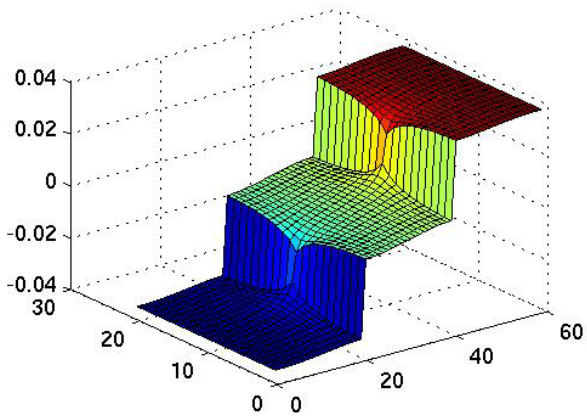
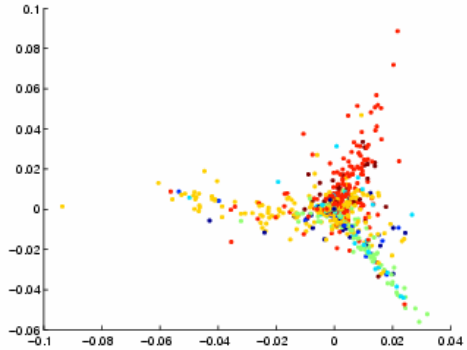




New Frontiers in Representation Discovery

Sridhar Mahadevan

Department of Computer Science

University of Massachusetts, Amherst

Collaborators: Mauro Maggioni (Duke), Jeff Johns, Sarah Osentoski, Chang Wang, Kimberly Ferguson

National Science Foundation
WHERE DISCOVERIES BEGIN

Structure of Tutorial

PART 1	Motivation: Why automate representation discovery?
PART II	Representation Discovery using Fourier Manifold Learning
	COFFEE BREAK
PART III	*Multiscale* Representation Discovery using Wavelet Manifold Learning
PART IV	Advanced Topics and Challenges; Discussion

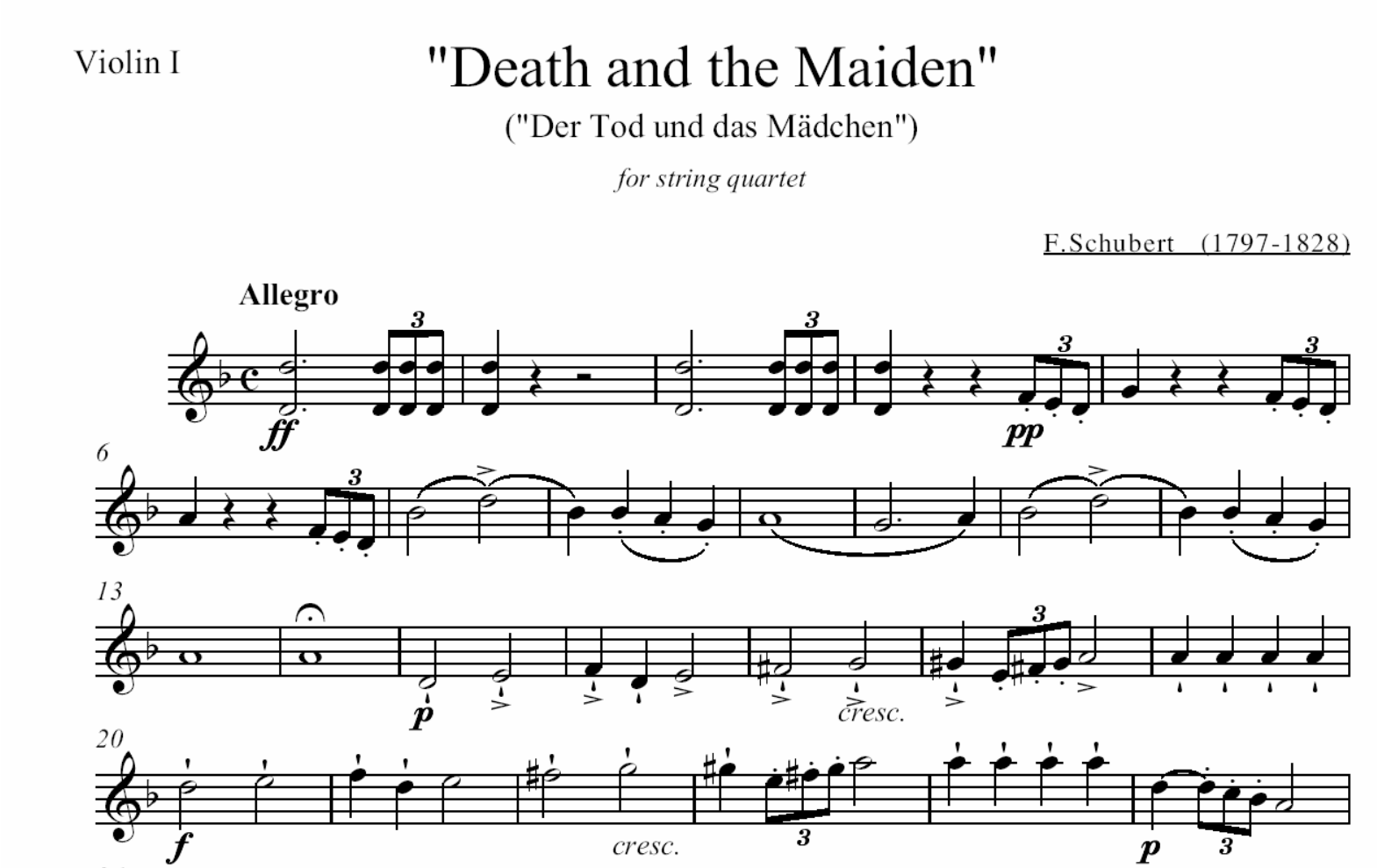
Two Approaches to Representation Discovery

- **Diagonalization**:
  - First discovered by Joseph Fourier in 1807
  - Time/Space $\rightarrow$ Frequency
  - Basis functions are localized in frequency alone
  - Matrix “Eigenvector” methods
- **Dilation**:
  - Multiscale framework that decomposes time/space simultaneously
  - Basis functions localized in space and time
  - Principle underlying “wavelets” (Daubechies, 1991)

What is a representation?
(Marr and Nishihara, 1978)

- "A *representation* is a formal system for making explicit certain entities or types of information, together with a specification of how the system does this."
- "For example, a representation for shape would be a formal scheme for describing some aspects of shape, together with rules that specify how the scheme is applied to any particular shape.
- "A musical score provides a way of representing a symphony."

Representation of Music



Representations of Numbers

- Roman:
  - I, II, III, IV, …
  - Can you multiply MCXII by LMIV?
  - Fashionable on wristwatches, but not very practical
- Decimal: (invented in India!)
  - 0, 1, 2, 3, 4, …
  - Positional system, widely used in science & engineering
- Binary:
  - 0, 1, 10, 11, 100,…
  - Made it possible to implement computers on hardware
- Many others: octal, hexadecimal,…

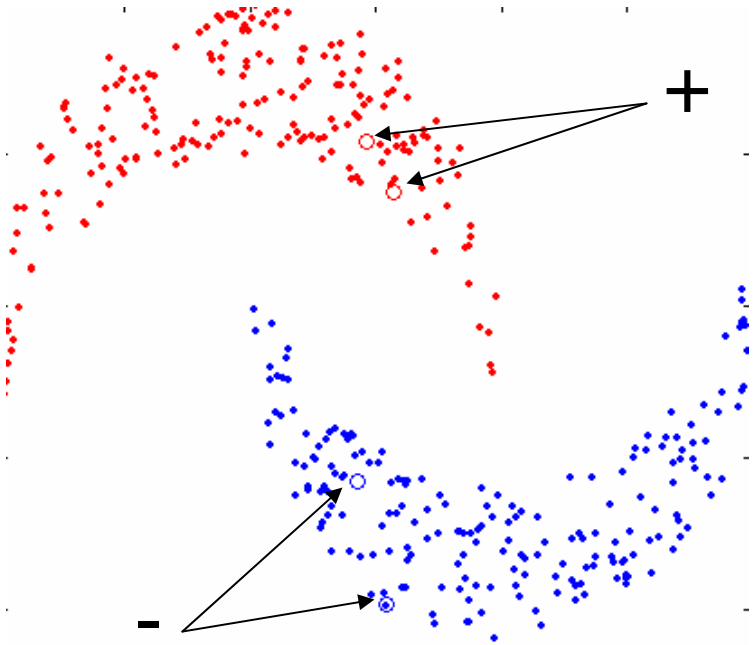
Popular Representations in AI

- Algebraic structures:
  - Matrices and vector spaces
- Relational structures:
  - Graphs
- Factored Representations
  - Graphical models
- Probabilistic finite-state representations:
  - Markov chains and Markov decision processes
- Propositional and predicate logic
  - Combining probability and logic

Why Automate Representation Discovery?

- Existing methods may be inadequate
  - Standard “Euclidean” methods fail
  - Data lies in an abstract space (e.g. a graph)
- Faster algorithms can be designed
  - By learning customized representations that exploit “smoothness” properties of functions on graphs
- Adaptive compression algorithms
  - 3D graphics and computer animation
- The environment may be non-stationary
  - Human-engineered representations cannot anticipate tasks an agent may face

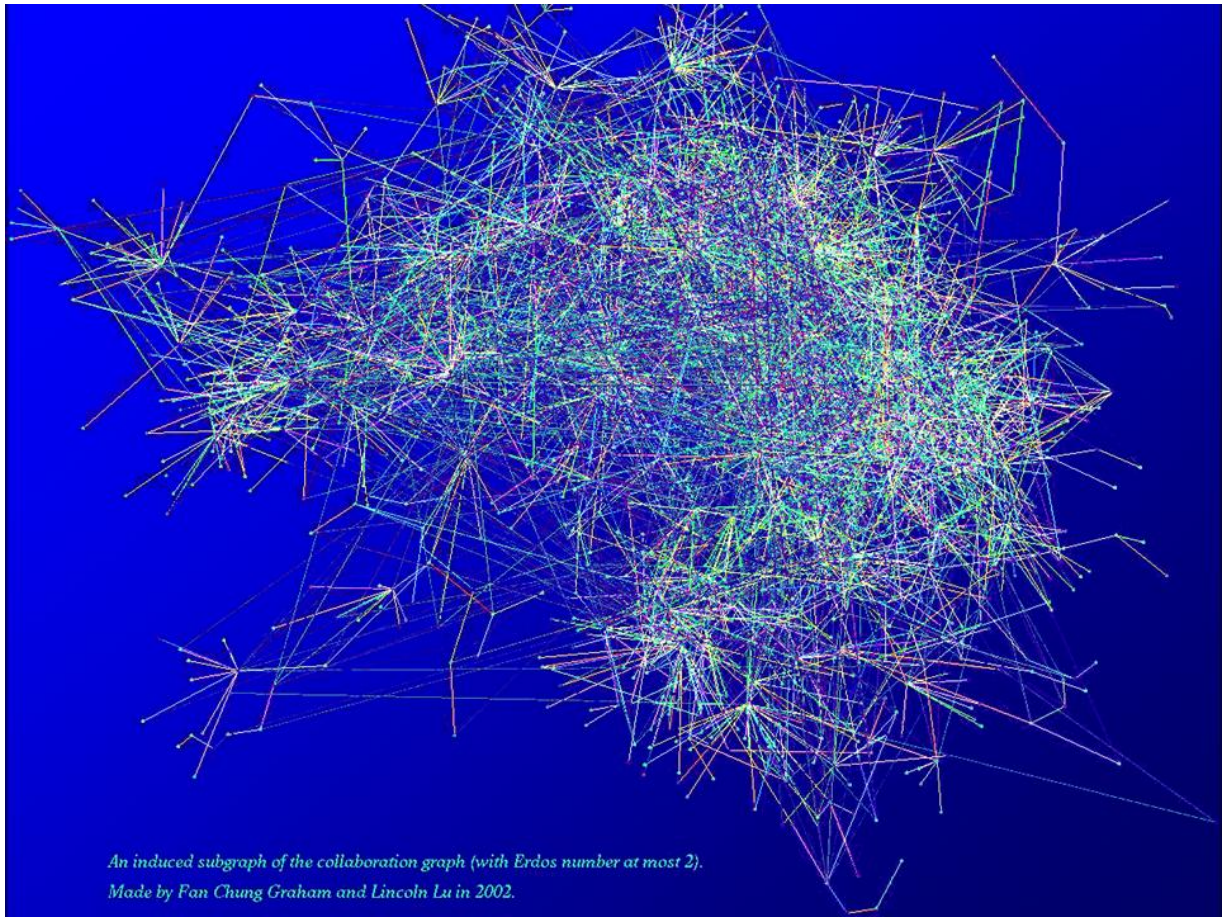
Semi-Supervised Learning



Euclidean methods
like “k-means”
or “nearest-neighbor”
fail on this task

“Two-moons
problem”

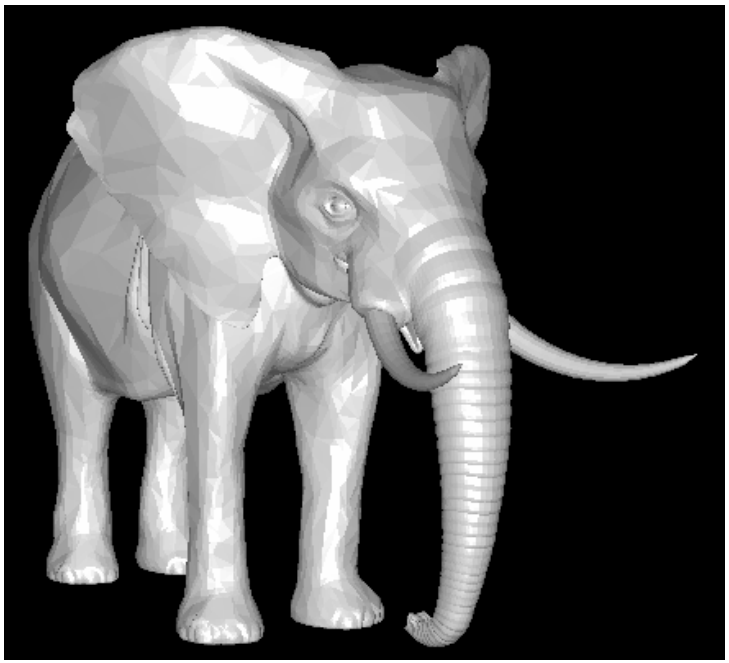
Social Network Analysis



Discovering hidden structure by embedding graphs in $\mathcal{R}^n$

Collaboration graph of authors with Erdos # =2

Compression of 3D Objects in Computer Graphics



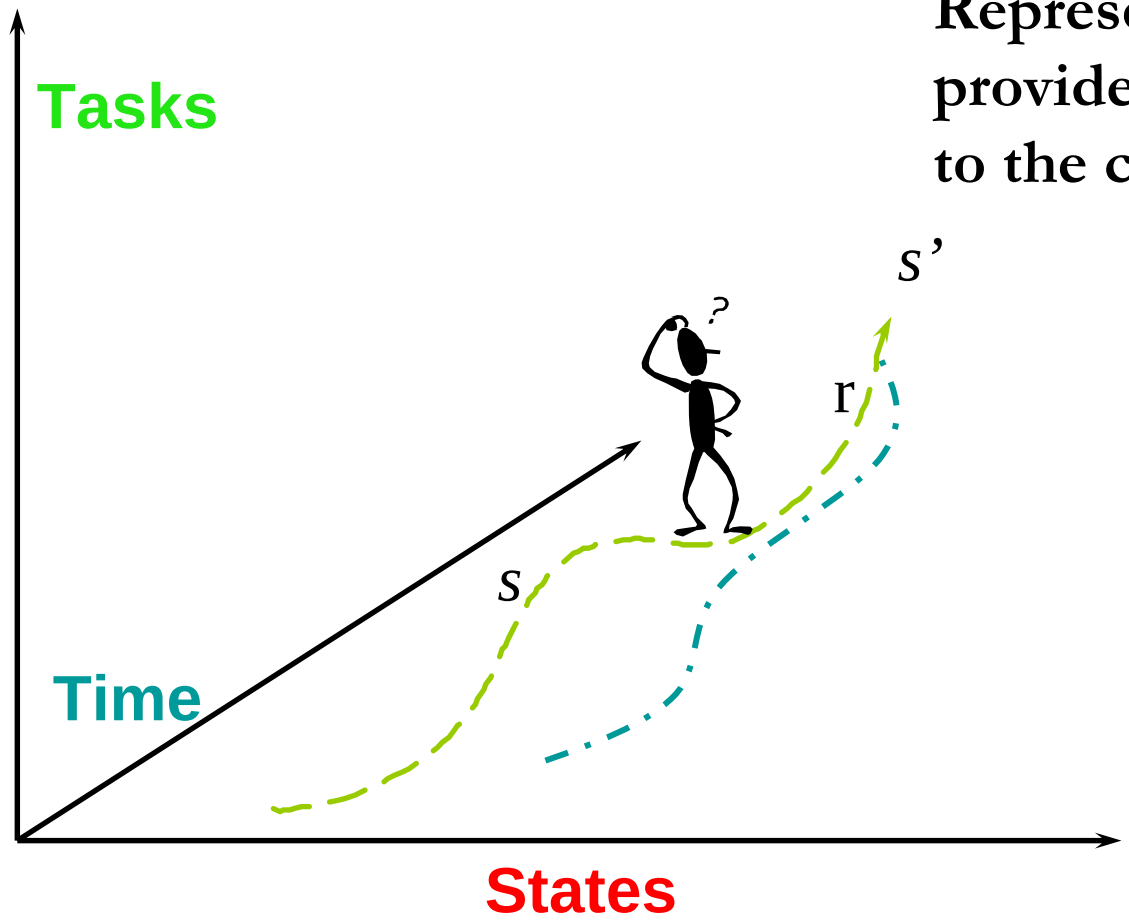
Spatial 3D Representation
1.5 Mbytes, 20,000 vertices

JPEG does not work for 3D objects.

Challenge: design an *adaptive* object-specific compression method

Credit Assignment Problem

(Minsky, Steps Toward AI, 1960)

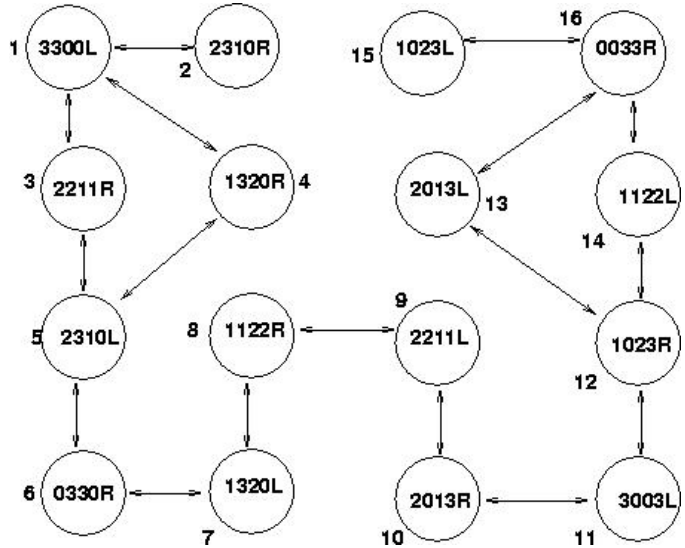


Representation Discovery provides a unified approach to the credit assignment problem

Representation Discovery in Problem-Solving

(Saul Amarel, 1960s)

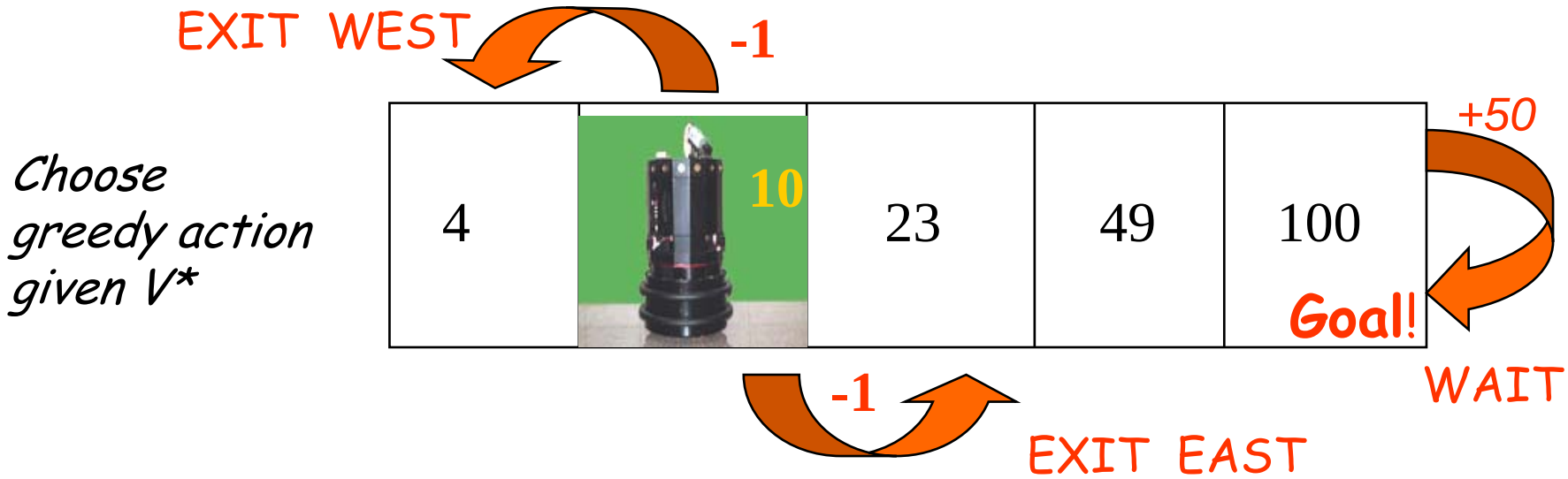
Missionaries and Cannibal



Find symmetries and bottlenecks in state spaces

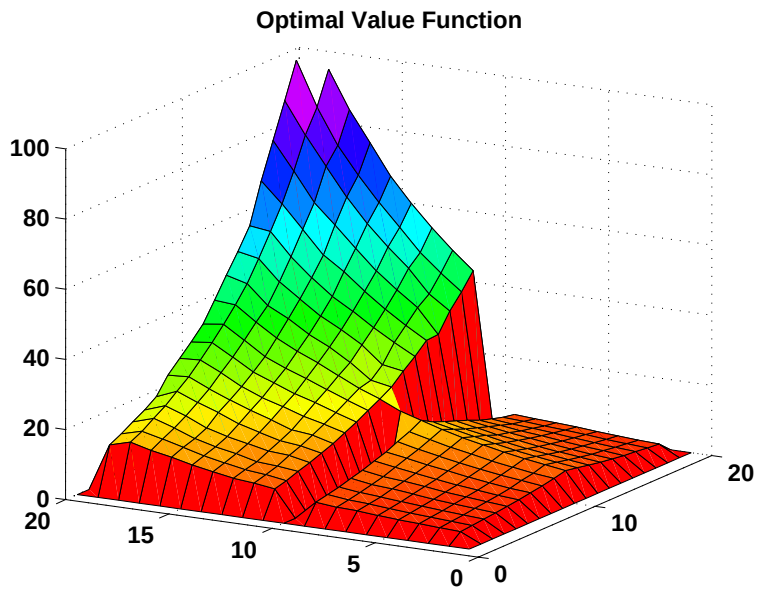
Markov Decision Processes
(Bellman, Howard)

$$V^*(x) = \max_{a \in A(x)} \left(r(x,a) + \gamma \sum_y P^a_{xy} V^*(y) \right)$$

