathcal{Z} \mapsto \Phi$, with $\boldsymbol{\phi}(g(x,y,\theta)) \approx \boldsymbol{\varphi}(x,y,\theta)$**



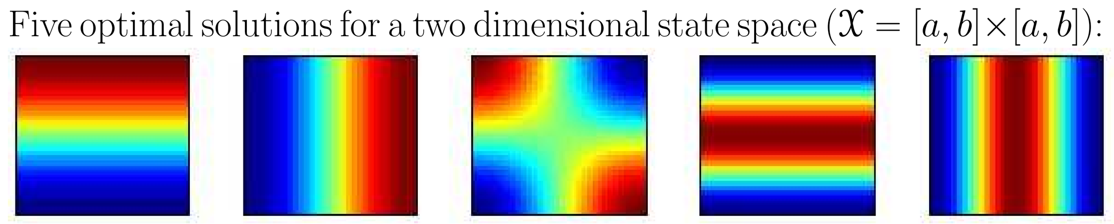
Slow Feature Analysis

- SFA [2] analyzes second order statistics of *time series* $z^{(1)}, \ldots, z^{(n)} \in \mathcal{Z}$
- Learns a set of mappings $\boldsymbol{\phi}_i : \mathcal{Z} \mapsto \mathbb{R}$ that change slowly over time
- Infinite time series and unrestricted model class $\Rightarrow$ Mappings become *trigonometric polynomials* in the domain of a *hidden state* $x^{(t)} \in \mathcal{X}$
- $x^{(t)}$ is the *minimal explanation* of $z^{(t)}$, in the sense that there exists an unknown invertible function $g : \mathcal{X} \mapsto \mathcal{Z}$ such that $\forall t : z^{(t)} = g(x^{(t)})$

Optimization Problem

$$\begin{aligned}
\min \quad s(\boldsymbol{\phi}_i) &:= \mathbb{E}[\dot{\boldsymbol{\phi}}_i^2] && \text{(Slowness)}\\
\text{s.t.} \quad \mathbb{E}[\boldsymbol{\phi}_i] &= 0 && \text{(Zero Mean)}\\
\mathbb{E}[\boldsymbol{\phi}_i^2] &= 1 && \text{(Unit Variance)}\\
\forall j \neq i : \mathbb{E}[\boldsymbol{\phi}_i \boldsymbol{\phi}_j] &= 0 && \text{(Decorrelation),}\\
\forall j > i : s(\boldsymbol{\phi}_i) &\leq s(\boldsymbol{\phi}_j) && \text{(Order)}
\end{aligned}$$

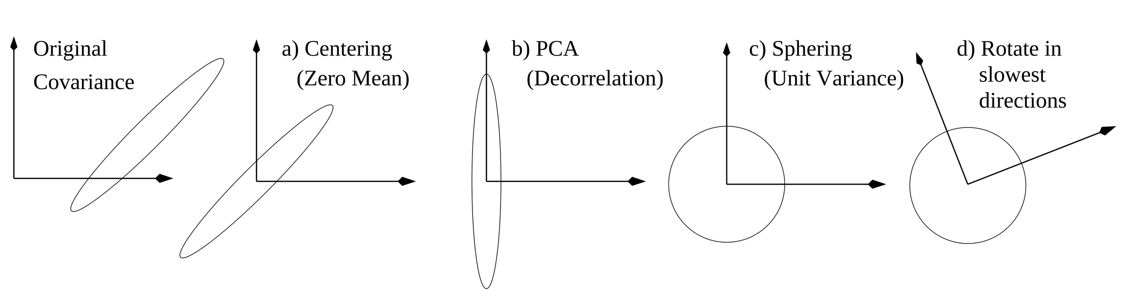
Five optimal solutions for a two dimensional state space ($\mathcal{X} = [a,b] \times [a,b]$):



Features of equal slowness can also exhibit an arbitrary rotation in their subspace, i.e. $s(\boldsymbol{\phi}_i) = s(\boldsymbol{\phi}_j) = s(a\boldsymbol{\phi}_i + b\boldsymbol{\phi}_j)$ with $a^2 + b^2 = 1$.