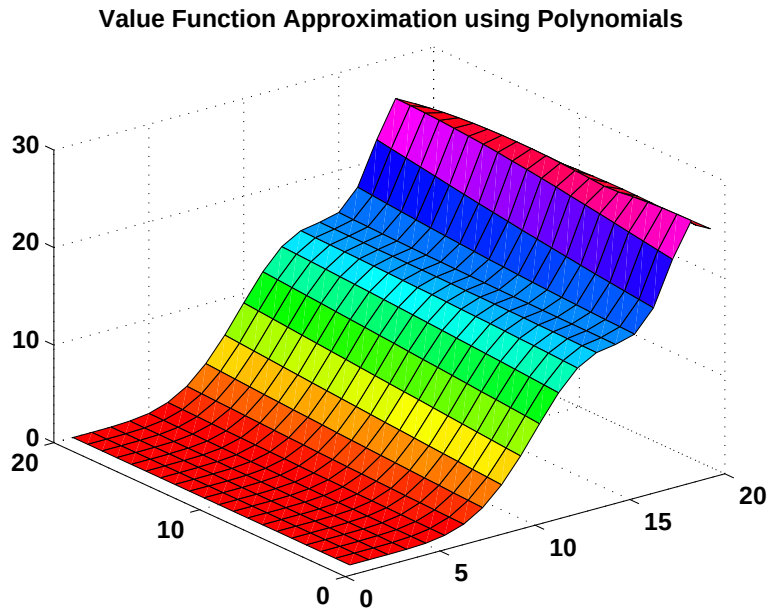


Representation Discovery in MDPs

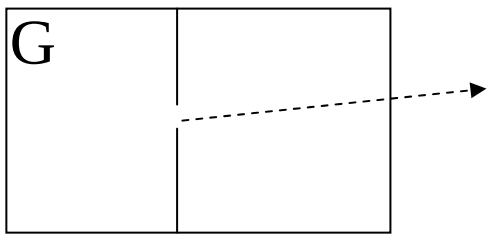
(Mahadevan, AAAI 2005)



OPTIMAL



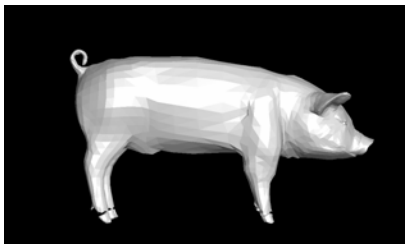
Human-engineered basis does poorly here



Bottleneck

Representations: The Hidden Dimension

Domains



Learning a Basis

Feature discovery

Fourier

Wavelet

Problems/Techniques

Clustering

Classification

Reinforcement Learning

Regression

Structure of Tutorial

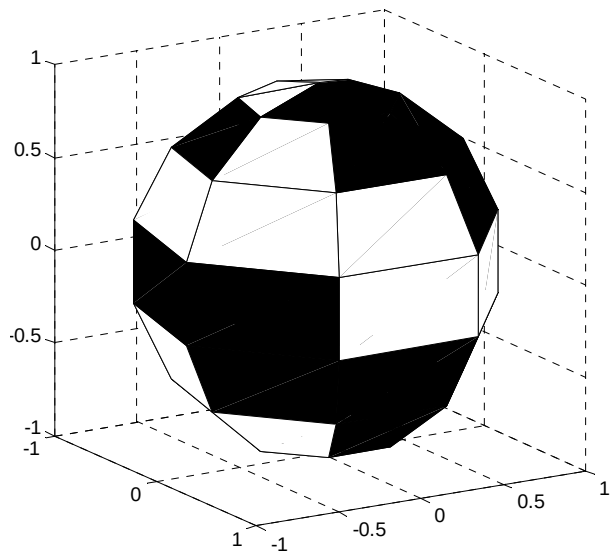
PART 1	Motivation: Why automate representation discovery?
PART II	Representation Discovery using Fourier Manifold Learning
	COFFEE BREAK
PART III	*Multiscale* Representation Discovery using Wavelet Manifold Learning
PART IV	Advanced Topics and Challenges; Discussion

Fourier Representation of Boolean Functions

$f = x_1 \neg x_3 \vee \neg x_2 x_3$

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Localized in space



Hadamard transform

Fourier Representation

4
0
2
-2
-2
-2
0
0

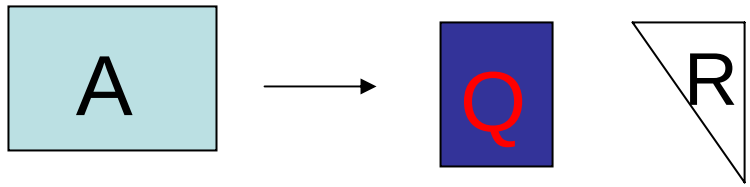
Localized in frequency

Fourier Manifold Methods

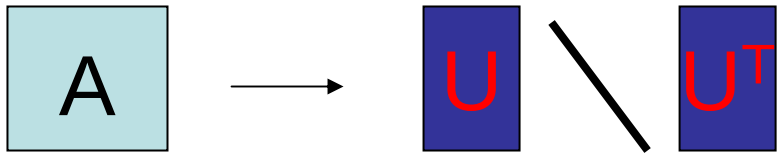
- Same basis construction principle as PCA
  - Diagonalization (or eigenvector construction)
  - The matrix representation changes from Σ to $\mathcal{L}$
- Spectral methods are based on computing eigenvectors of a normalized "affinity" matrix
  - [Shi and Malik, IEEE PAMI 1997; Ng, Jordan, and Weiss, NIPS 2001; Page, Brin, Motwani, Winograd, 1998]
- Manifold methods model the local geometry of the data by constructing a graph
  - [Roweis and Saul; Tenenbaum, de Silva, Langford, Science 2000; Belkin and Niyogi, MLJ 2004]

Some Matrix Decompositions

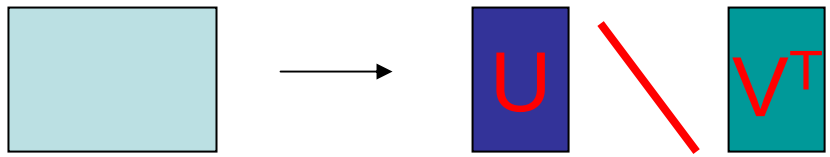
- **Gram-Schmidt:**
  - **A = QR**



- **Spectral:**
  - **A = U Λ U^T**



- **SVD:**
  - **A = U Σ V^T**



All of these represent a change of basis

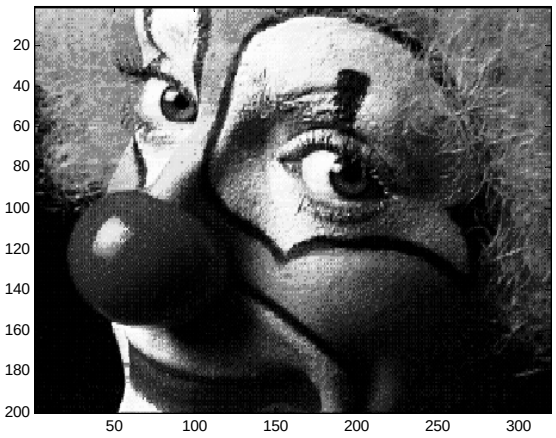
Singular Value Decomposition

$$A = U \Sigma V^T$$

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T) = V \Sigma^2 V^T$$

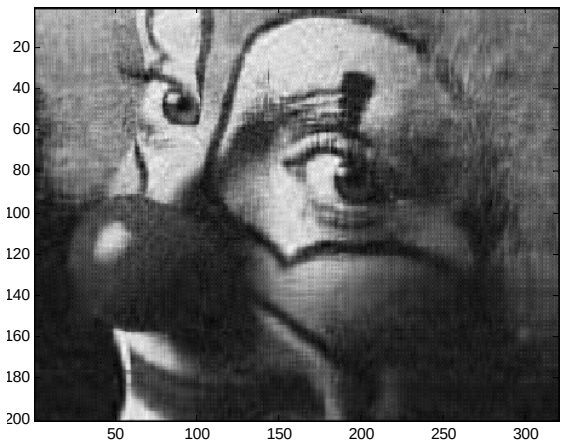
$$AA^T = (U \Sigma V) (U \Sigma V^T)^T = U \Sigma^2 U^T$$

Original: 200x320



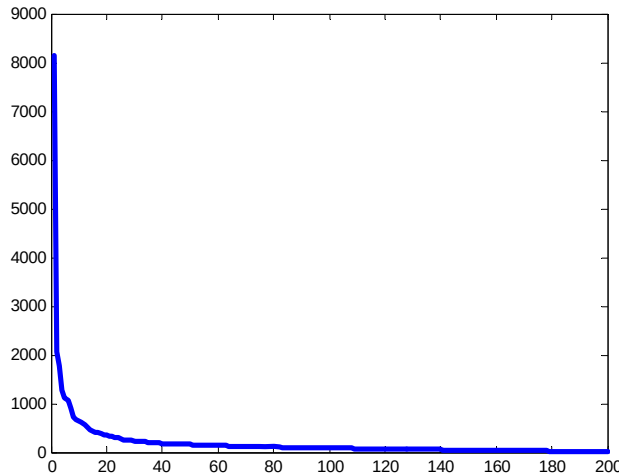
“Spatial Coordinates”

Low-rank approximation using 30 basis functions



“Frequency Coordinates”

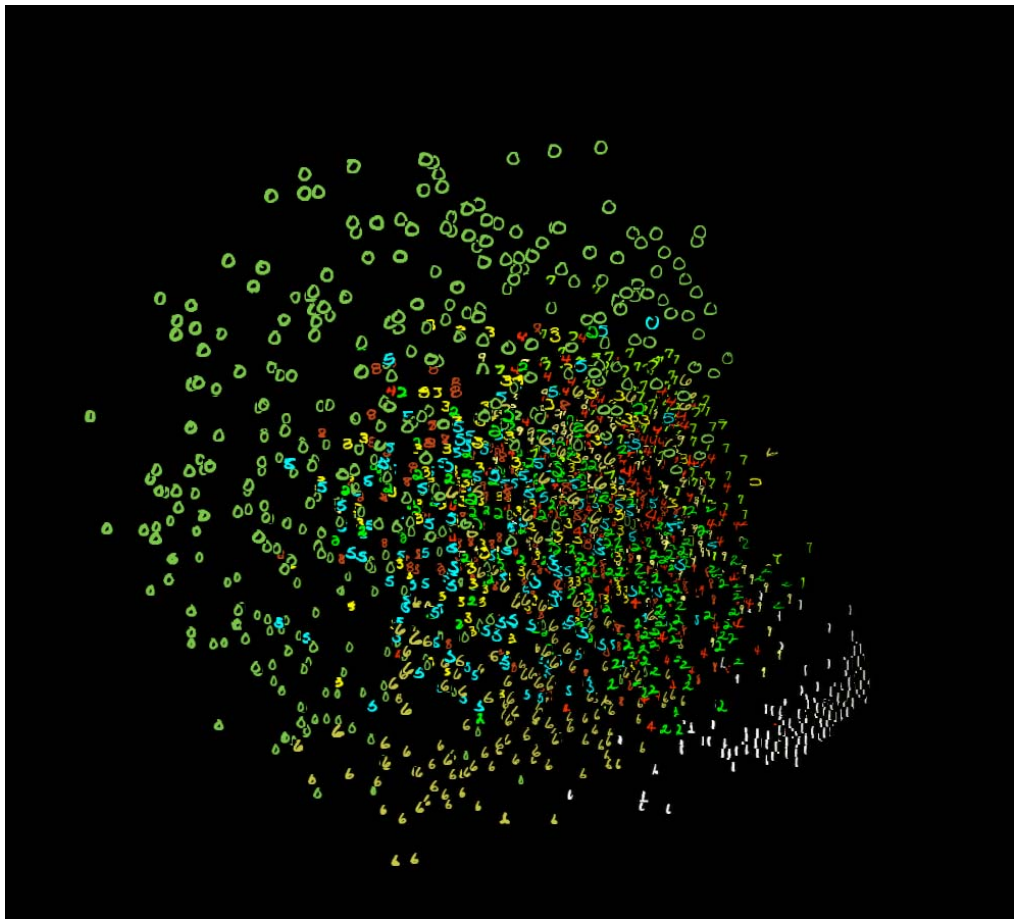
Image “energy”



$$A \sim \sum_i \sigma_i u_i v_i^T$$

Principal Components Analysis

Digit recognition task



- Form Covariance matrix

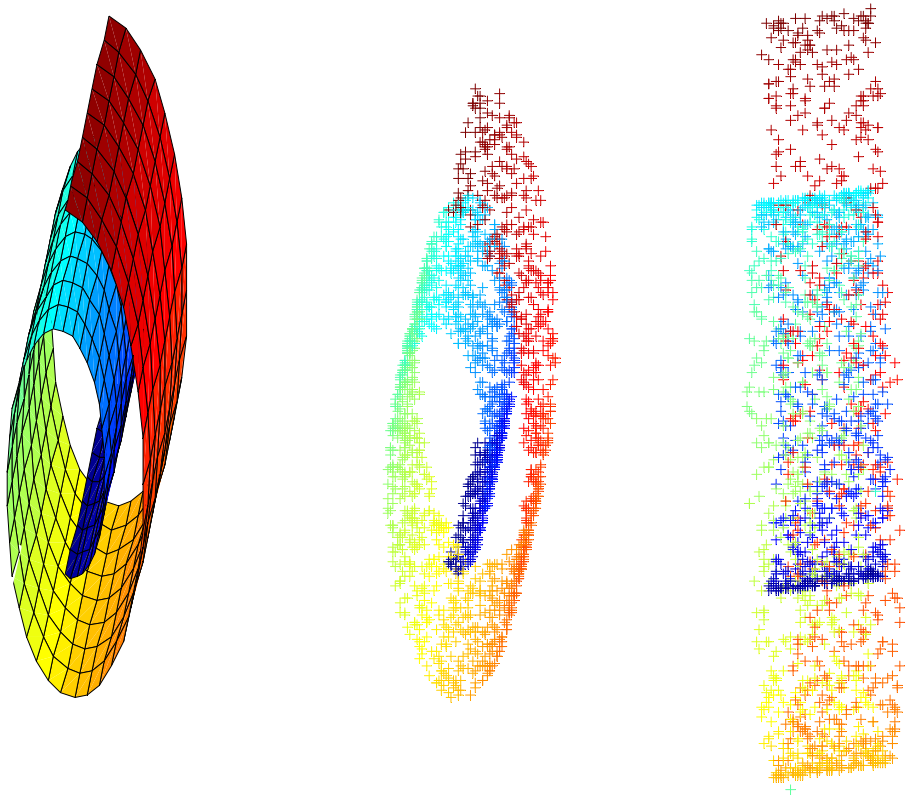
$$\Sigma = 1/n \sum x_i \, x_i^T$$

- Diagonalize $\Sigma = U \, \Lambda \, U^T$

PCA does poorly when data is on a nonlinear manifold

An Example where PCA Fails

"Swissroll"



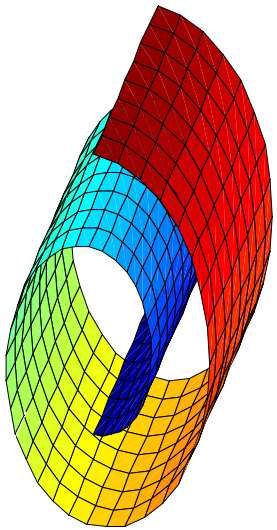
PCA

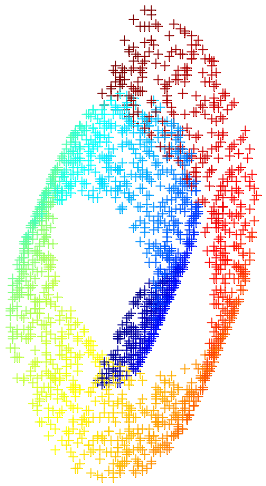
PCA finds the direction of "maximal variance"

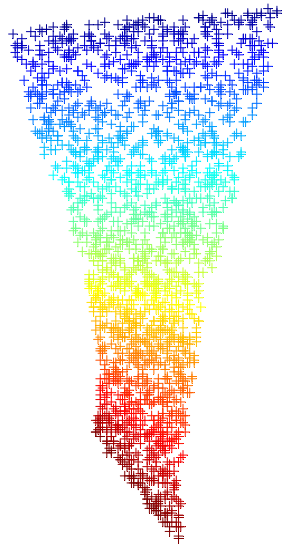
This does not preserve "locality"

Nonlinear dimensionality reduction

"Swissroll"







Embedding

Embedding should preserve "locality"

Graph Embedding

- Consider the following optimization problem mapping, where $y_i \in R$ is a mapping of the i^{th} vertex to the real line

$$\text{Min}_y \sum_{i,j} (y_i - y_j)^2 w_{i,j} \quad \text{s.t. } y^T D y = 1$$

- The best mapping is found by solving the generalized eigenvector problem

$$W \phi = \lambda D \phi$$

- If the graph is connected, this can be written as

$$D^{-1} W \phi = \lambda \phi$$

Random Walk Operator

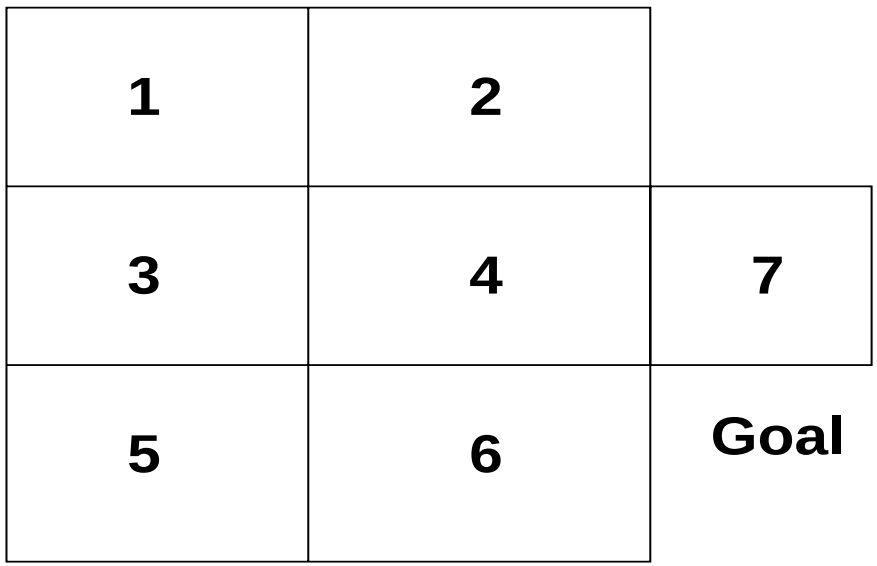
1	**2**	
3	**4**	**7**
5	**6**	**Goal**

Random Walk
Matrix = $D^{-1}W$

0	.5	.5	0	0	0	0
.5	0	0	.5	0	0	0
.33	0	0	.33	.33	0	0
0	.25	.25	0	0	.25	.25
0	0	.5	0	0	.5	0
0	0	0	.5	0	.5	0
0	0	0	1	0	0	0

Non-symmetric!

Combinatorial Graph Laplacian



Laplacian
Matrix = L = D-W

Row sums

2	-1	-1	0	0	0	0
-1	2	0	-1	0	0	0
-1	0	3	-1	-1	0	0
0	-1	-1	4	0	-1	-1
0	0	-1	0	2	-1	0
0	0	0	-1	-1	2	0
0	0	0	-1	0	0	1

Negative of weights

Properties of the Laplacian

- The Laplacian L is *positive semidefinite*

- The Laplacian for this graph is = $\begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}$

- Note that $\langle f, Lf \rangle = f^T L f = (f_1 - f_2)^2$
- Hence, for any $f \neq 0$, $\langle f, Lf \rangle \geq 0$
- All the eigenvalues of L are non-negative
- Combinatorial Laplacian L = D – W acts on f

$$(L f)(i) = \sum_{i \sim j} (f_i - f_j) w_{ij}$$

Dirichlet Sums

- The quadratic form <f, Lf> is given by

$$\sum_i f_i \, (Lf)_i = \sum_i f_i \sum_j (f_i - f_j) \, w_{ij}$$

- Note that for each term of the form $f_i(f_i - f_j)$, there must be another term of the form $f_j(f_j - f_i)$

- We can express <f, Lf> as a *Dirichlet* sum

$$\langle f, Lf \rangle = \sum_{(u,v) \in E} (f_u - f_v)^2 \, w_{ij}$$

- The pseudo-inverse of the Laplacian L^+ defines a reproducing kernel Hilbert space (RKHS)
  - The quadratic form <f, Lf> induces a regularization prior that favors smooth functions
  - Laplacian embedding is a form of kernel PCA

Laplacian and Random Walks on a Graph

Operator	Spectrum
Adjacency $= W$	Real, $\lvert\lambda\rvert \leq d_v$
Combinatorial Laplacian $= D - W$	PSD, $\lambda \in [0, d_v]$
Normalized Laplacian $= I - D^{-1/2} W D^{-1/2}$	PSD, $\lambda \in [0, 2]$
Random Walk $= D^{-1} W$	$\lambda \in [-1, 1]$

$D^{-1} W = D^{-1/2} (D^{-1/2} W D^{-1/2}) D^{1/2} = D^{-1/2} (I - \mathcal{L}) D^{1/2}$

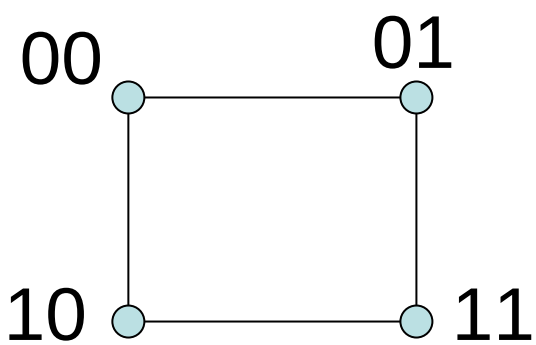
Hence, $D^{-1}W$ and $I - \mathcal{L}$ have the same eigenvalues

Spectral bounds follow
from Gershgorin's theorem

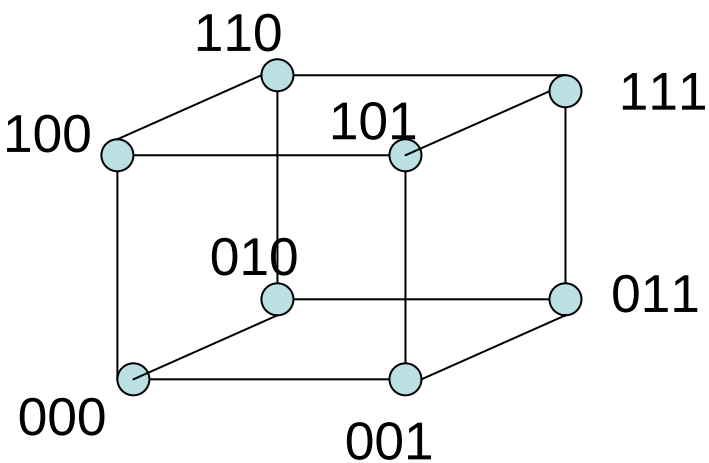
Laplacian on Boolean Hypercube

$L_n = n\, I_n - A_n$

Eigenvectors of L_n = columns of H_n
(H_n = *Hadamard* Matrix)



Eigenvalues of $L_n = 2\,[i]$, $0 \leq i \leq 2^{n-1}$



$$H_1 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

$$H_n = \begin{vmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{vmatrix}$$

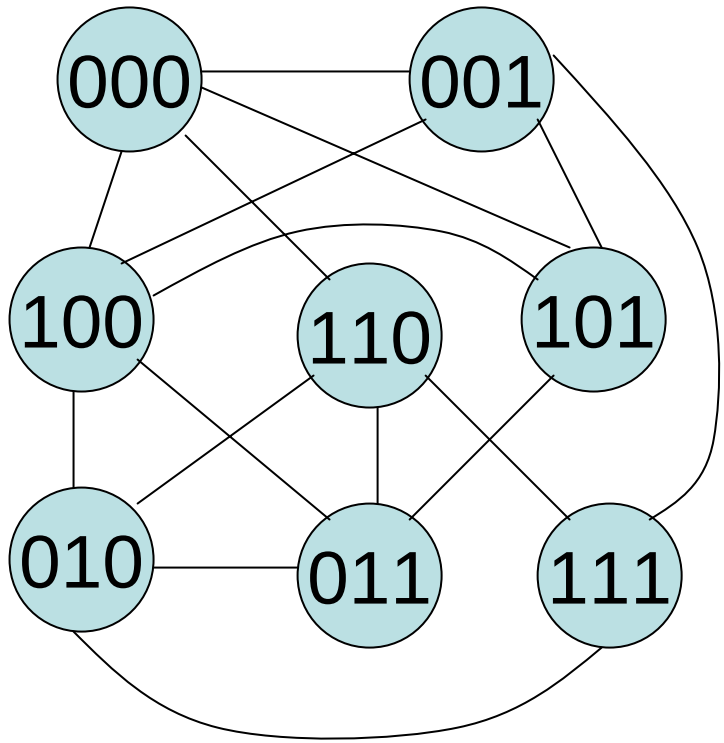
Spectral Representation of Boolean Functions

(Bernasconi, '98)

$f = x_1 \neg x_3 \vee \neg x_2 x_3$

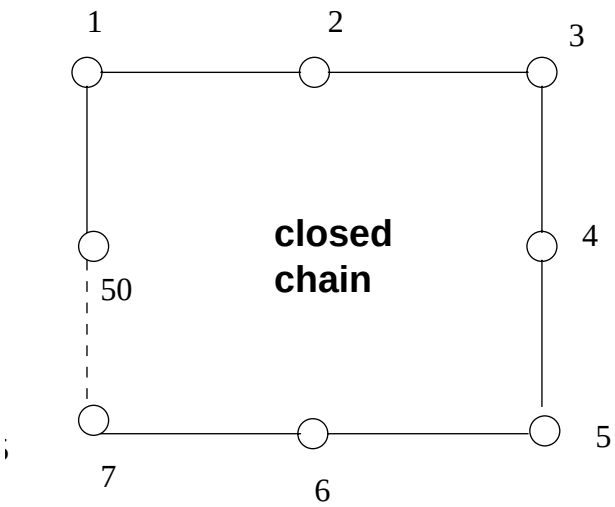
x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

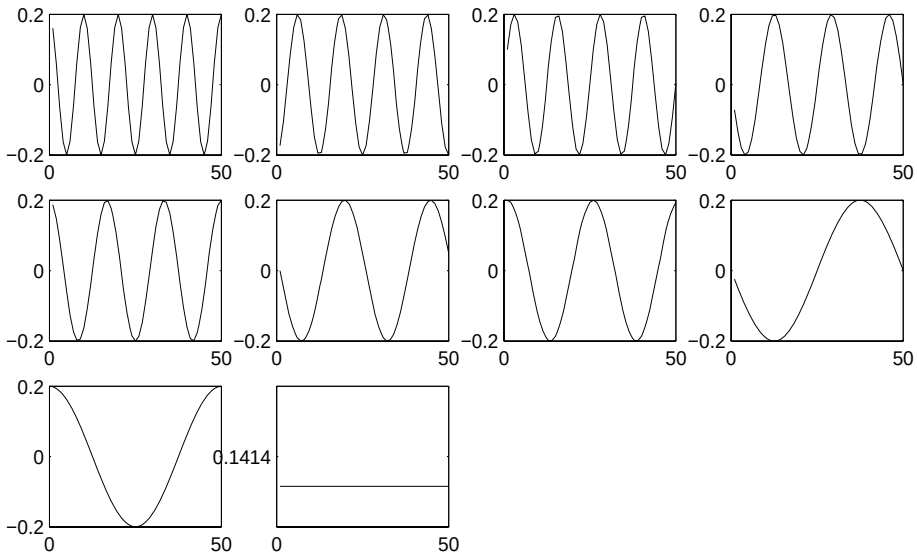
$f(m_1 \oplus m_2) = 1$



Eigenvalues of this graph
= Fourier representation!

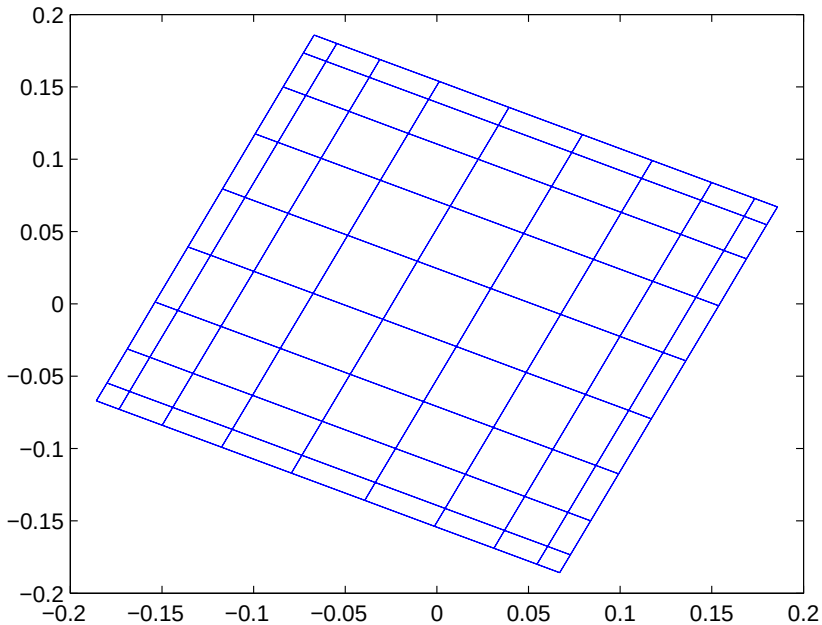
Fourier Basis on 1D Graph





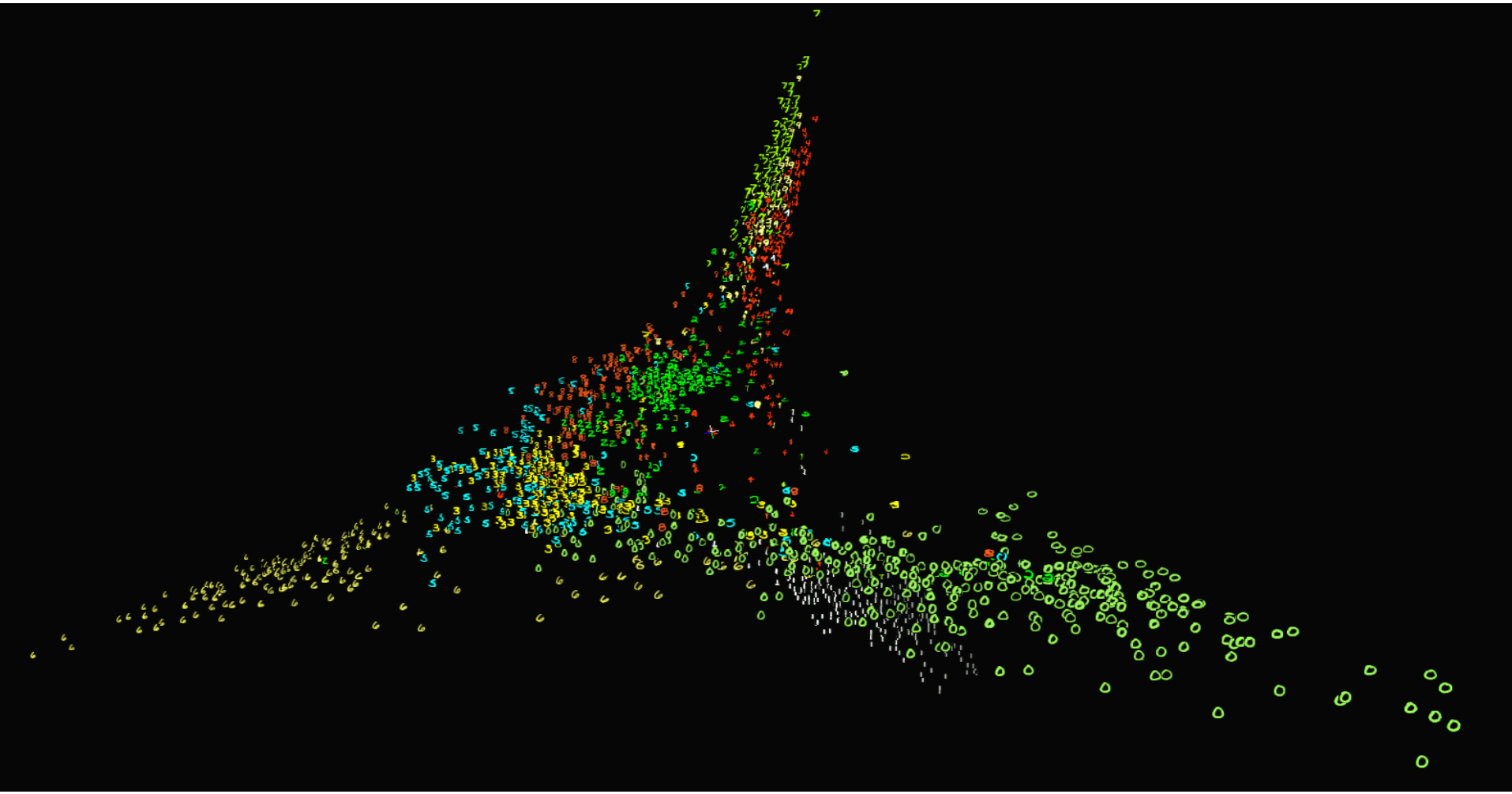
Fourier Basis:
Eigenvectors of the Graph Laplacian

Laplacian Embedding



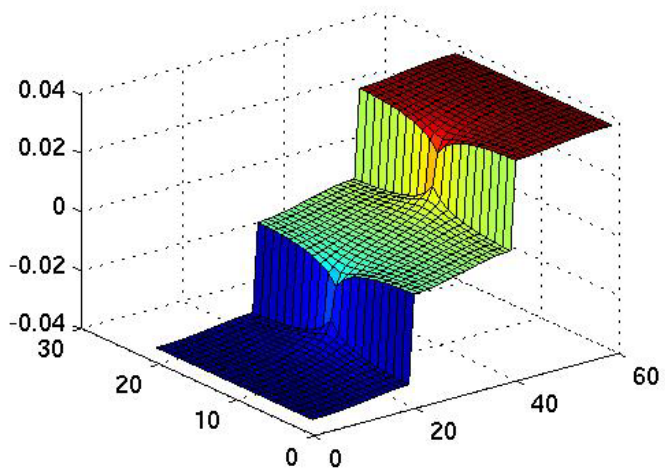
Embedding of a 10x10 grid using the 2nd and 3rd eigenvectors

Laplacian Embedding of Digit Data

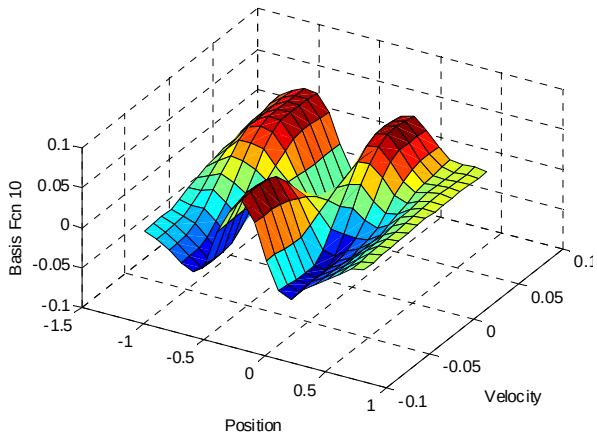


Eigenvectors of Graph Laplacian:
Discrete and Continuous Domains

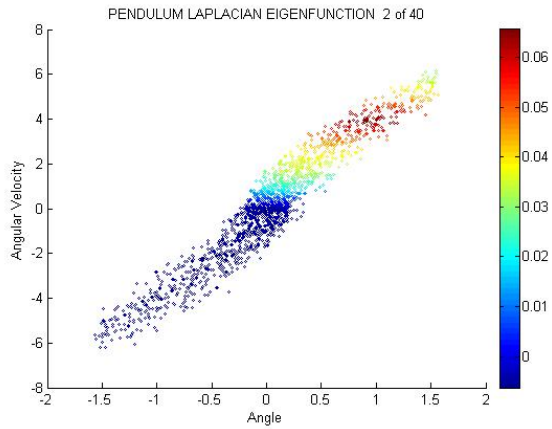
Three rooms with bottlenecks



Mountain Car Problem

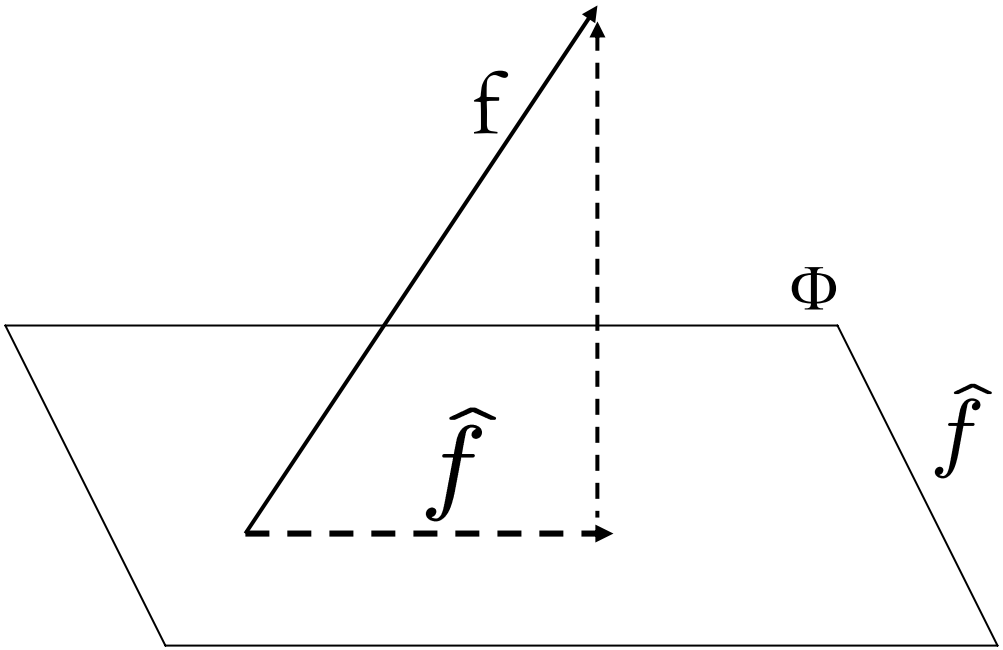


Inverted pendulum



3D Compression

Least-Squares Projection



Theory of least-squares tells us how to find the closest vector in a subspace to a given vector

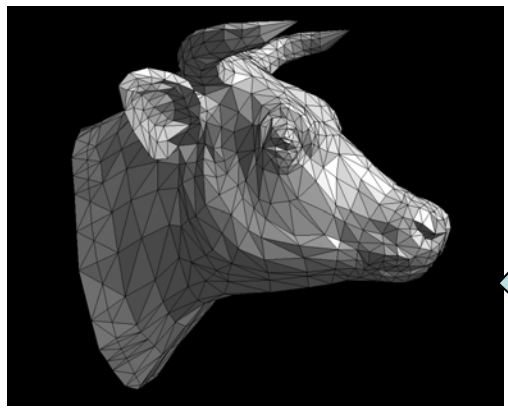
$$\hat{f} = \Phi(\Phi^T\Phi)^{-1}\Phi^T f$$

What is an optimal subspace Φ for approximating f on a graph?

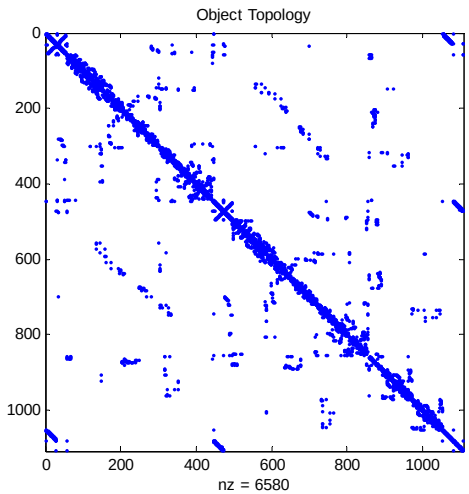
Standard bases (polynomials, RBFs) don't exploit geometry of graph

Compression of 3D Objects

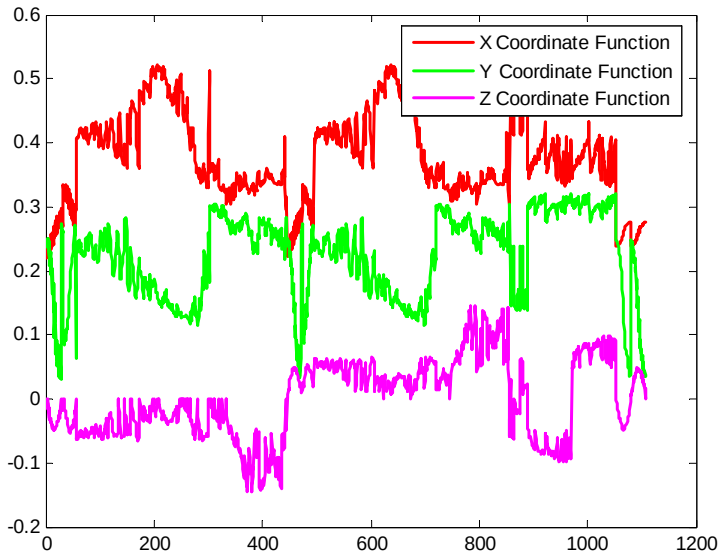
~1000 vertices
~ 25 Kb



Topology



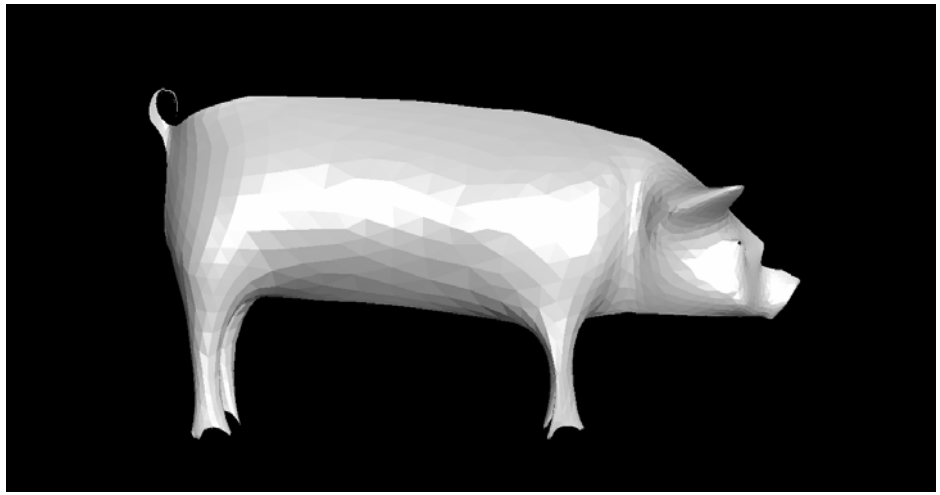
Geometry



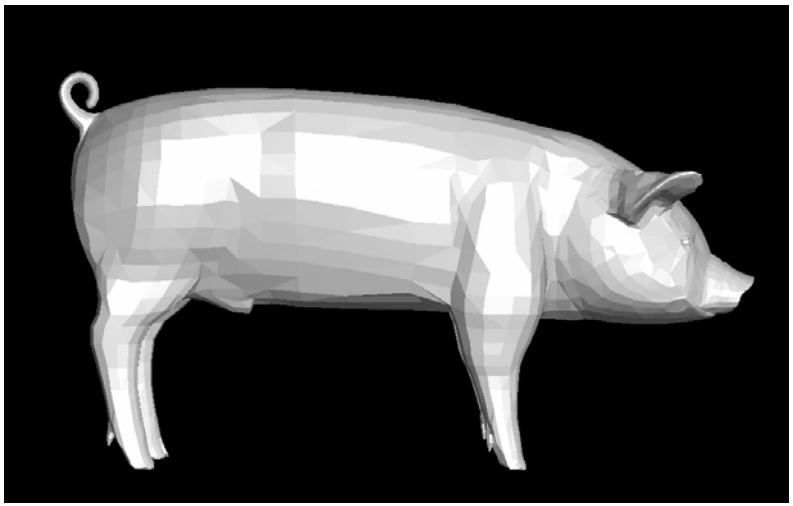
Laplacian Compression of 3D Objects

(Karni and Gotsman, SIGGRAPH 2000)