Linear SFA

- Given $\boldsymbol{z}^{(t)} \in \mathbb{R}^d$, assume $\boldsymbol{\phi}_i(\boldsymbol{z}) := \boldsymbol{w}_i^\top \boldsymbol{z} - c_i$ to be an affine function
- Fulfill constrains (a-c); requires eigenvalue decomposition of $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top]$
- Minimize objective (d) by rotation to lowest eigenvectors of $\mathbb{E}[\dot{\boldsymbol{z}}\dot{\boldsymbol{z}}^\top]$
- Overall time complexity $\mathcal{O}(d^2 n)$



Kernel SFA

- Implicit projection $\boldsymbol{\psi}(\cdot)$ in high dimensional feature space
- *Positive semidefinite kernel function* $k(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\psi}(\boldsymbol{z}')$
- Steps analogous to linear SFA (for projected data $\boldsymbol{\Psi}_{it} = \boldsymbol{\psi}_i(\boldsymbol{z}^{(t)})$):

a) Centered kernel matrix $\mathbf{K}_c = \boldsymbol{\Psi}_c^\top \boldsymbol{\Psi}_c = (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{\Psi}^\top\boldsymbol{\Psi}(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)$

b) Analogous to kernel PCA [4]: Let $\mathrm{cov}(\psi(x)) = \frac{1}{n}\boldsymbol{\Psi}_c\boldsymbol{\Psi}_c^\top := \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ and $\mathbf{K}_c = \boldsymbol{\Psi}_c^\top\boldsymbol{\Psi}_c := \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$, then $\mathbf{U} = \frac{1}{\sqrt{n}}\boldsymbol{\Psi}_c\mathbf{V}\boldsymbol{\Lambda}^{-1/2}$ [1]

c) Sphered expanded data $\boldsymbol{\Psi}_s = \frac{1}{\sqrt{n}}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top\mathbf{K}_c$

d) Rotation to eigenvectors $\mathbf{R}$ of $\dot{\boldsymbol{\Psi}}_s\dot{\boldsymbol{\Psi}}_s^\top = \frac{1}{n}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top\dot{\mathbf{K}}_c\dot{\mathbf{K}}_c\mathbf{V}\boldsymbol{\Lambda}^{-1}$

$$\Rightarrow \quad \boldsymbol{\phi}(\boldsymbol{z}) = \mathbf{W}^\top \boldsymbol{k}(\boldsymbol{z}) - \boldsymbol{c}$$

$$\text{where } \mathbf{W} = \tfrac{1}{\sqrt{n}}(\mathbf{I} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{R} \quad \text{and} \quad \boldsymbol{c} = \tfrac{1}{n}\mathbf{W}^\top\mathbf{K}\mathbf{1}$$

- Raises time complexity to $\mathcal{O}(n^3)$

Kernel Matrix Sparsification

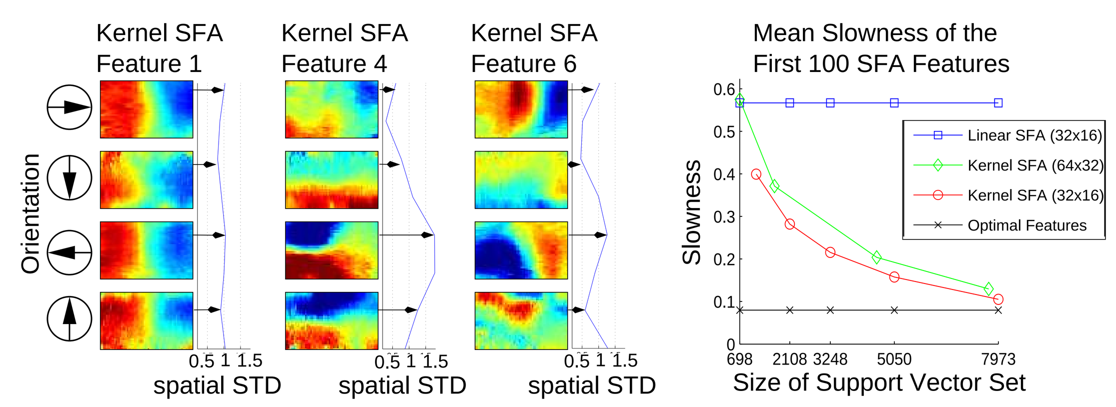
- $\boldsymbol{\phi}(\boldsymbol{z})$ depends on *all* training samples $\boldsymbol{k}(\boldsymbol{z}) = [k(\boldsymbol{z}^{(1)}, \boldsymbol{z}), \ldots, k(\boldsymbol{z}^{(n)}, \boldsymbol{z})]^\top$
- Express $\boldsymbol{\phi}(\cdot)$ with subset $\{\boldsymbol{s}^{(i)}\}_{i=1}^m \subset \{\boldsymbol{z}^{(i)}\}_{i=1}^n$
- Select only samples that contribute to $\mathcal{A} := \mathbf{aff}\{\boldsymbol{\psi}(\boldsymbol{s}^{(i)})\}_{i=1}^m$ largely [5]
- After one pass, all $\boldsymbol{\psi}(\boldsymbol{z}^{(i)})$ lie within a *sparsity threshold* ϵ of $\mathcal{A}$, i.e.

$$\min_\alpha \|\boldsymbol{\psi}(\boldsymbol{z}) - \sum_{i=1}^m \alpha_i \boldsymbol{\psi}(\boldsymbol{s}^{(i)})\|_2^2 = k(\boldsymbol{z},\boldsymbol{z}) - \boldsymbol{k}(\boldsymbol{z})^\top\mathbf{K}^{-1}\boldsymbol{k}(\boldsymbol{z}) \leq \epsilon^2$$

- The algorithm keeps track of $\mathbf{K}^{-1}$ and has time complexity $\mathcal{O}(m^2 n)$

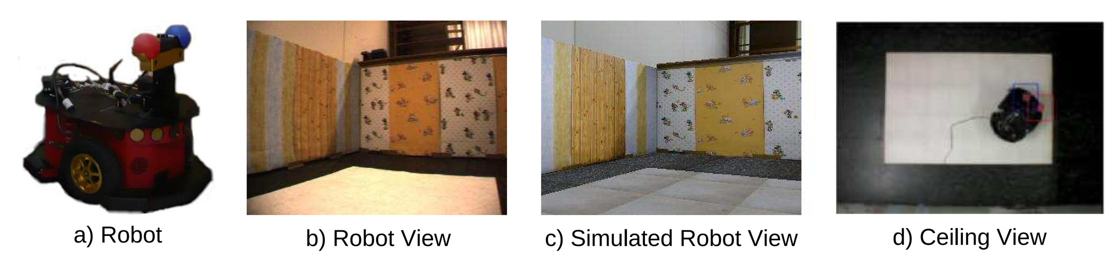
Sparse Kernel SFA

- Using m *support vectors*, time complexity reduces to $\mathcal{O}(m^2 n)$
- Vary m to find a tradeoff between complexity and slowness
- Experiments with ego perspective images as inputs (see next section):



Left: Spatial sparse kernel SFA feature output with 8,000 out of 35,000 support vectors

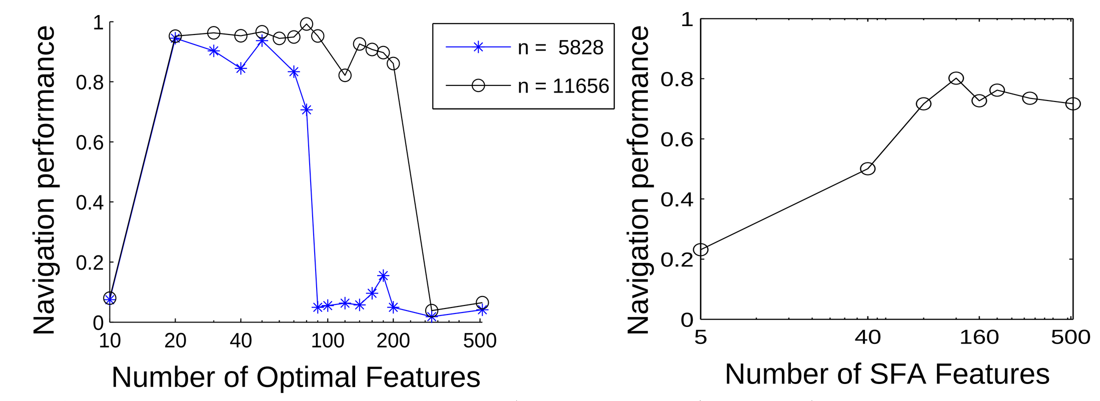
Robot Experiments



- Robots environment is a small rectangular room (wallpapered walls)
- Ego perspective images (head mounted camera) as states
- Choose between 3 actions: *move forward* and *turn left/right* 45°
- *Goal area* is not marked, but receives positive reward
- Positions close to walls are punished
- *Least Squares Policy Iteration* (LSPI) algorithm to learn policy π
- Estimate *navigation performance* of policy π: $\mathbb{E}\left[\frac{Q^\pi(x,\pi(x))}{Q^{\pi^*}(x,\pi^*(x))}\right]$