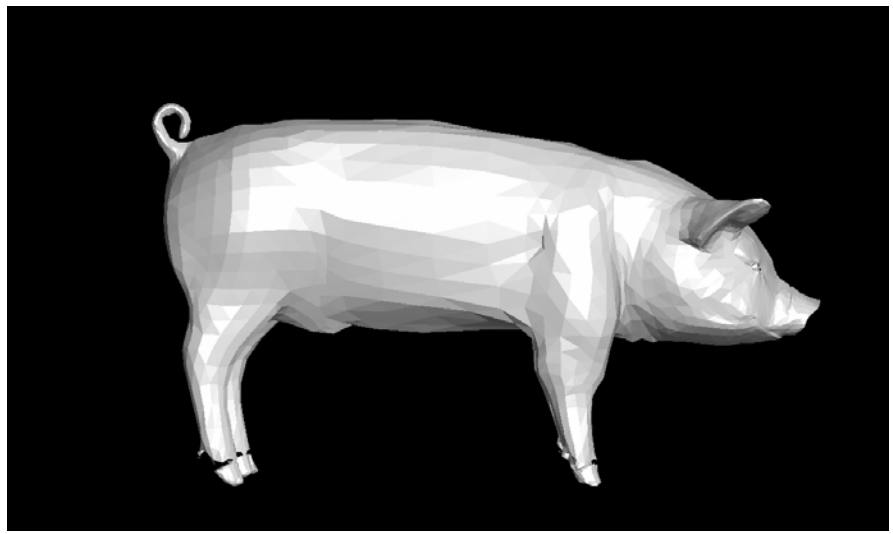
100 Basis Functions



Original object



800 Basis Functions

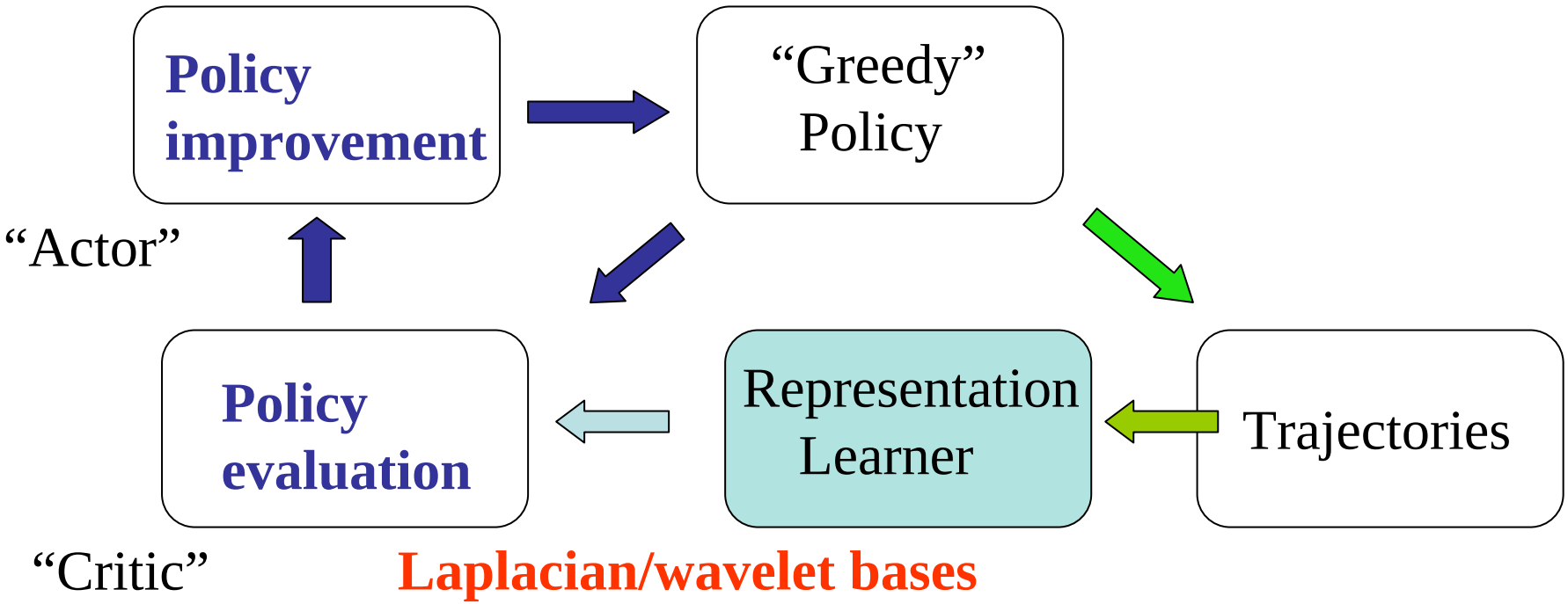


$$f \sim \sum_{i \in I} \langle f, \phi_i \rangle \phi_i$$

Fourier series expansion

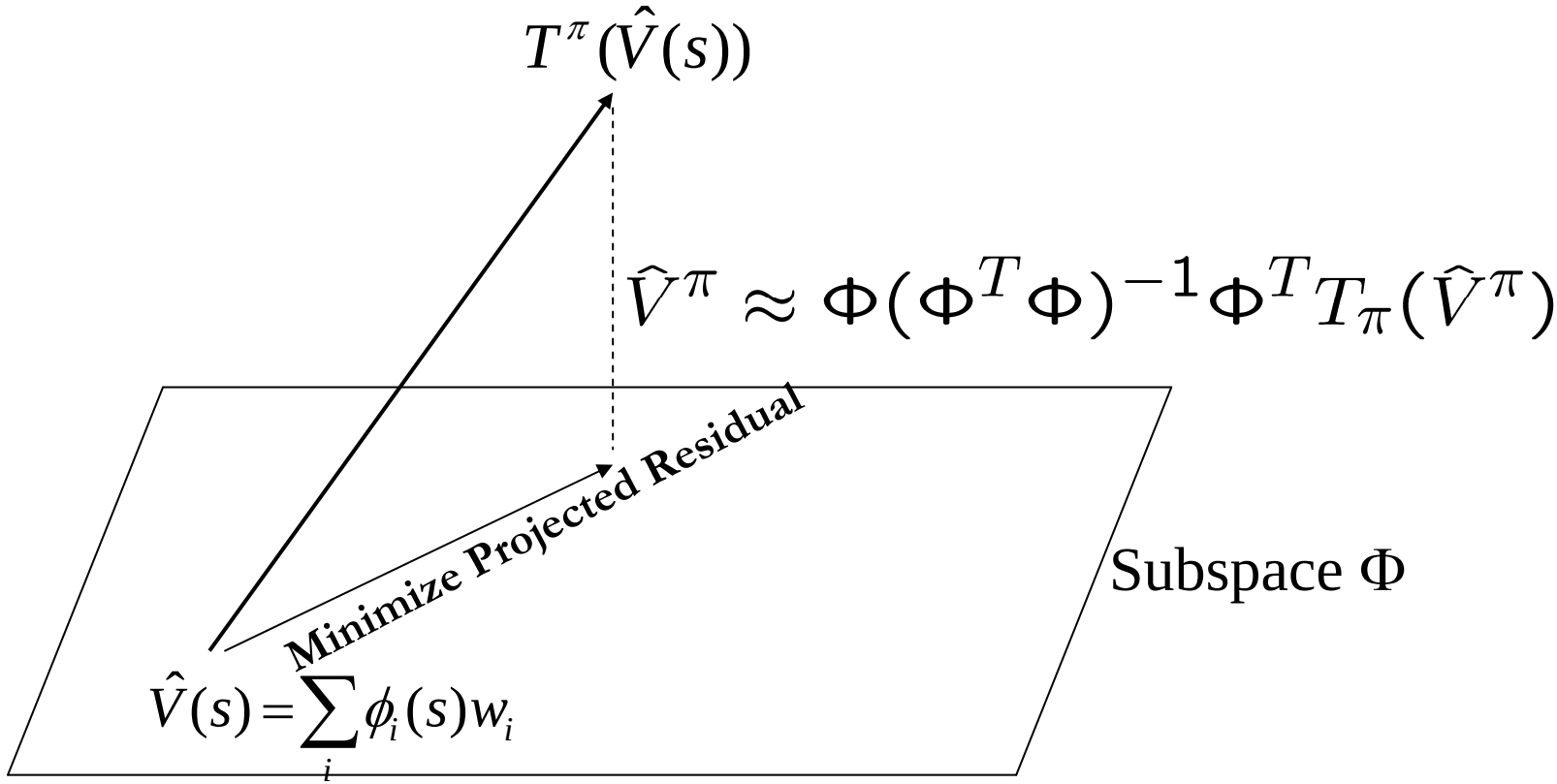
Representation Policy Iteration

(Mahadevan, UAI 2005)



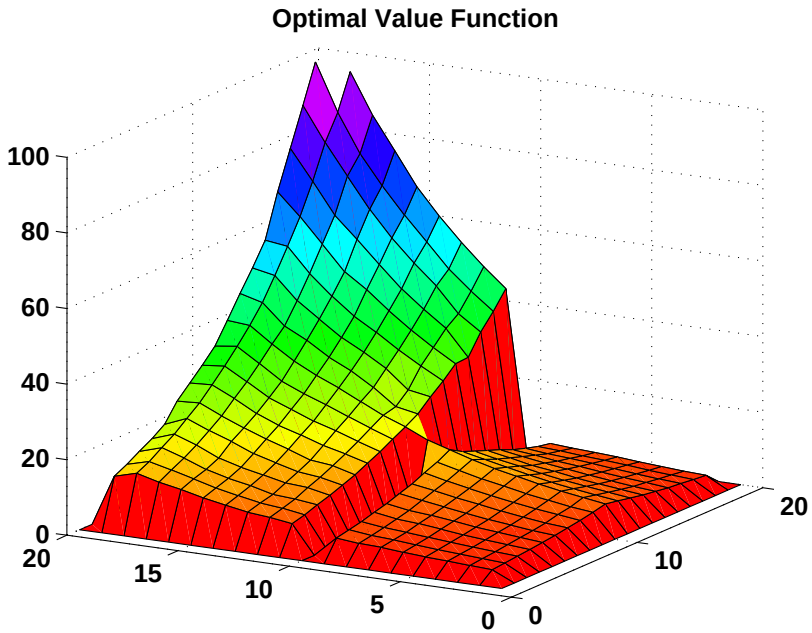
VFA using Least-Squares Projection

(Boyan, Bradtke and Barto, Bertsekas and Nedic, Lagoudakis and Parr)

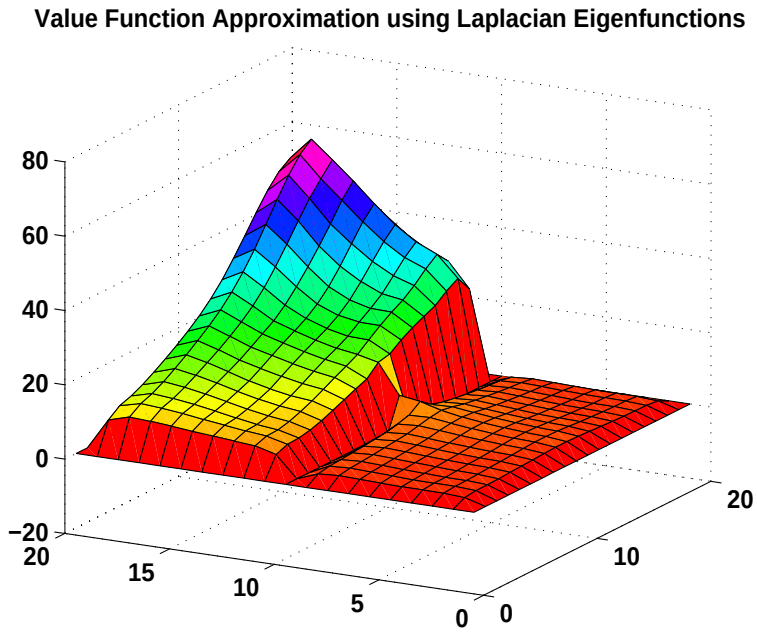


RPI: Two-Room World

(Mahadevan, ICML 2005)



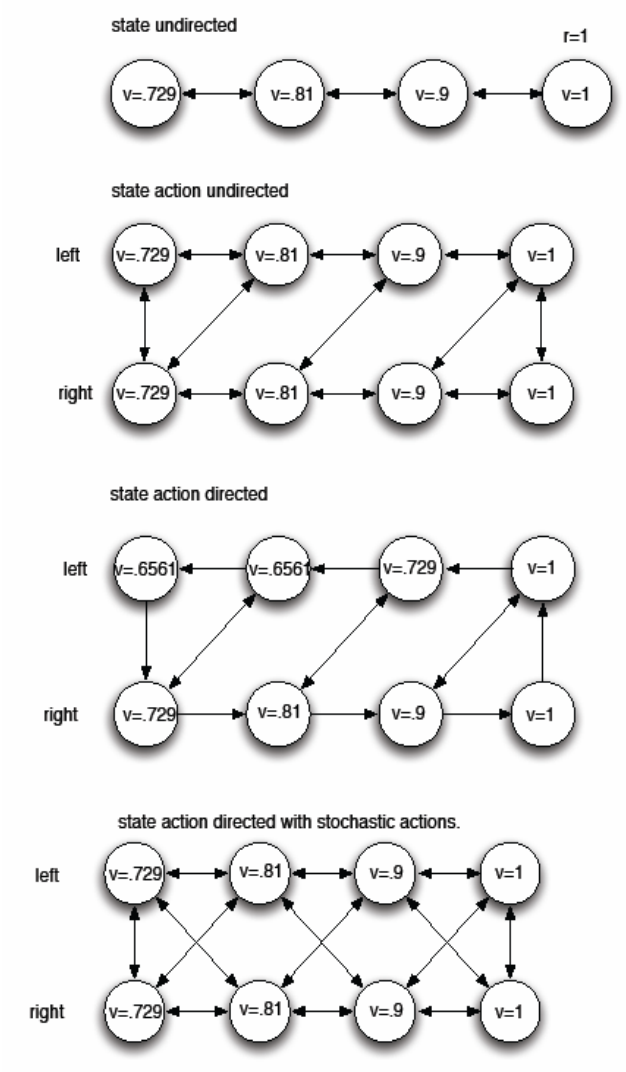
OPTIMAL VF



LAPLACIAN BASIS

Graph Construction

- The graph assumes a *local* similarity metric
  - In discrete MDPs, state s is connected to s' if an action led the agent from $s \rightarrow s'$
- Distance metrics:
  - Nearest neighbor: connect an edge from s to s' if s' is one of k nearest neighbors of s
  - Heat kernel: connect s to s' if $|s - s'|^2 < \varepsilon$ with $w(s,s') = e^{-|s-s'|^2/2} < \varepsilon$
- Weights can depend on target function
  - The gradient of the function

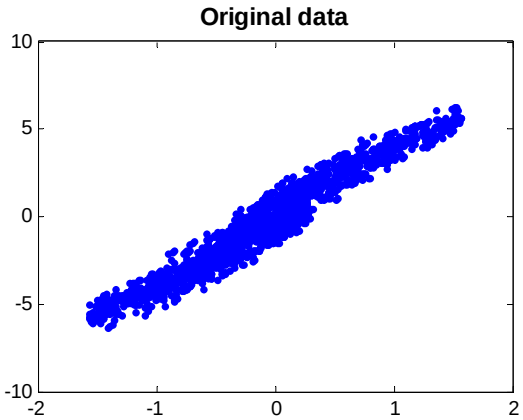


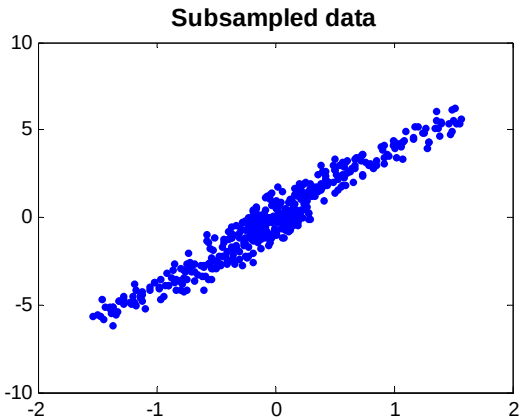
Manifold Construction in Continuous MDPs

(Mahadevan, Maggioni, Ferguson, Osentoski, AAAI 2006)

- How to deal with new samples?
  - The Nystrom extension interpolates eigenfunctions from sample points to new points
- Many practical issues are involved
  - How many samples to use to build the graph?
  - Local distance metric: Gaussian distance, k-NN
  - Graph operator: Normalized Laplacian, Combinatorial Laplacian, Random Walk, …
  - Type of graph: Undirected, directed, state-action graph

Sampling from a Continuous Manifold

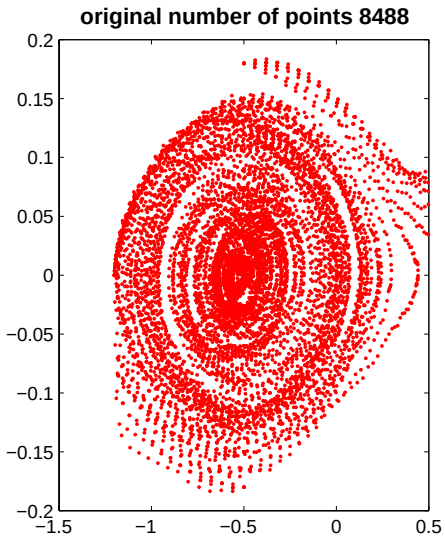


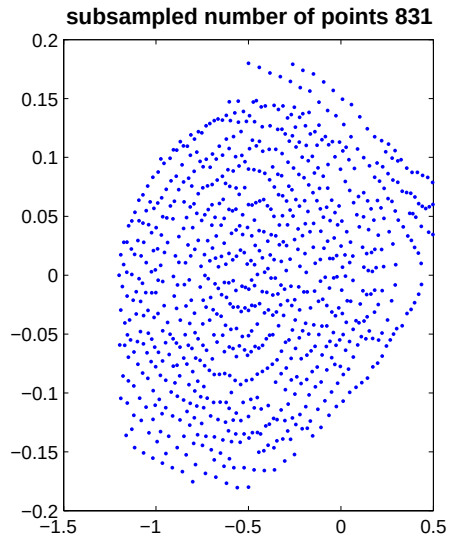


Random

Inverted
Pendulum

Trajectory subsampling



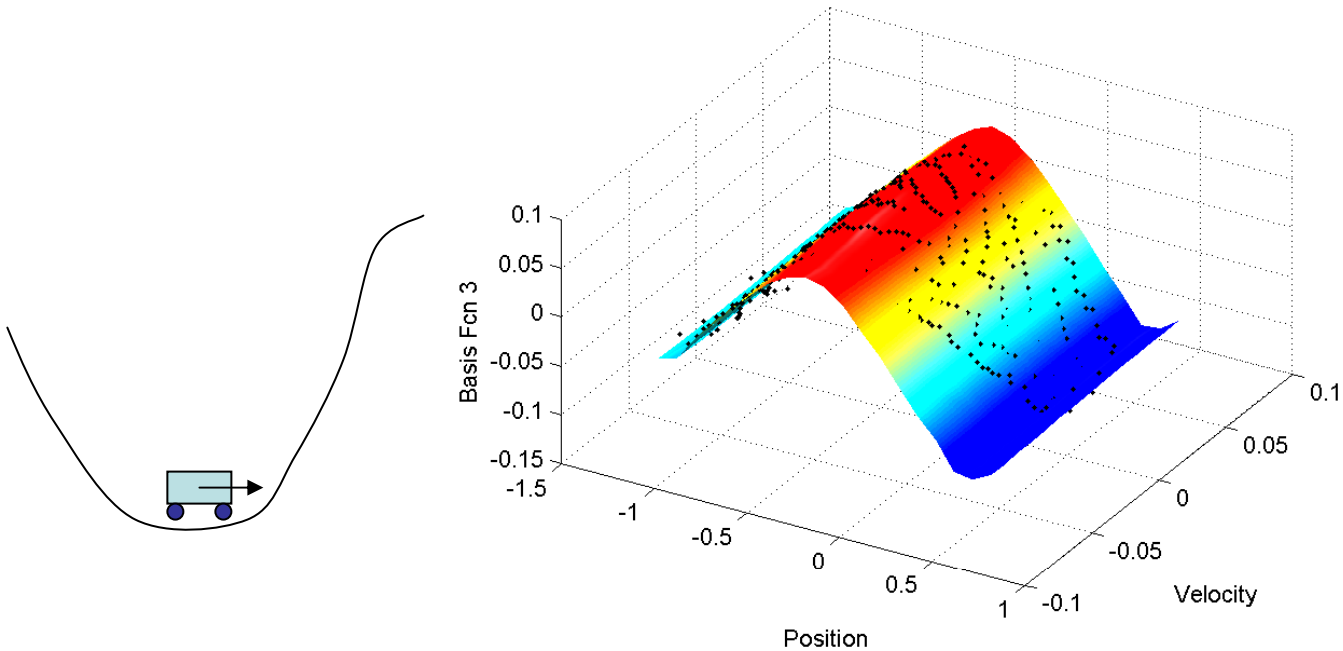


Mountain
Car domain

Out of Sample Extension

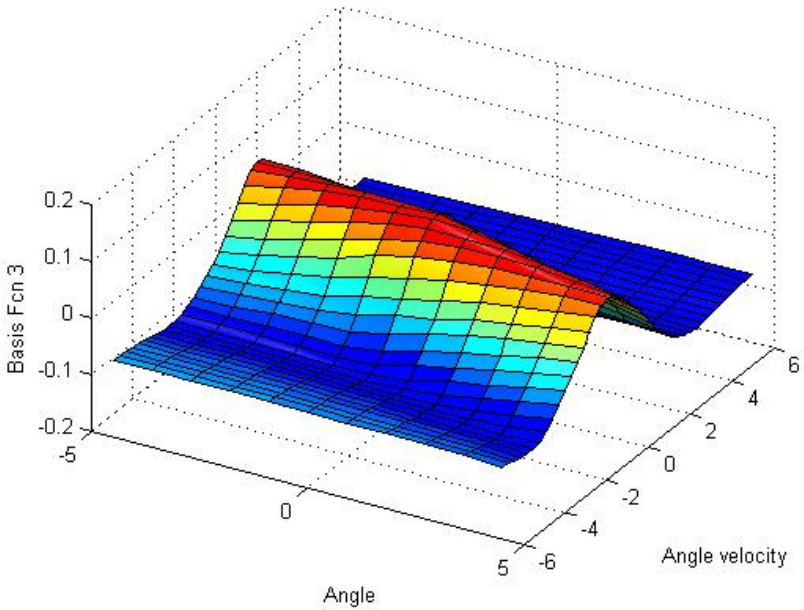
- The testing of a learned policy requires computing the basis in novel states
- The Nystrom extension is a classical method developed in the solution of integral equations

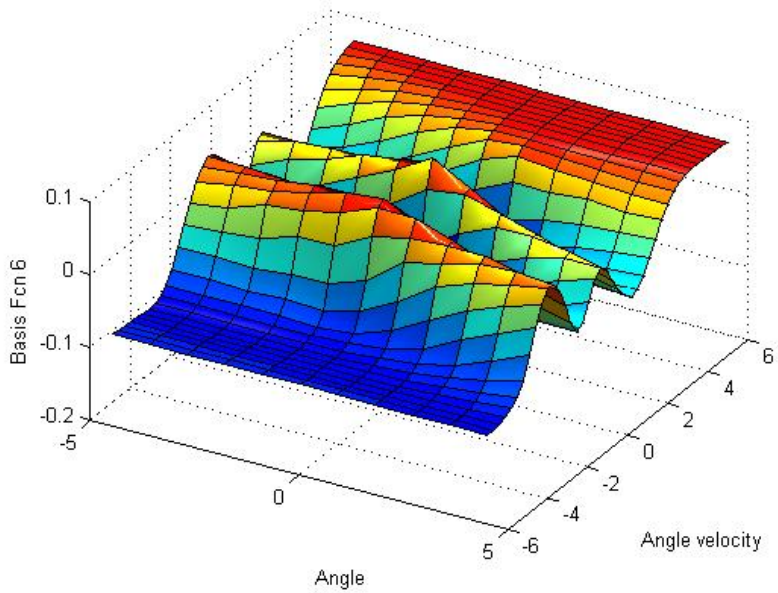
$$\phi_m(x) = 1/\lambda_m \sum_j w_j k(x,s) \phi_m(s_j)$$