Simulated Experiment

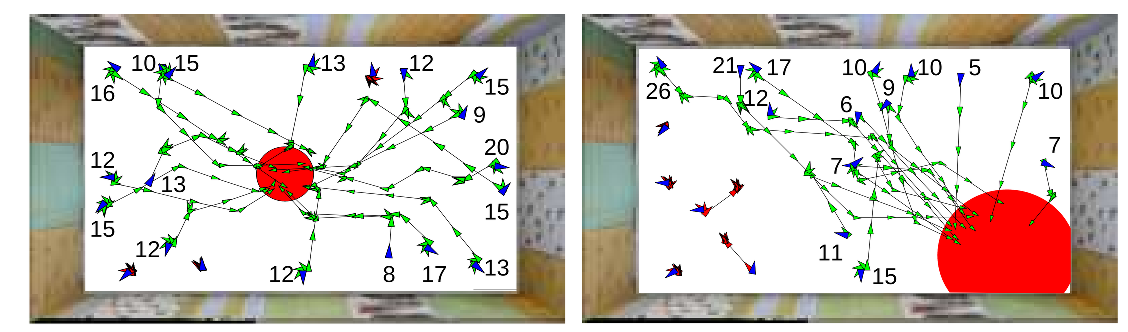
- *Control experiment* with optimal features (left figure)
- 3D-model of real world experiment (photographed textures)
- Learn sparse kernel SFA features with 35,000 rendered images
- Learn policy for centered goal area (ca. 12,000 seen transitions)
- Policy needs more features to reach working regime (right figure)
- Navigation performance is lower (ca. 2-3 unnecessary steps)



Left: Control experiment on optimal features (1000 trajectories/estimation). n ≙ # training samples.
Right: Simulated experiment with sparse kernel SFA features (8k/35k) (200 trajectories/estimation).

Real-World Experiment

- Same trajectories as in the simulated experiment (real images)
- Too time expensive to estimate navigation performance
- Two experiments with different reward (red goal areas):



Numbers indicate needed actions. 128 sparse kernel SFA features (8k/35k) are used for approximation.

- 75% success rate, but much more unnecessary steps
- Oscillatory behavior in areas of similar Q-value ($\Rightarrow$ strong noise)

$\Rightarrow$ **Learned feature space works but not yet noise robust**

Outlook

- **Approximation in noise afflicted feature spaces**
  - External influences and a weak model class appear as *noise*
  - ▹ Can we give *bounds* and *convergence statements*?
  - SFA features do not take past observations into account
  - ▹ Can we construct a *recurrent filter* to reduce noise?
- **Computational efficiency and alternative feature spaces**
  - Kernel expansion is computational very expensive
  - Trigonometric polynomials need support on the complete domain $\mathcal{X}$
  - ▹ Can we learn a *sparse feature space* (e.g. Gaussian bells)?
  - Hidden state might be high dimensional (e.g. moving humans)
  - ▹ Can we filter out state dimensions that are *uncorrelated* to the target?

References

[1] Shawe-Taylor, J.; Cristianini, N.: *Kernel Methods for Pattern Analysis*, Cambridge University Press (2004)

[2] Wiskott, L.: *Slow Feature Analysis: A Theoretical Analysis of Optimal Free Responses*, Neural Computation 15(9): 2147-2177 (2003)

[3] Franzius, M.; Sprekeler, H.; Wiskott, L: *Slowness and sparseness leads to place, head-direction and spartial-view cells*, PLoS Computational Biology (2007)

[4] Schölkopf, B.; Smola, A. J.; Müller, K.-R.: *Kernel principal component analysis*, Artificial Neural Networks ICANN (1997)

[5] Csató, L.; Opper, M.: *Sparse Online Gaussian Processes*, Neural Computation 14(3): 641-668 (2002)