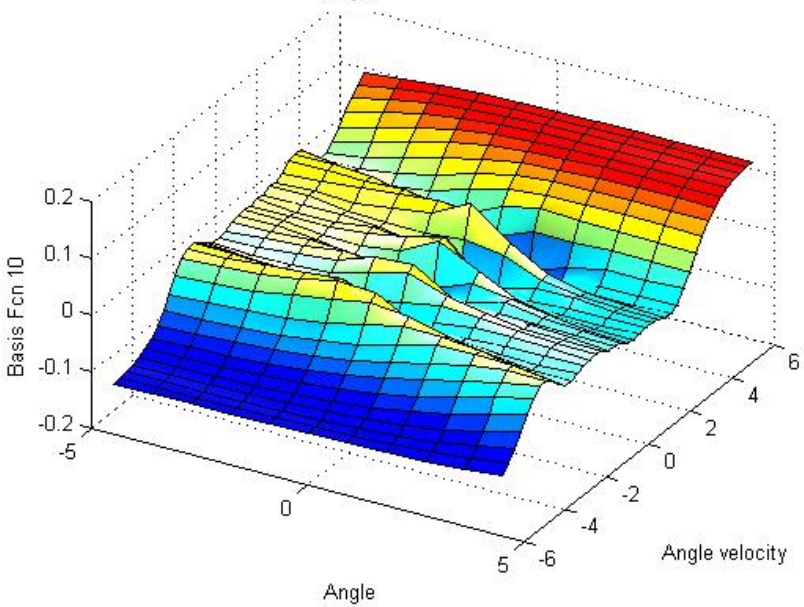


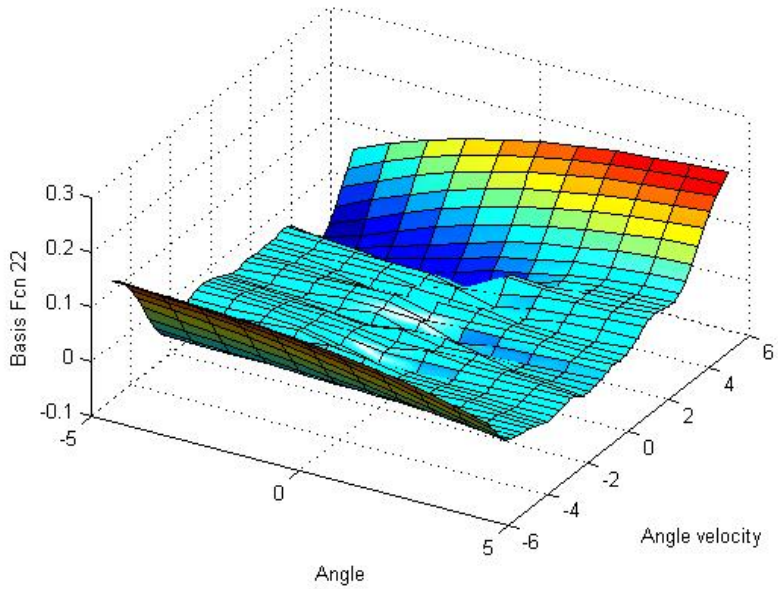
Mountain Car MDP

Pendulum Proto-Value Functions

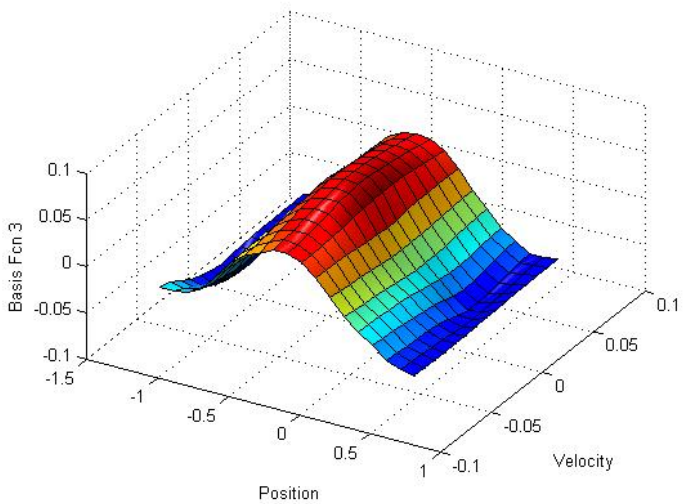


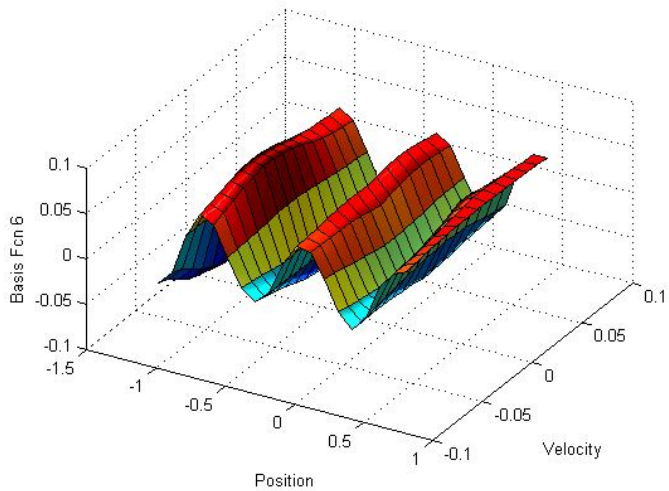


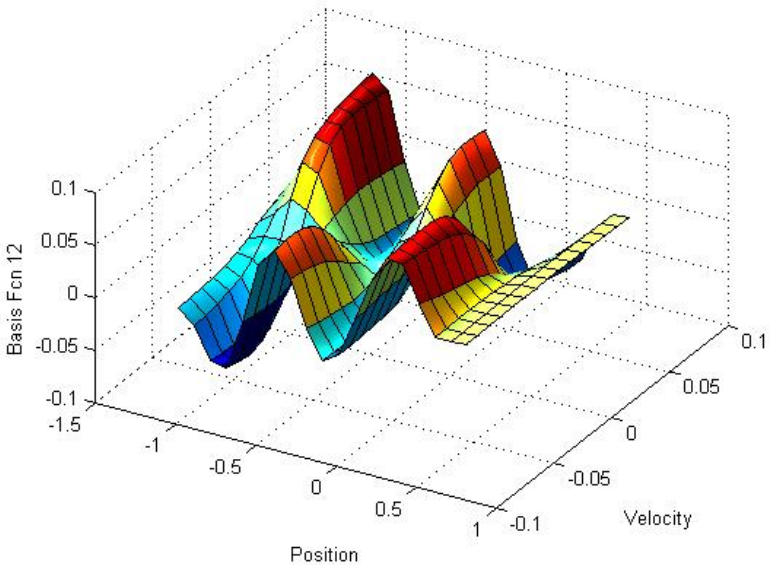


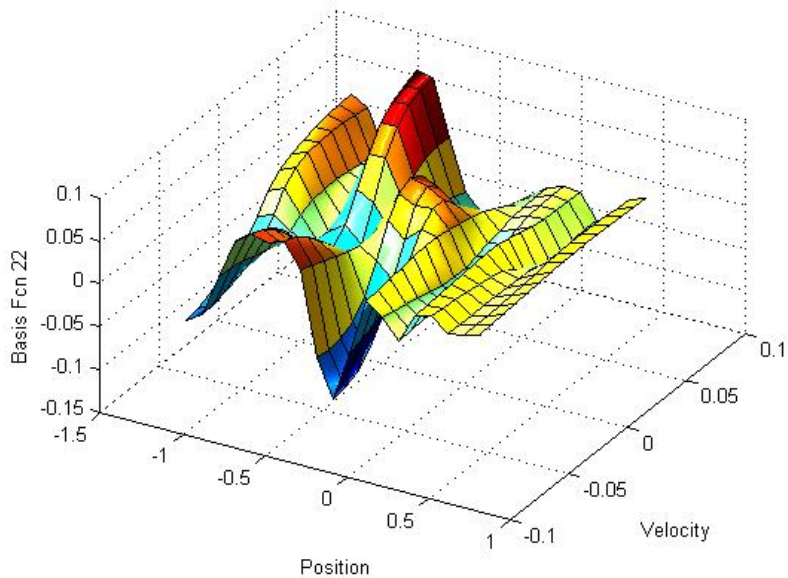


Mountain Car Proto-Value Functions

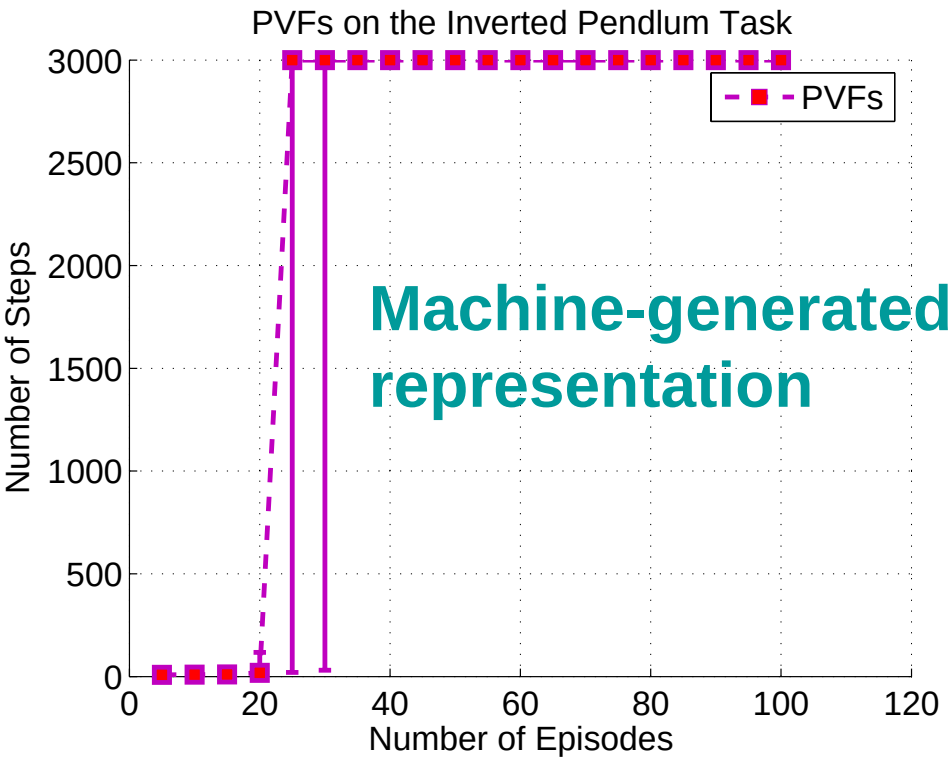




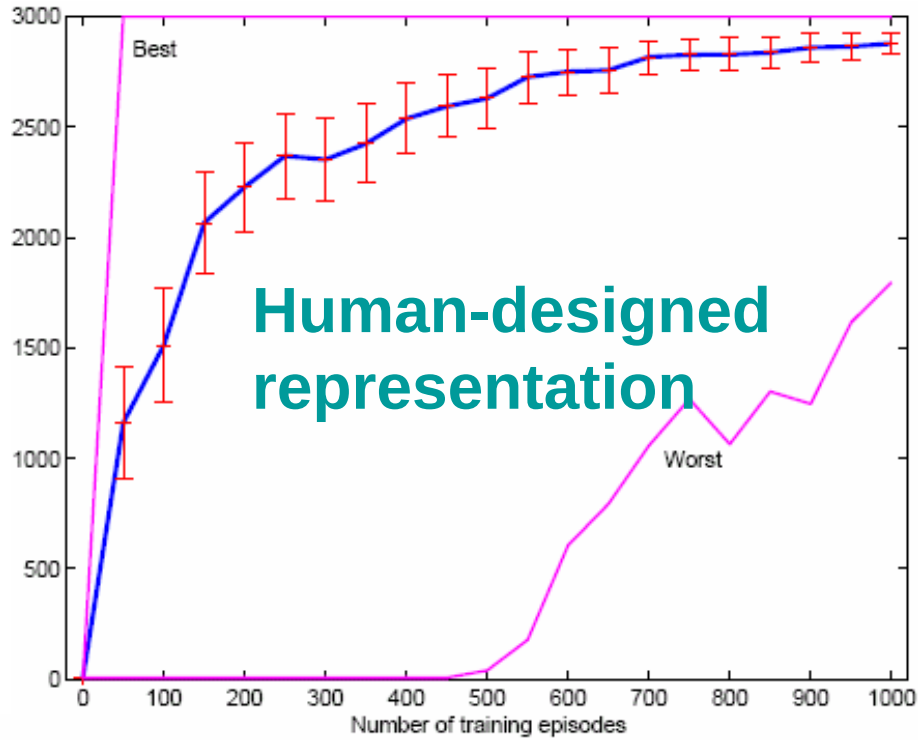




RPI in Continuous Domains: Inverted Pendulum

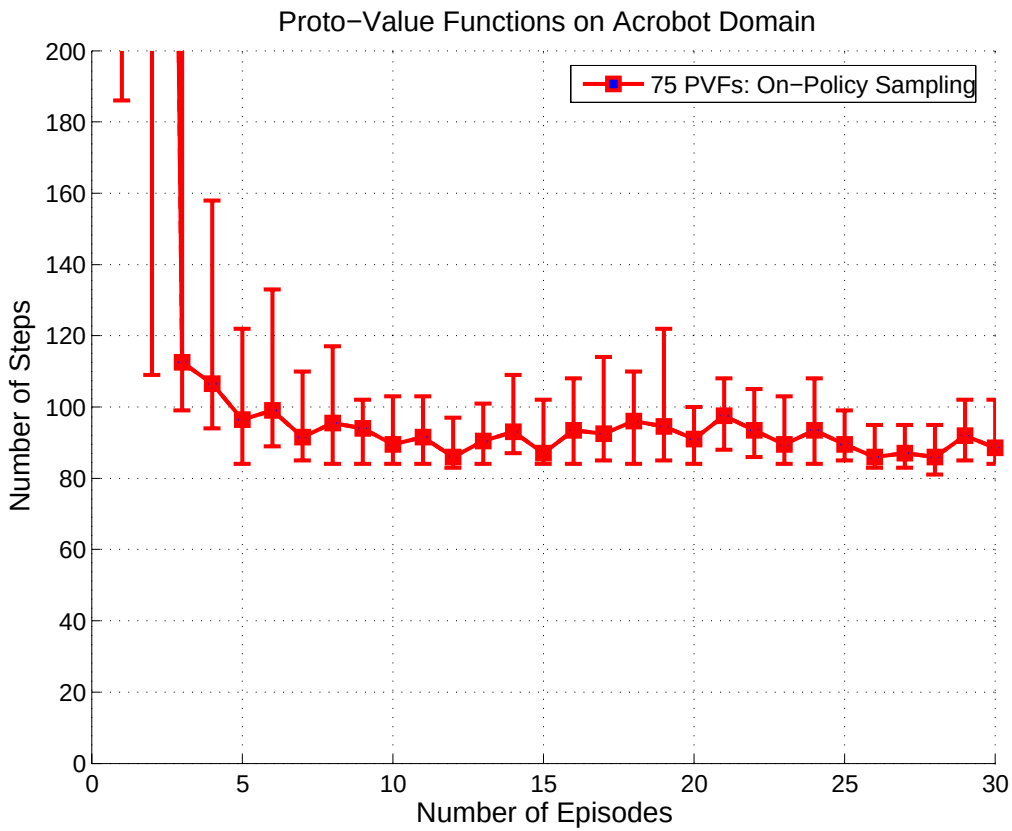


[Lagoudakis and Parr, JMLR 2003]

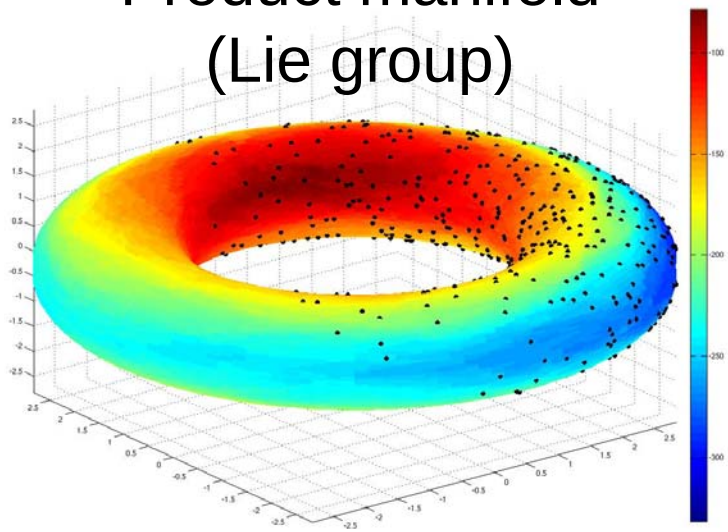


Each episode:
random walk of 6-7 steps

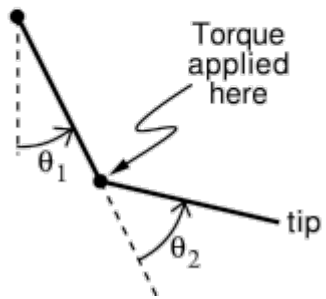
Acrobot Task



Product manifold
(Lie group)

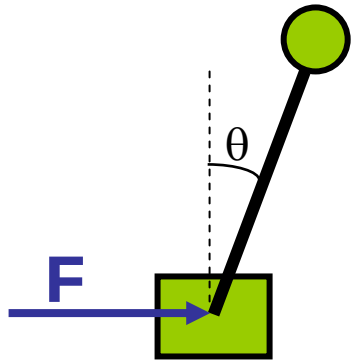


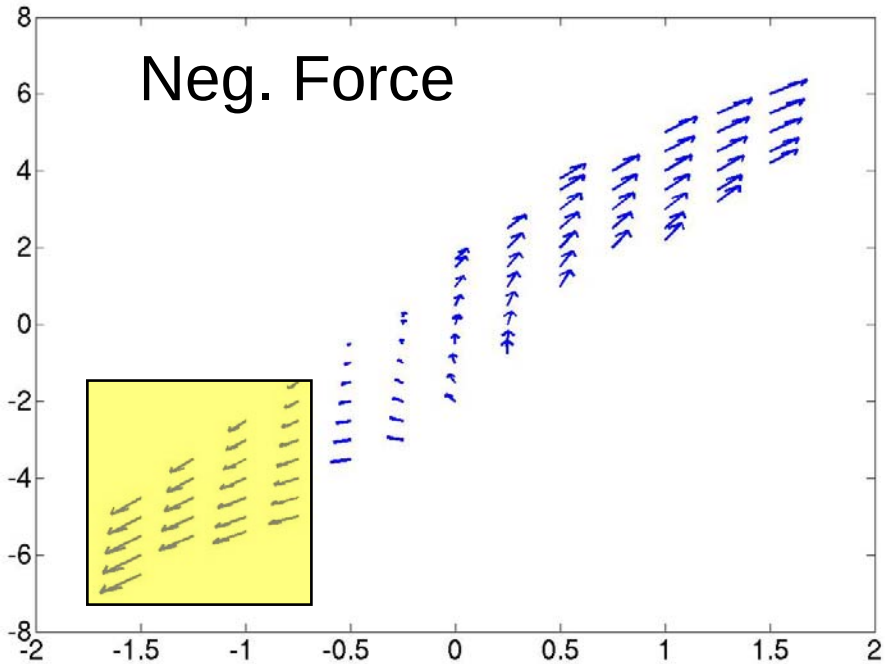
Goal: Raise tip above line

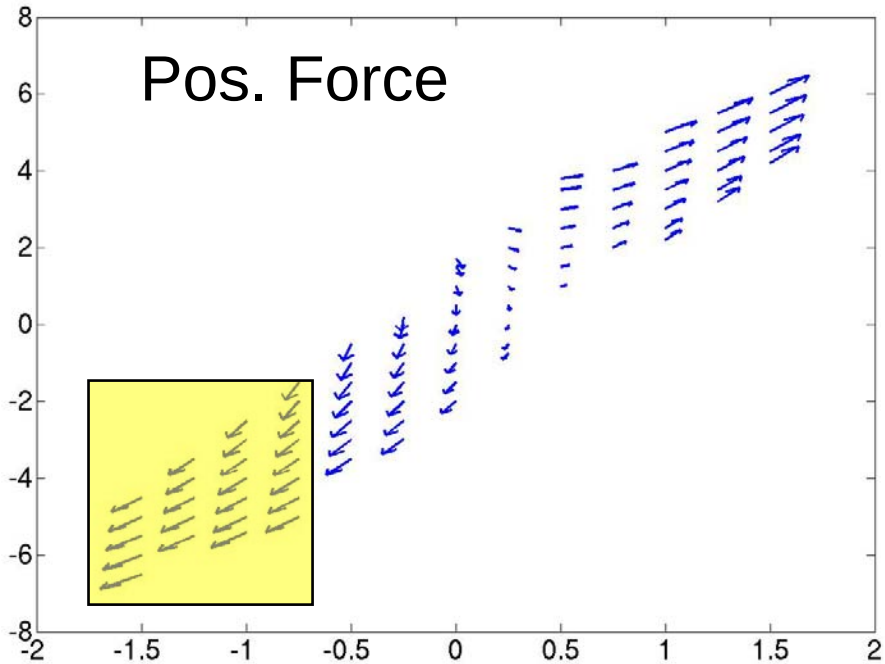


Directionality in Inverted Pendulum

- 2d state space (angle, angular velocity)
- 3 actions (+, 0, - force)
- Many non-reversible actions







Directed Laplacian
[Chung '05]

- Digraph weight matrix W (potentially asymmetric)
  - $P = D^{-1} W$ (stochastic random walk matrix)

- Invariant distribution ψ
  - Definition: $\psi^T P = \psi^T$ and $\sum \psi_i = 1$
  - Ψ = diagonal matrix with $\Psi(i,i) = \psi_i$
  - Also referred to as the *Perron vector*

- Directed Graph Laplacian: $\Psi - (\Psi P + P^T \Psi)/2$

Invariant Distribution ψ

- Perron-Frobenius Theorem ensures P has a unique left eigenvector ψ with all positive values as long as the graph is strongly connected

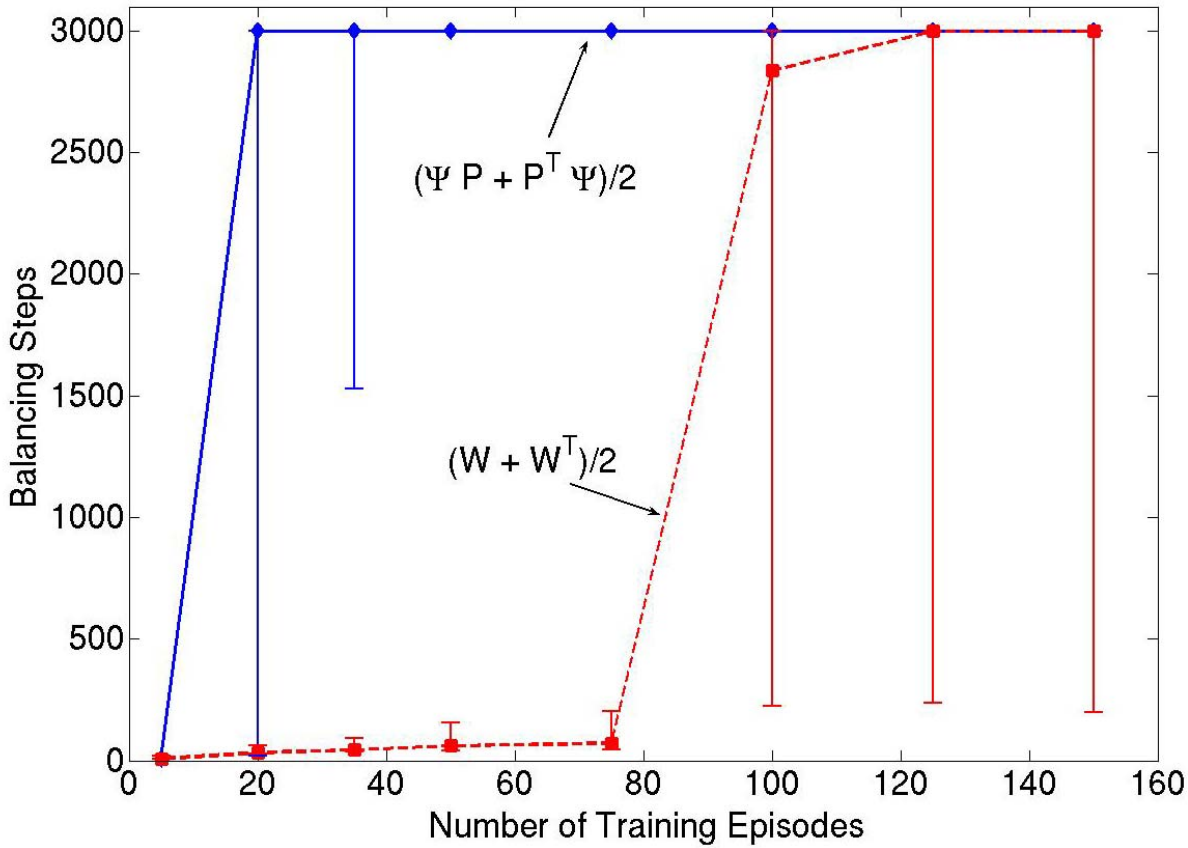
- Teleporting random walk ensures the graph is strongly connected
  - $P_{teleport} = \eta P + (1-\eta)(\mathbf{1}\,\mathbf{1}^T)/n$ $\quad (\eta \approx 0.99)$

- Use the Power method to compute ψ
  - Iterate: $\psi_i^T P_{teleport} = \psi_{i+1}^T$ until ψ converges
  - Computing ψ is the only additional cost for the directed Laplacian

Inverted Pendulum Results

(Johns and Mahadevan, ICML 2007)

- Improved performance with less training data (results using 8 basis functions)

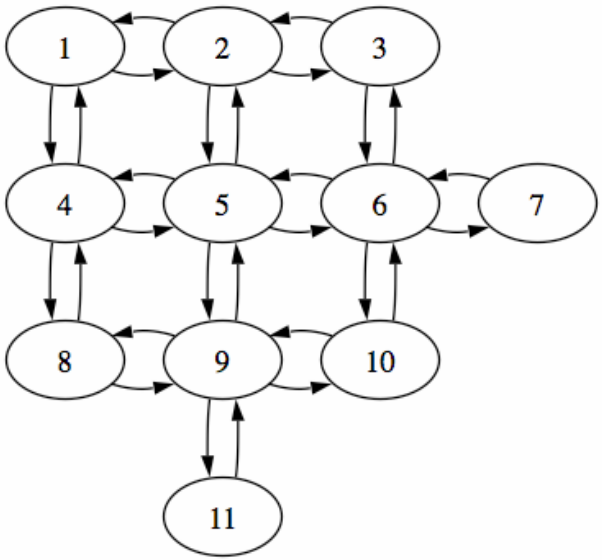


Learning Basis Functions in SMDPs using state-Action Graphs

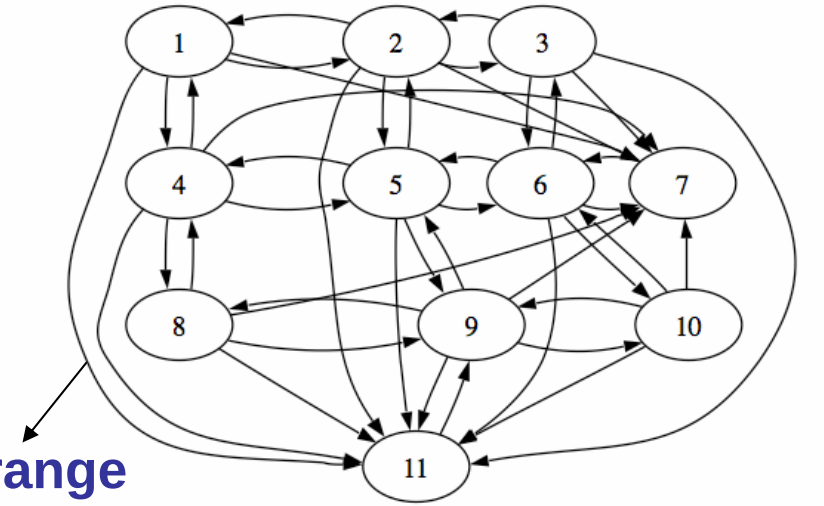
(Osentoski and Mahadevan, ICML 2007)

- Q(s,a) is a function over states and actions
- Thus far, we have generated basis functions for Q by "copying" basis functions $\phi(s)$ over states $|A|$ times
- A more efficient method is to directly generate state-action bases by diagonalizing the directed Laplacian on state-action graphs
- We can also exploit the hierarchical nature of actions by using semi-Markov decision processes

State Graphs



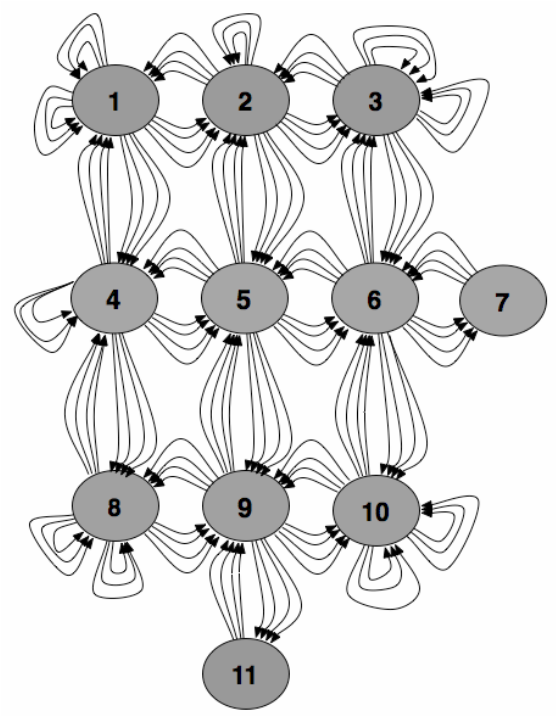
State Graph with primitive actions



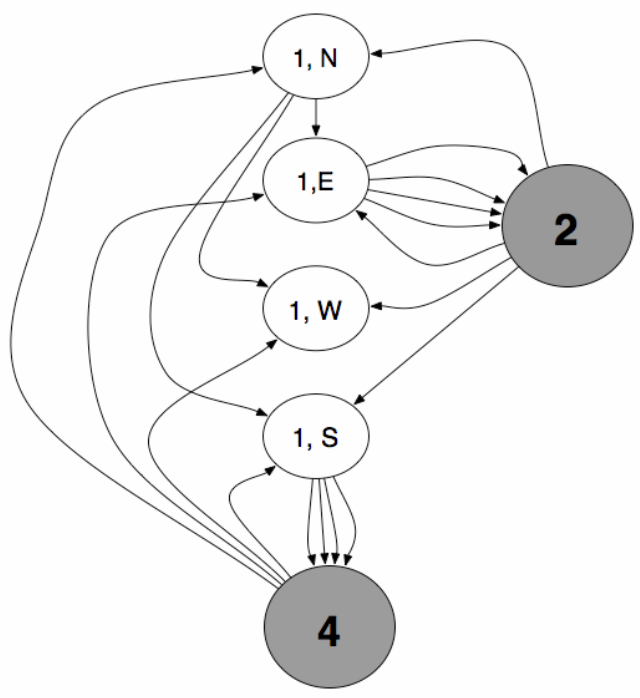
Long-range connections

State Graph in SMDPs with temporally extended actions

State-Action Graphs

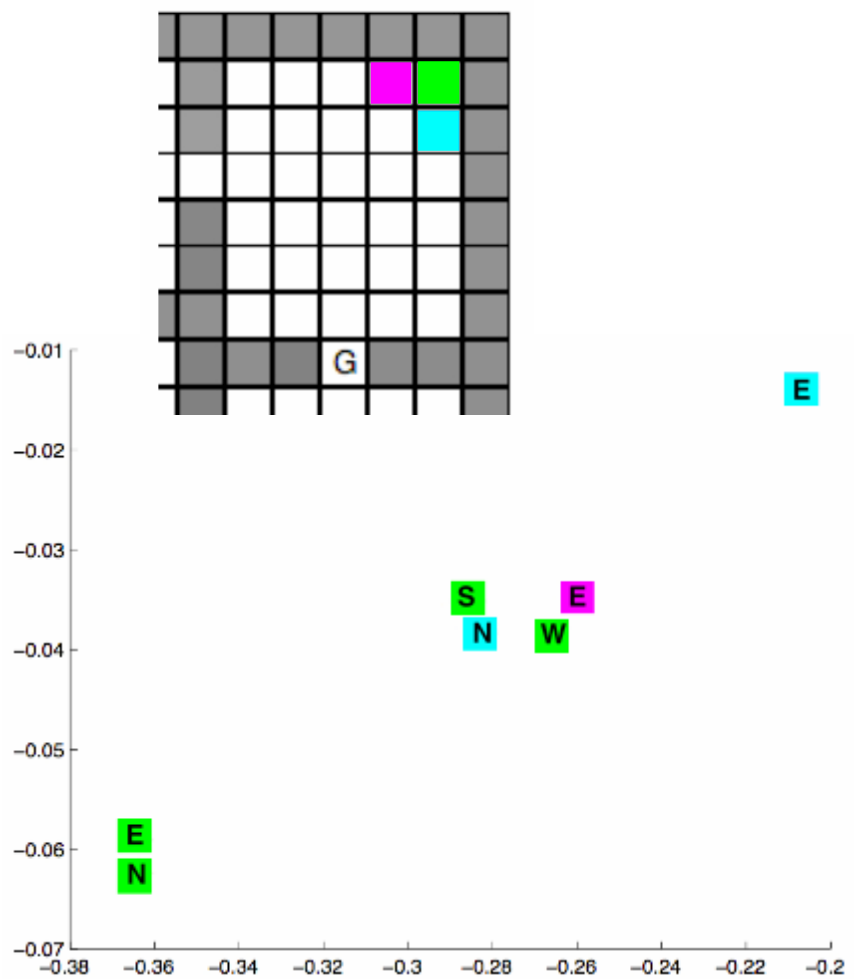


State-action graph with primitive actions

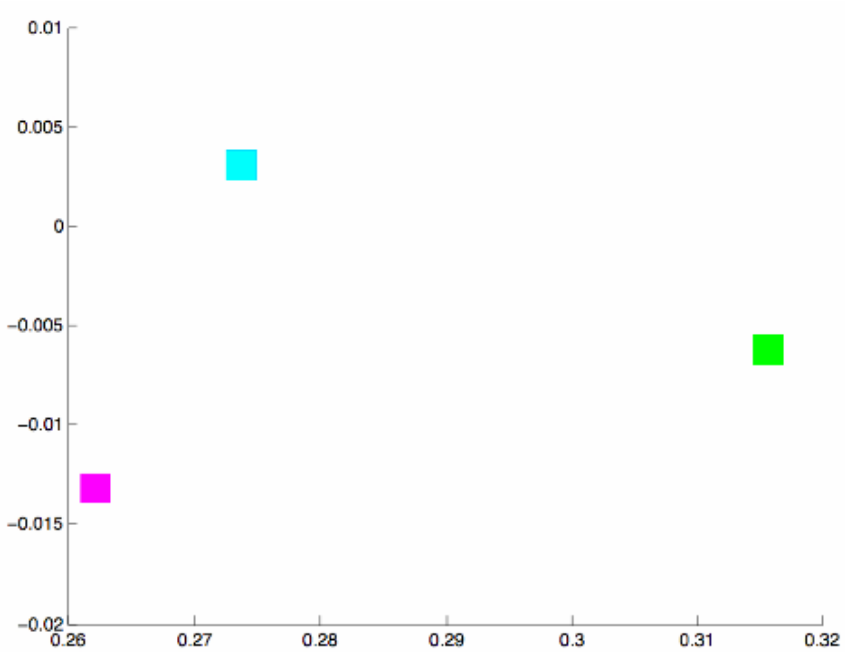


Close up of first node

Close up of Embeddings



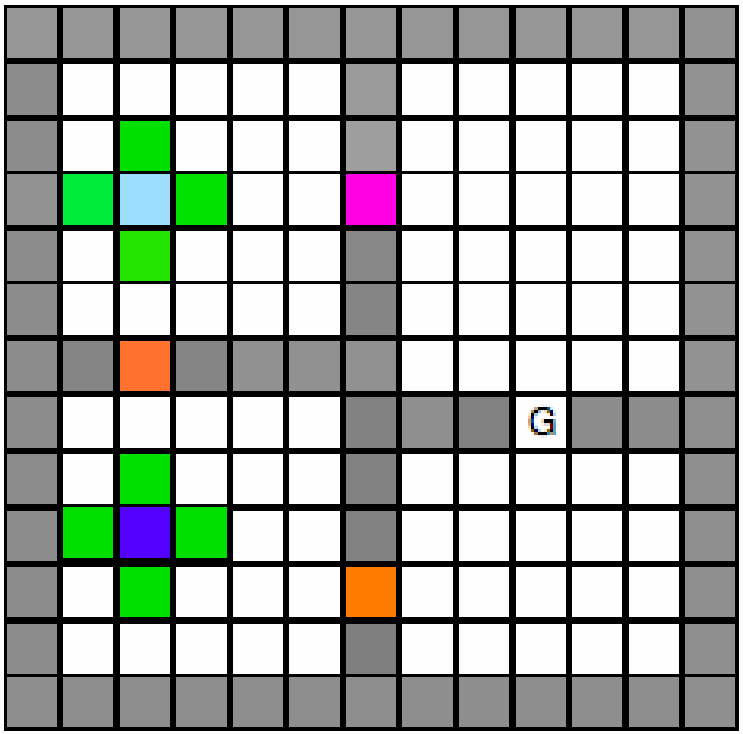
State-option Graph Embedding



State Graph Embedding

Four Room Gridworld

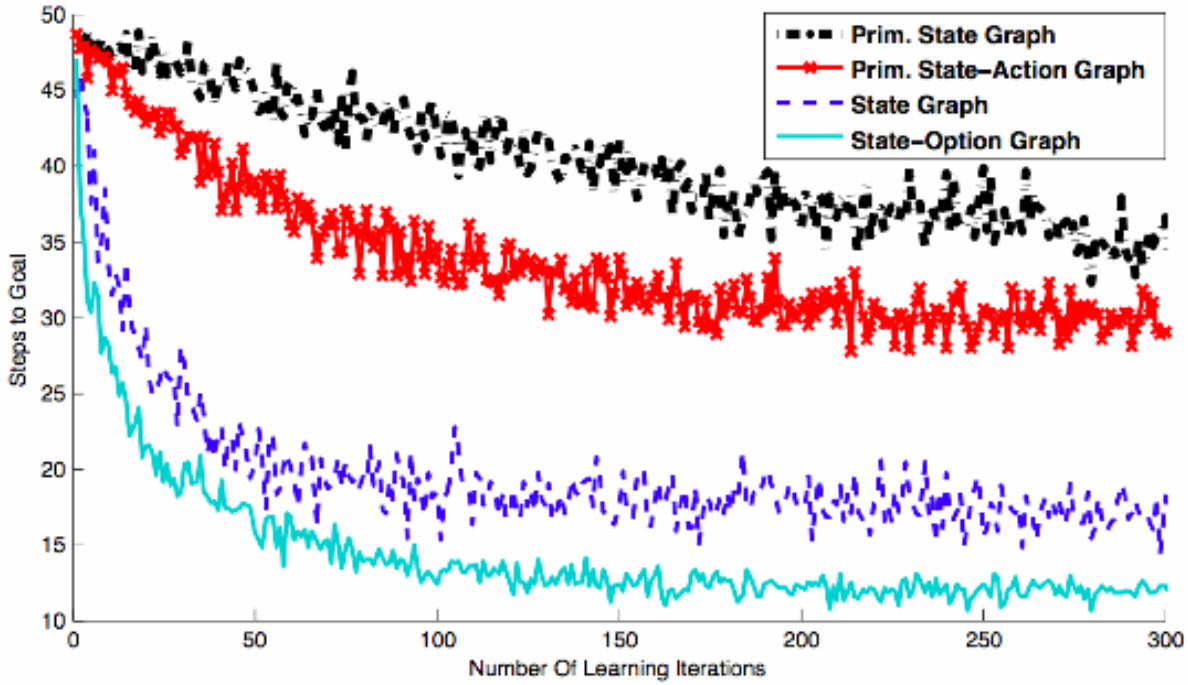
(Sutton et al., AIJ, 1999)



- 104 states
- 4 primitive actions (N,E,S,W)
- Two hallway options per room
- Total number of actions: 12
- Stochastic environment: 10% probability an action will fail

Results on Four Room Gridworld

(Osentoski and Mahadevan, ICML 2007)



Prim State Graph: 400 basis functions

Prim State-action Graph: 260 basis functions

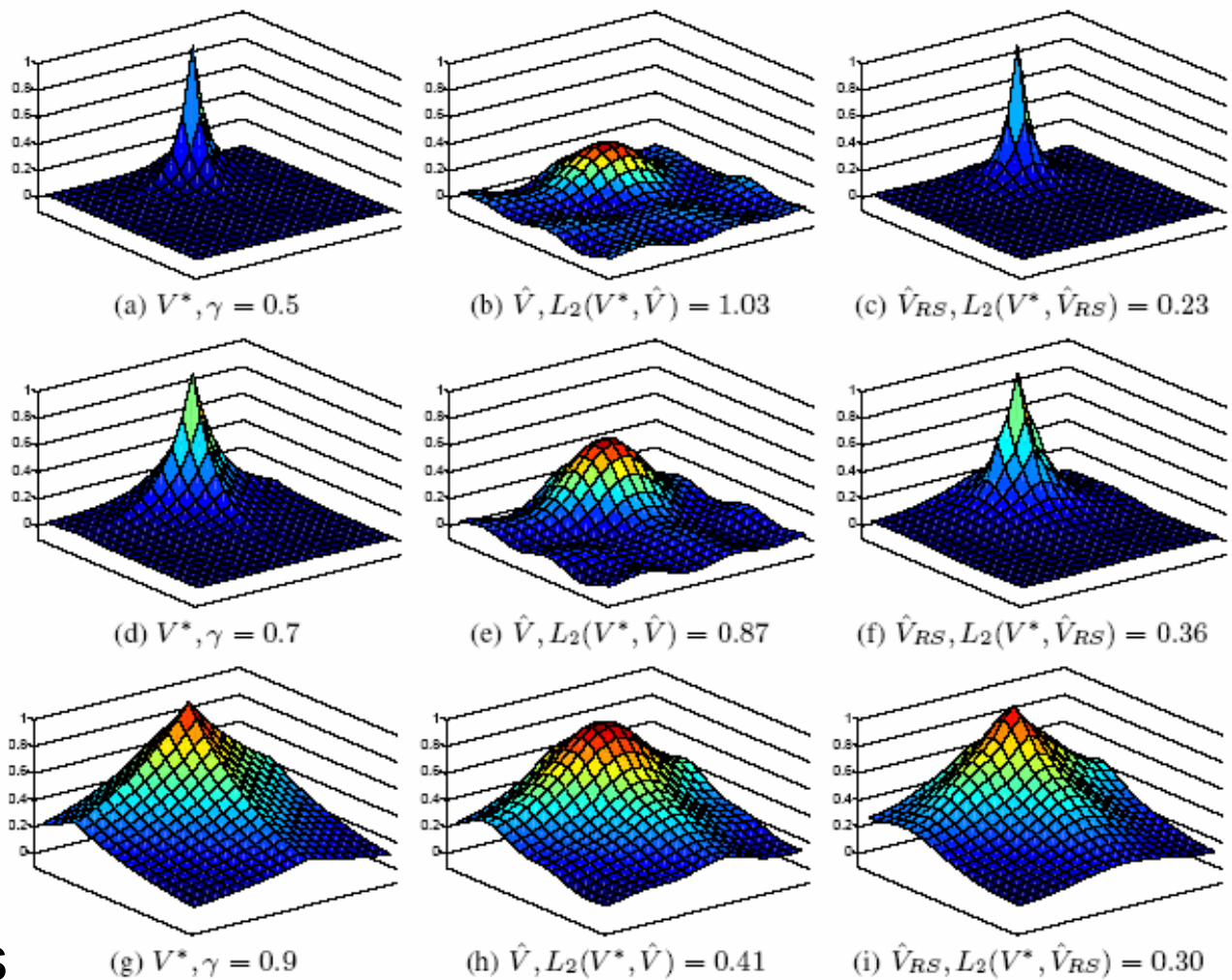
State graph: 264 basis functions

State-option graph: 260 basis functions

Reward Sensitive Basis Functions

(Johns, 2007; Parr et al., ICML 2007; Keller, ICML 2006)

- The graph weight matrix W can reflect the reward function
- This results in bases tuned to a value function
- If the target function is unknown, it is harder to do this tuning



Exact

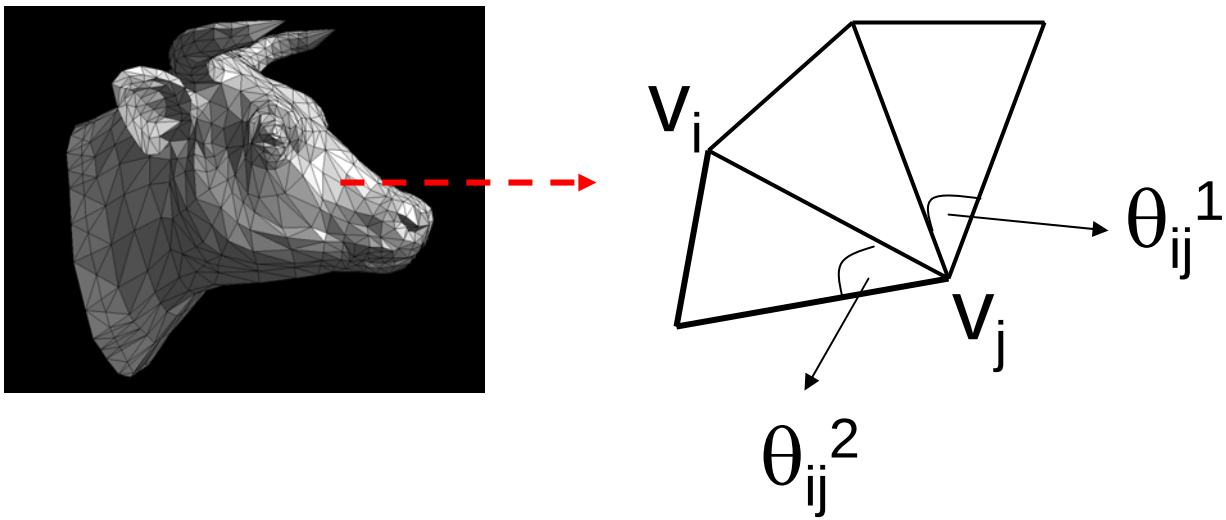
Topological approximation

Reward-sensitive approximation

Mean-Value Representation of Laplacian over 2-Manifolds

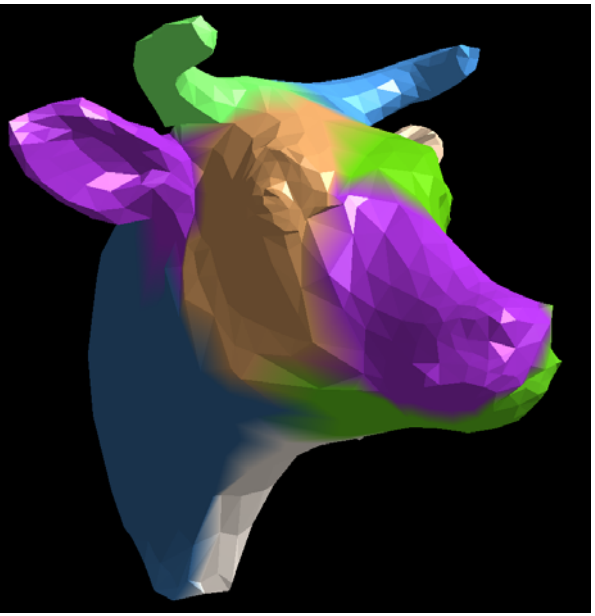
(Floater et al., 2003)

- 3D meshes are triangulations of 2-manifolds (surfaces)
- The weight matrix W can be specified for triangulated meshes based on geometry



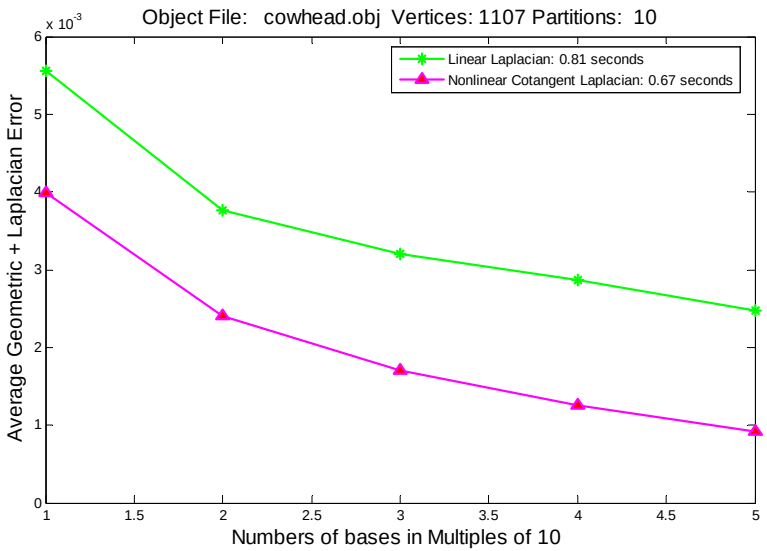
$$W_{ij} = [\tan(\theta_{ij}^1/2) + \tan(\theta_{ij}^2/2)]/(|| v_i - v_j ||)$$

Topological vs. Geometry-Aware Laplacian Bases



Colors indicate subgraphs

Topological bases



Geometry-aware bases

Spectral Clustering

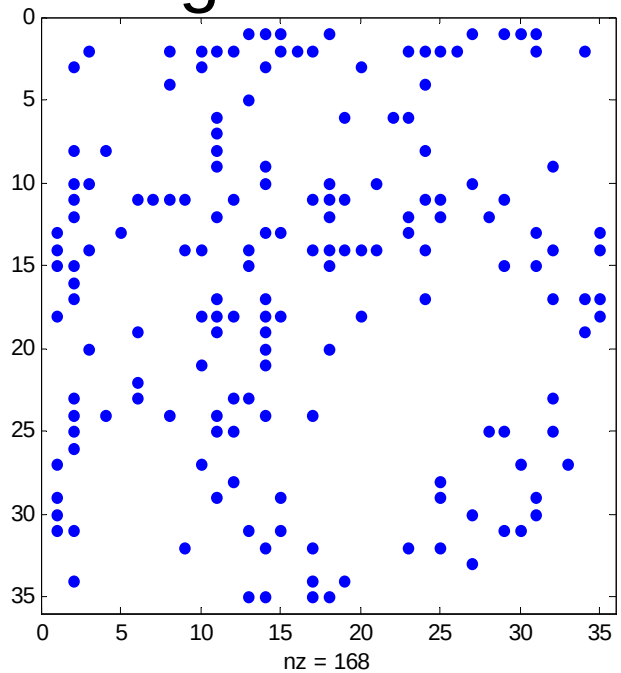
(Ng, Jordan,Weiss, NIPS 2001)

- Compute the normalized graph Laplacian defined as

$$\mathcal{L} = D^{-1/2} (D-W) D^{-1/2}$$

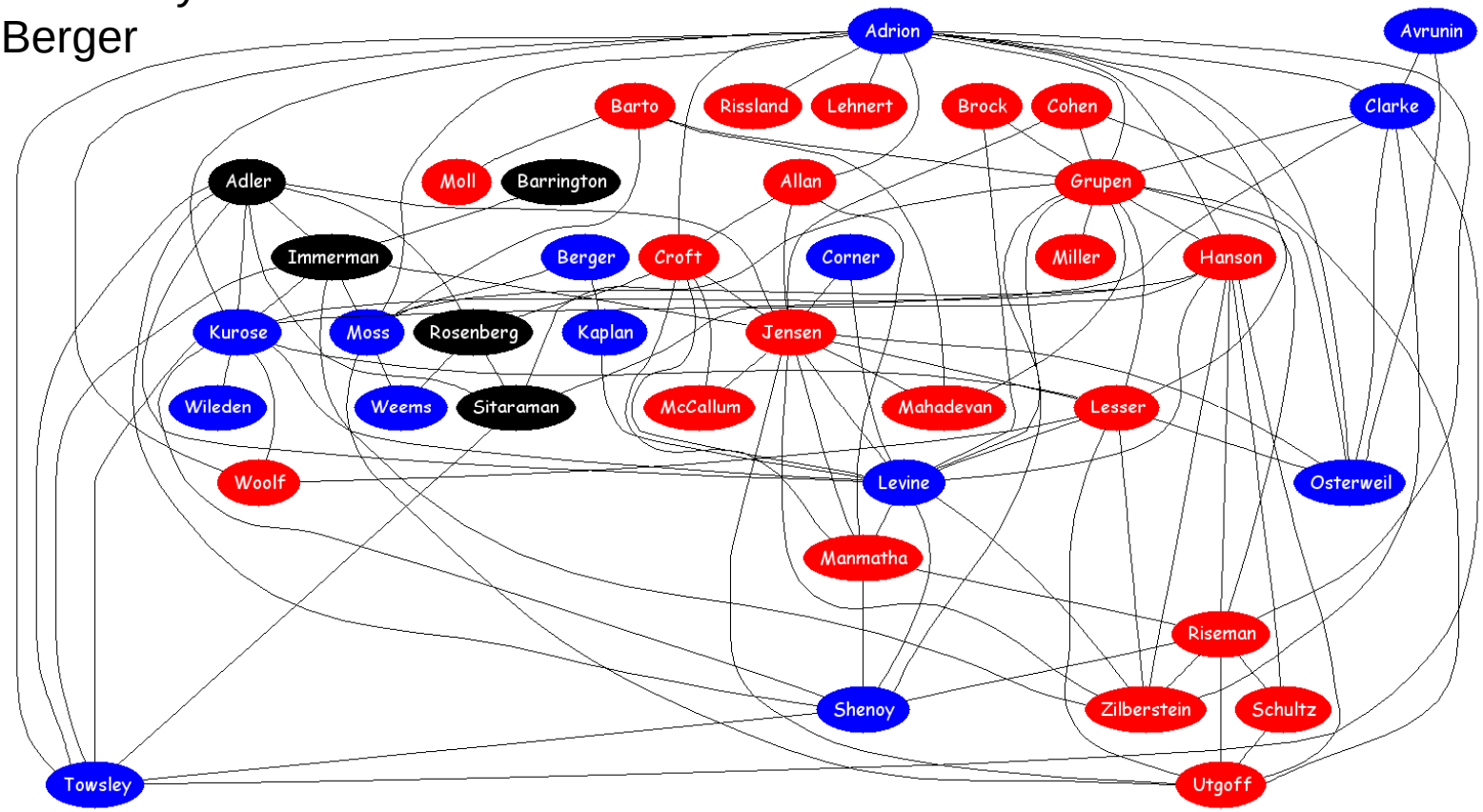
- $\mathcal{L}$ is the discrete version of the Laplace-Beltrami operator on a Riemannian manifold
- Project the data onto the low-order eigenvectors of $\mathcal{L}$
- Use k-means method on projected data

Weight matrix W



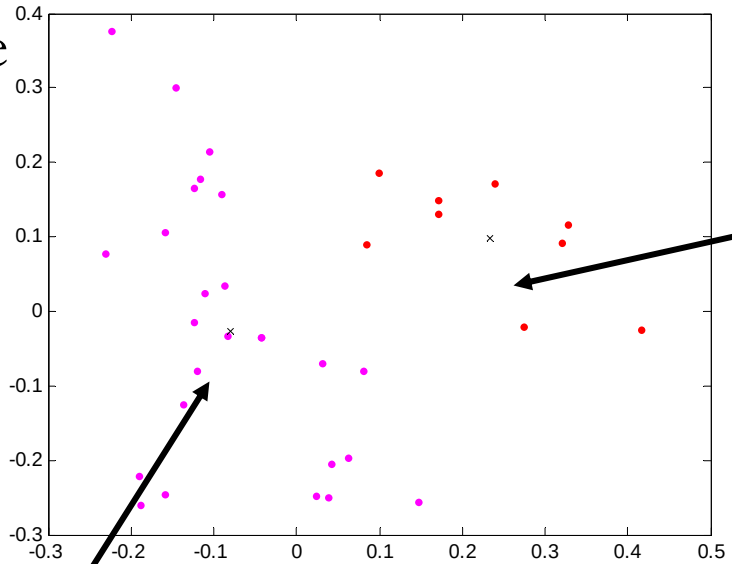
Faculty Collaboration Graph
(U.Mass Amherst CS Dept)

Graph formed by
Emery Berger



Spectral Clustering using Graph Laplacian

Embedding using the 2nd and 3rd eigenvector of the graph Laplacian



"Theory And Networks"

Cluster: 1
Adler
Barrington
Immerman
Kurose
Rosenberg
Shenoy
Sitaraman
Towsley
Weems

cluster: 2
Adrion Allan Avrunin Barto Brock Clarke Cohen Croft Grupen Hanson Jensen Lehnert Lesser Levine Mahadevan Manmatha McCallum Moll Moss Osterweil Riseman Rissland Schultz Utgoff Woolf Zilberstein

"AI & Systems"

Limitation of Fourier Methods

- “Global” representation that captures long-time scales
  - Basis functions span the size of the graph
  - Eigenvectors reflect long-term regularities
- Difficult to approximate functions that are not globally smooth
  - Local discontinuities cause global “ripples”
  - Regularities at different spatial/temporal frequencies cannot be modeled easily
- After the break, we will explore multi-scale manifold methods
  - Unlike global eigenvector methods, these are based on constructing compact multiscale basis functions

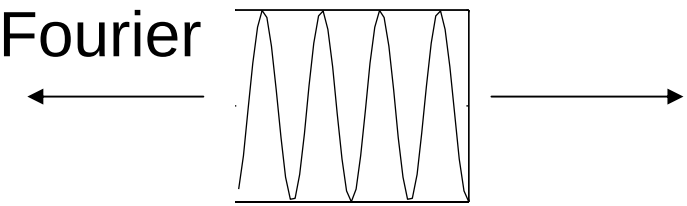
Structure of Tutorial

PART 1	Motivation: Why automate representation discovery?
PART II	Representation Discovery using Fourier Manifold Learning
	COFFEE BREAK
PART III	*Multiscale* Representation Discovery using Wavelet Manifold Learning
PART IV	Advanced Topics and Challenges; Discussion

Wavelet Representations

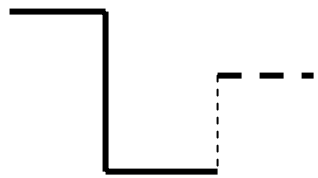
- Wavelets are a multiscale framework originally developed for Euclidean spaces [Haar, 1900; Daubechies, 1992; Mallat 1999]
  - Basis functions are *compact* and at *multiple scales*
  - Higher level basis functions are constructed by *dilations*
- Wavelets on graphs and manifolds [Coifman and Maggioni: ACHA 2006; Mahadevan and Maggioni, NIPS 2005, ICML 2006]
  - Extends classical wavelets to graphs and manifolds
  - Multi-resolution analysis of functions on graphs
  - Compact bases

Fourier



Localized in frequency
Stretched in time

Wavelet



Localized in frequency and time

Multiscale Analysis on Graphs

- Let T be a diffusion operator on a graph
  - Example: $T = D^{-1/2} W D^{-1/2}$
- Assume T is self-adjoint (symmetric)
  - We will relax this assumption soon
- Let the spectrum of T be normalized
  - $\lambda_i \in [0,1]$
- Denote the desired resolution by ε

Definition of Multiresolution

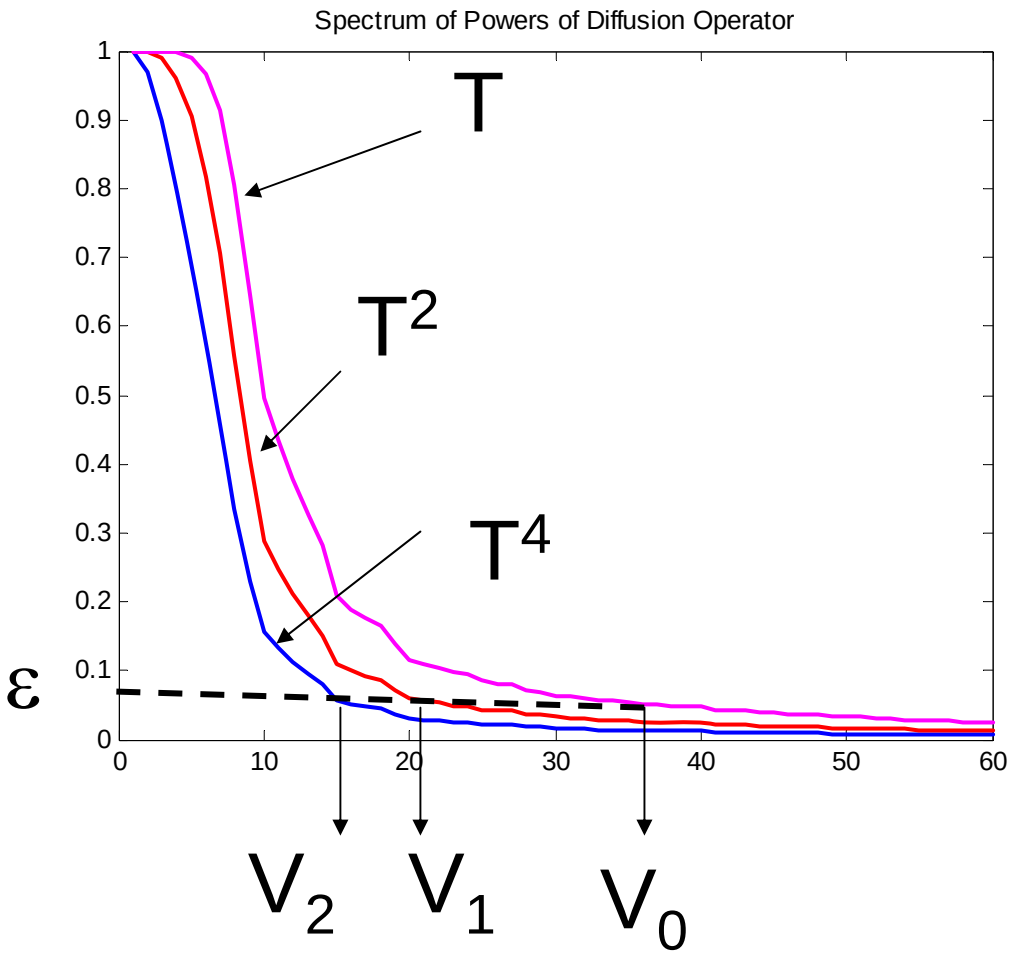
- Dyadic powers
  - $t_j = 2^j - 1$
- Multiscale spectral analysis

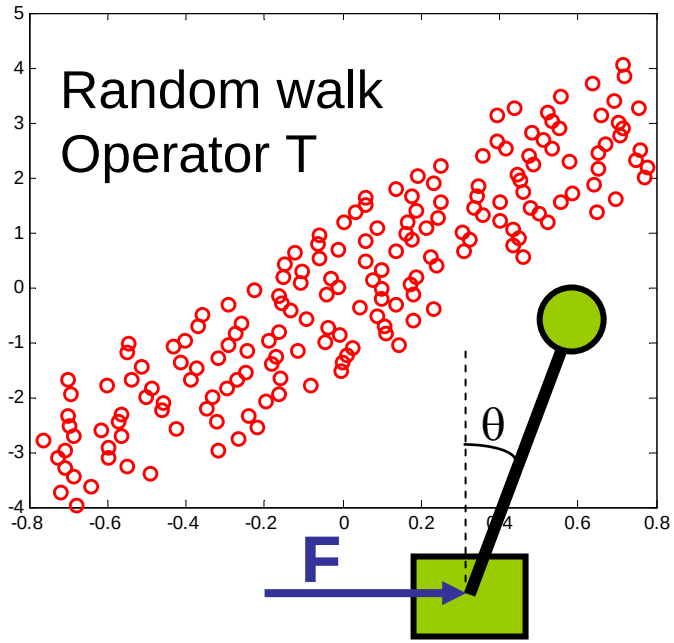
$$\sigma_j = \{\lambda \in \sigma(T) \mid \lambda^{t_j} \geq \varepsilon \}$$

- Vector space hierarchy

$$V_j = \text{span}(\{\xi_j : \lambda_j \in \sigma_j(T)\})$$

Compressing Powers of Diffusion Operator

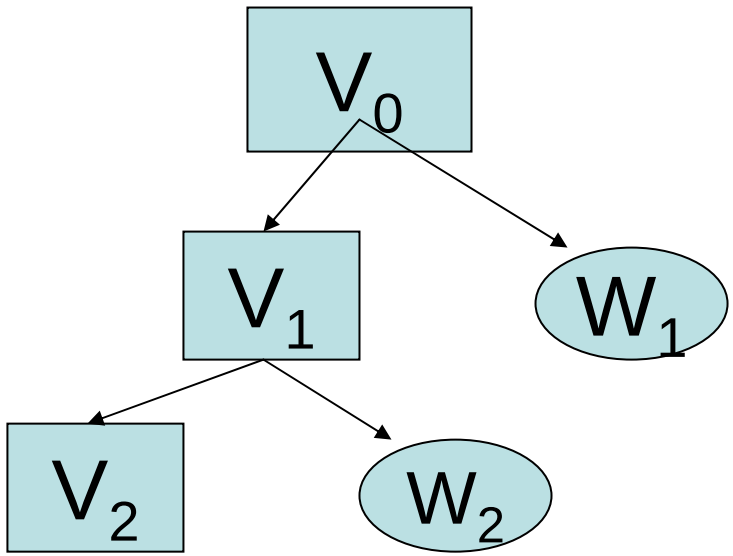




Samples from random walk on Inverted pendulum task

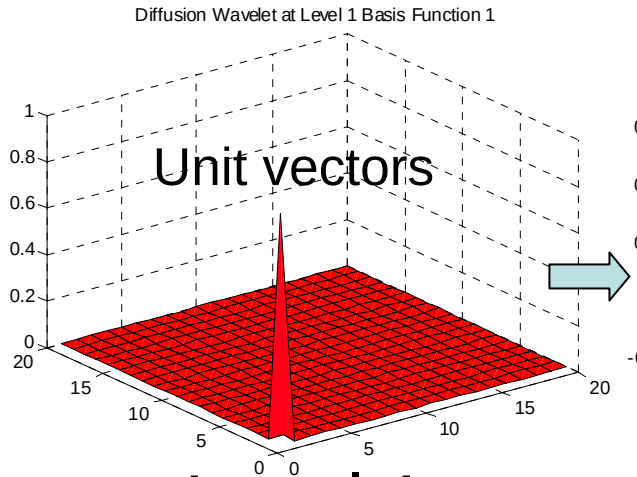
Scaling Functions and Wavelets

- The vector spaces V_j are "low-pass" filters
  - They are spanned by "scaling function" bases
- W_j: a series of subspaces orthogonal to V_j
  - These are "high-pass" filters, and capture the resolution lost in going from V_j to V_{j+1}
  - The W_j are spanned by "wavelet" bases

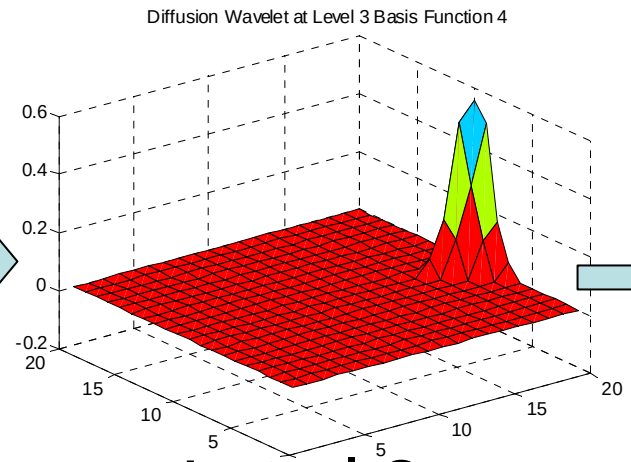


$$V_0 = V_1 \oplus W_1$$

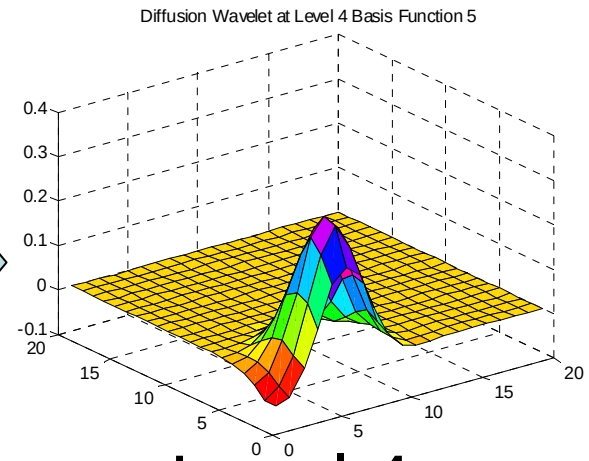
Scaling Function Bases on 2D Graph with Bottleneck



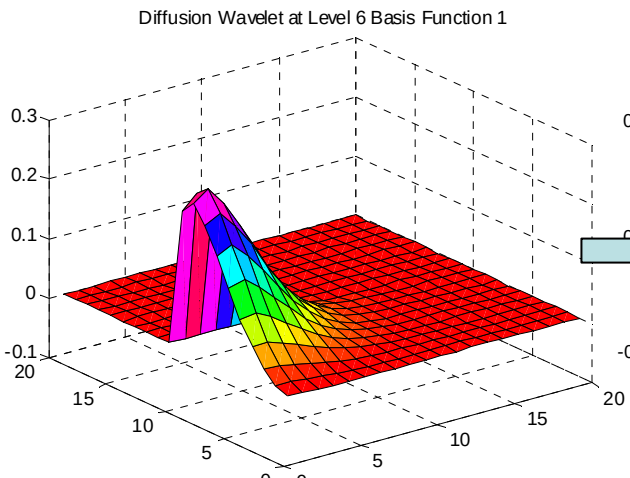
Level 1



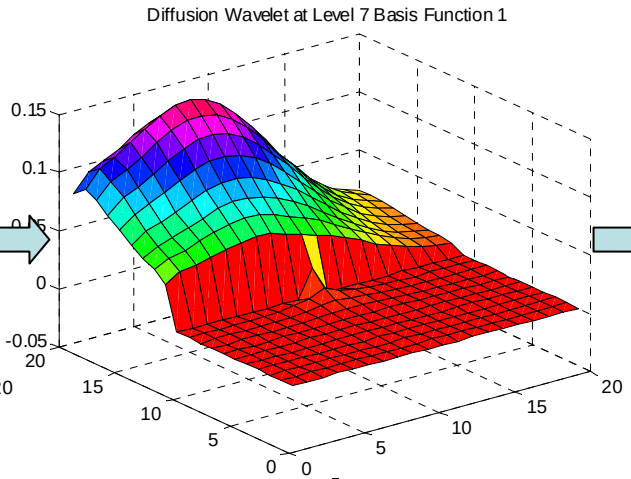
Level 3



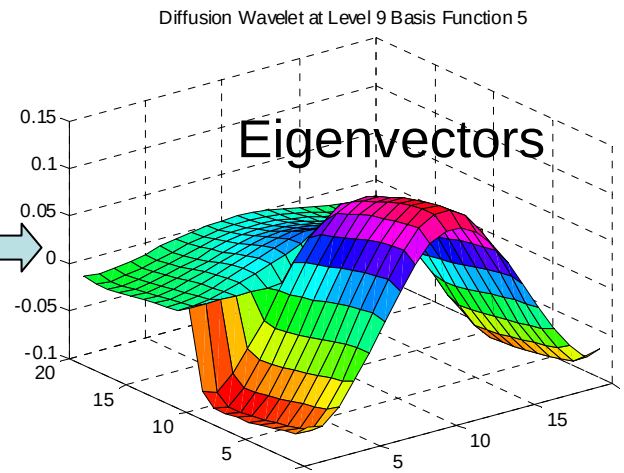
Level 4



Level 6



Level 7

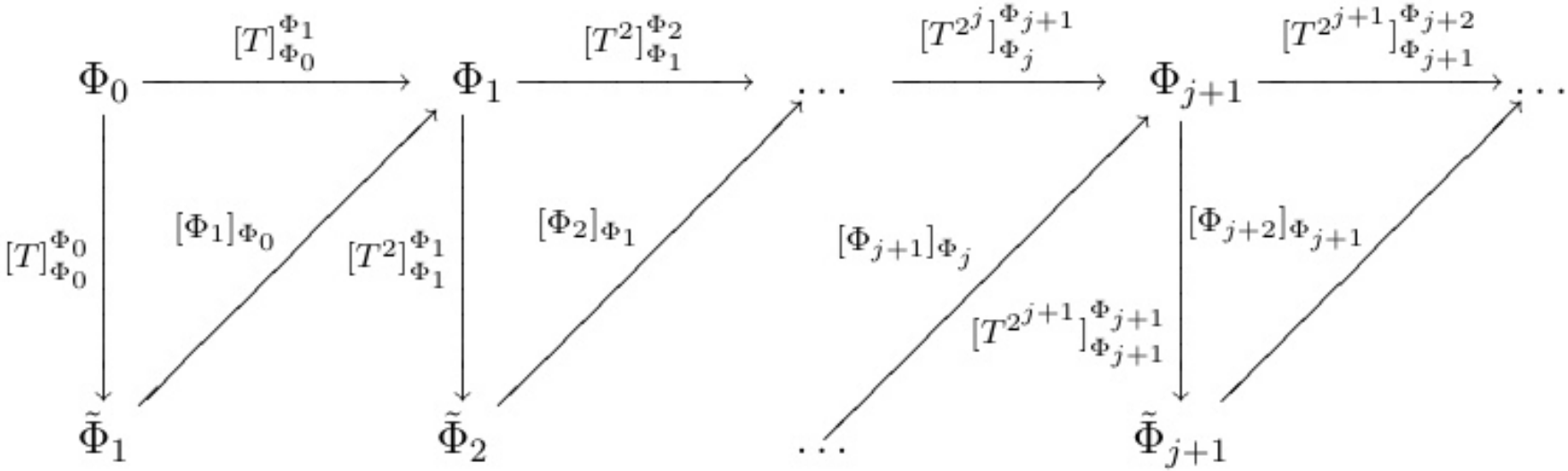


Level 9

Diffusion Wavelet Construction

Dilation

Downsampling



All triangles commute by construction

Multiscale Construction

- Orthogonalization and downsampling

$$[\Phi_{j+1}]_{\Phi_j}, [T^{2^j}]_{\Phi_j}^{\Phi_{j+1}} \leftarrow \mathtt{QR}([T^{2^j}]_{\Phi_j}^{\Phi_j}, \epsilon)$$

- Dilation and operator compression (non-symmetric case)

$$[T^{2^{j+1}}]_{\Phi_{j+1}}^{\Phi_{j+1}} = [\Phi_{j+1}]_{\Phi_j}([T^{2^j}]_{\Phi_j}^{\Phi_j})^2([\Phi_{j+1}]_{\Phi_j})^T$$

Multiscale Construction

- Orthogonalization and downsampling

$$[\Phi_{j+1}]_{\Phi_j}, [T^{2^j}]_{\Phi_j}^{\Phi_{j+1}} \leftarrow \mathtt{QR}([T^{2^j}]_{\Phi_j}^{\Phi_j}, \epsilon)$$

- Dilation and operator compression (symmetric case)

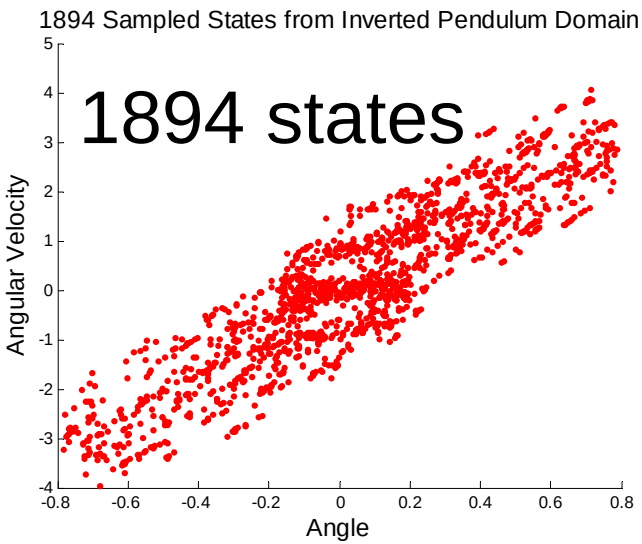
$$[T^{2^{j+1}}]_{\Phi_{j+1}}^{\Phi_{j+1}} = [T^{2^j}]_{\Phi_j}^{\Phi_{j+1}} ([T^{2^j}]_{\Phi_j}^{\Phi_{j+1}})^T$$

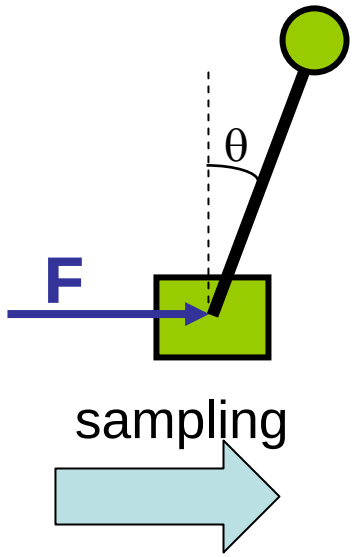
Wavelet Bases Construction

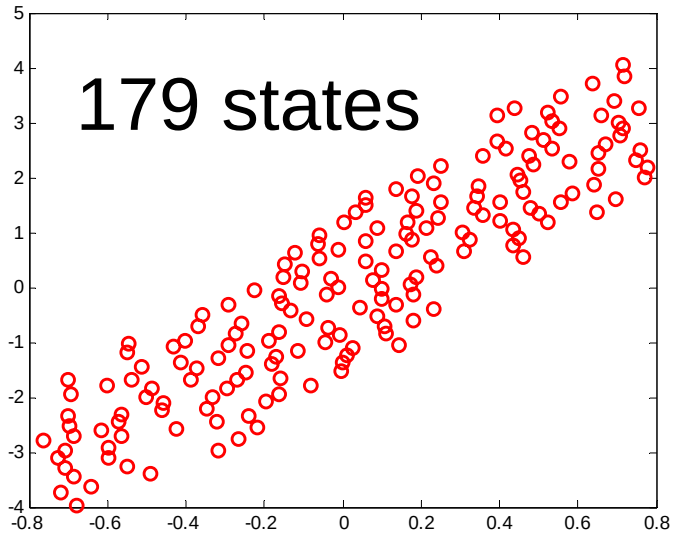
- Find basis functions orthogonal to scaling functions:

$$[\Psi_j]_{\Phi_j} = \mathtt{QR}(I_{<\Phi_j>} - [\Phi_{j+1}]_{\Phi_j}([\Phi_{j+1}]_{\Phi_j})^T, \epsilon)$$

Diffusion Wavelet Bases on Inverted Pendulum

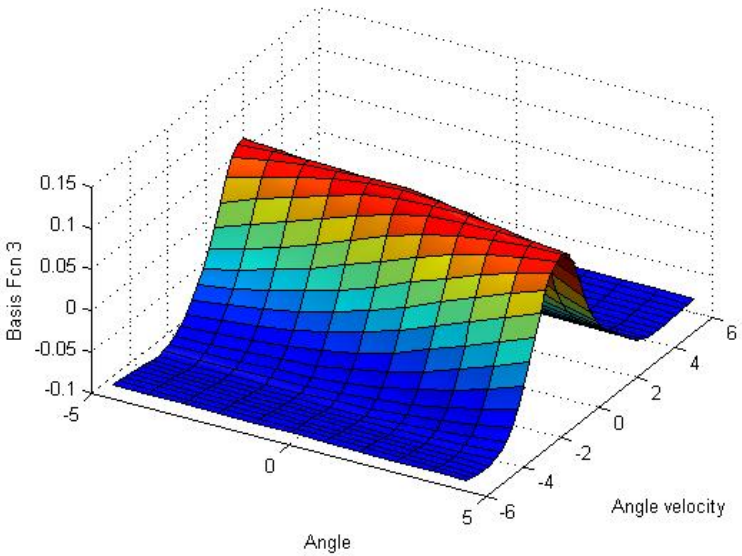


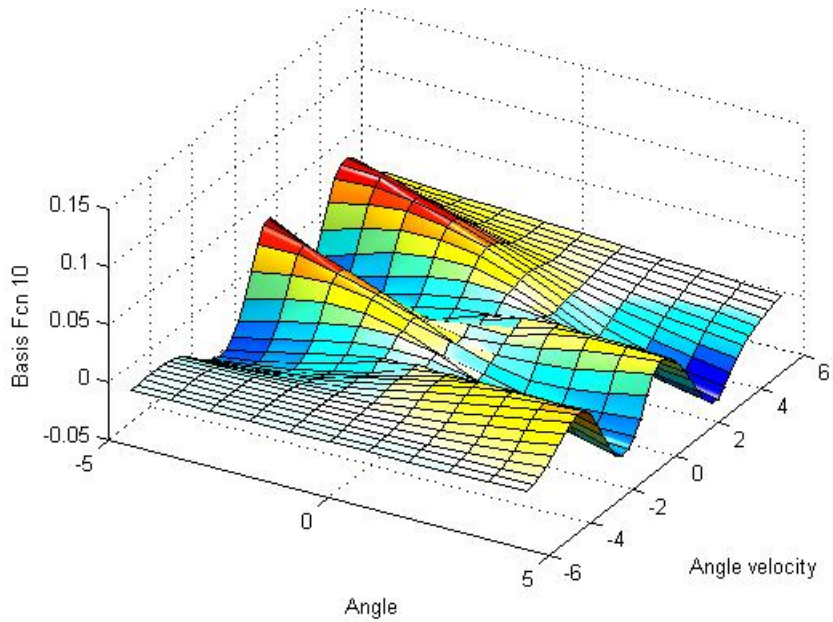


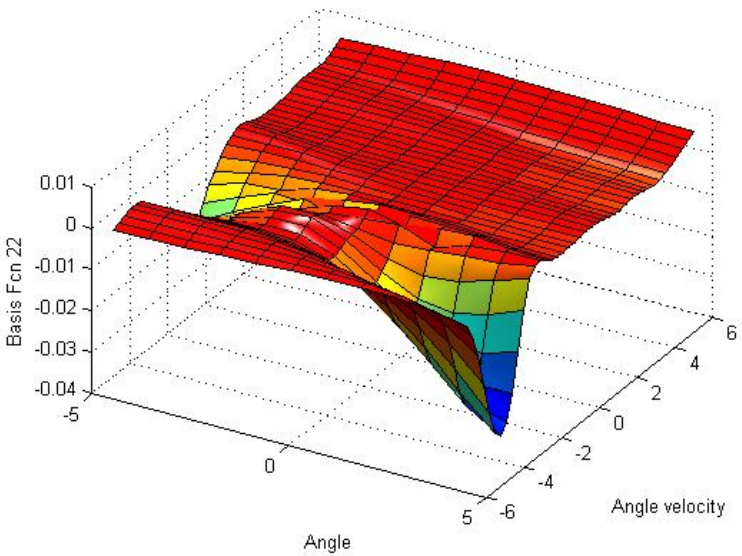


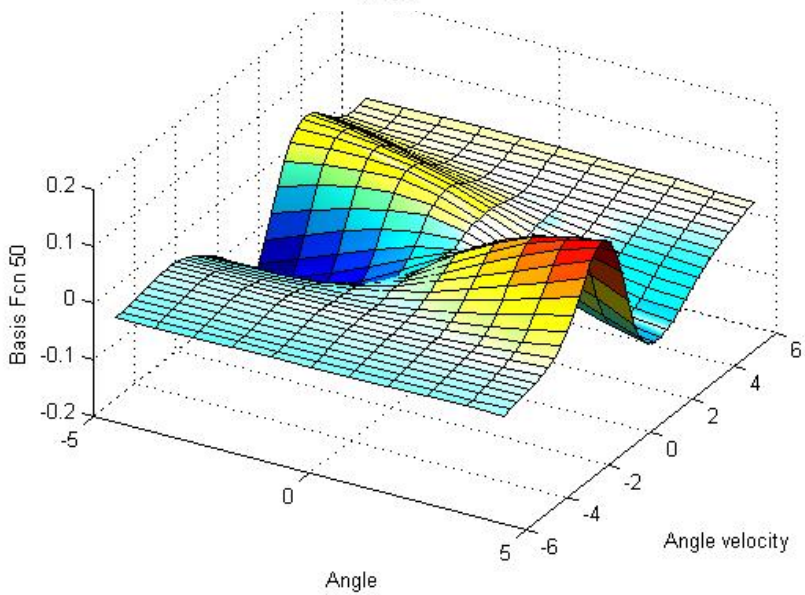

```
Constructing diffusion wavelet tree on L^4 on 179 points...
V_1          gsqr:   22 fcns, 0.09 secs     T reps: T^2: 0.00 secs     freq: [1e-006 1]
    W_1          gsqr:  157 fcns, 0.95 secs                freq: [0 1e-006]
V_2          gsqr:    9 fcns, 0.03 secs     T reps: T^2: 0.00 secs     freq: [0.001 1]
    W_2          gsqr:   13 fcns, 0.00 secs                freq: [1e-006 0.001]
V_3          gsqr:    6 fcns, 0.02 secs     T reps: T^2: 0.00 secs     freq: [0.0316228 1]
    W_3          gsqr:    3 fcns, 0.00 secs                freq: [0.001 0.0316228]
V_4          gsqr:    4 fcns, 0.00 secs     T reps: T^2: 0.00 secs     freq: [0.177828 1]
    W_4          gsqr:    2 fcns, 0.00 secs                freq: [0.0316228 0.177828]
V_5          gsqr:    3 fcns, 0.00 secs     T reps: T^2: 0.00 secs     freq: [0.421697 1]
    W_5          gsqr:    1 fcns, 0.00 secs                freq: [0.177828 0.421697]
```

Pendulum Diffusion Bases

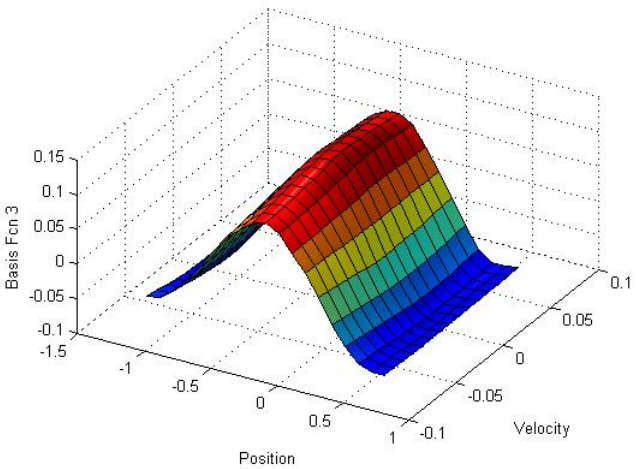


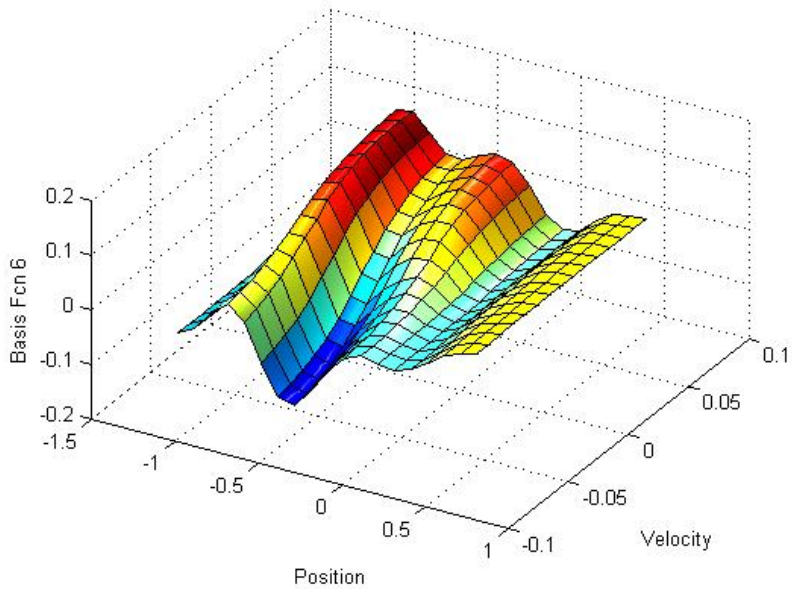


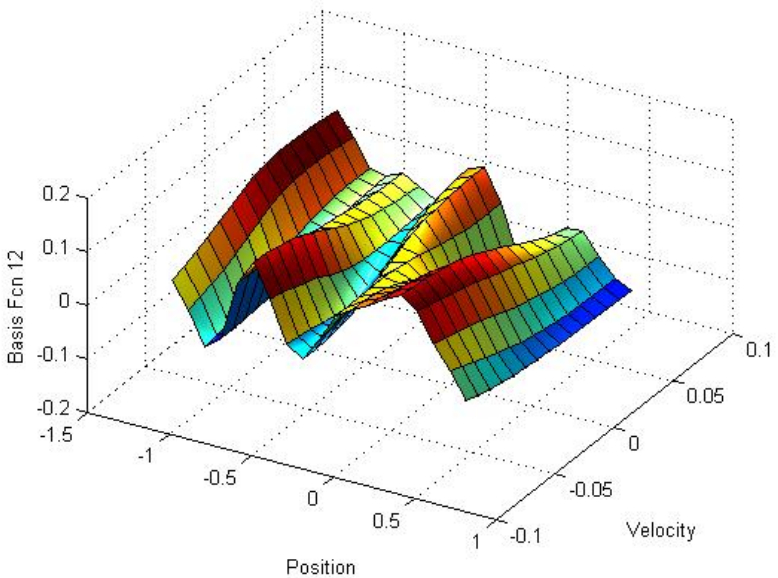


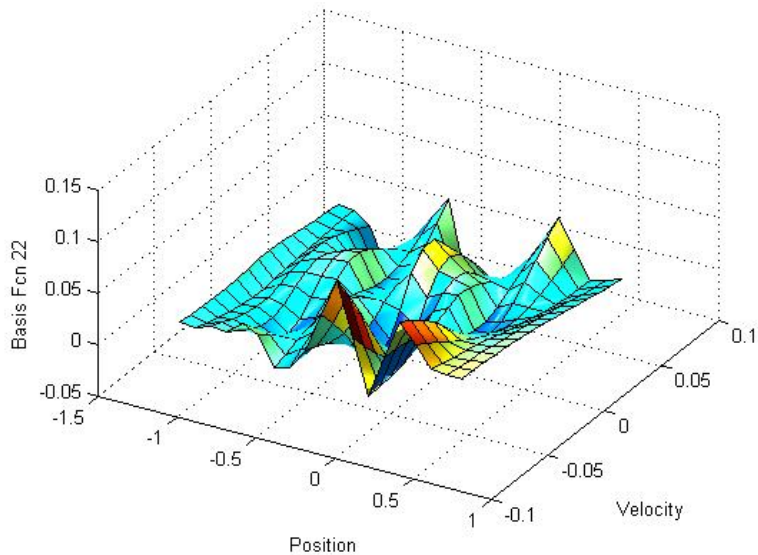


Mountain Car Scaling Functions







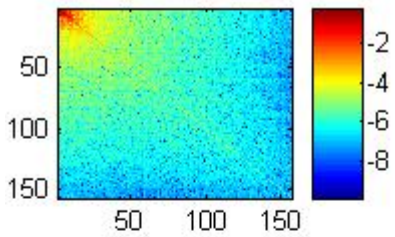


Diffusion Operator Compression

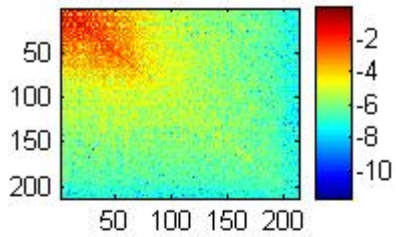
Pendulum

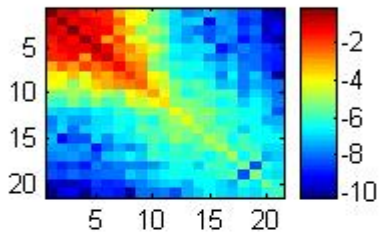
Plots in $\log_{10}$ scale

Mountain Car

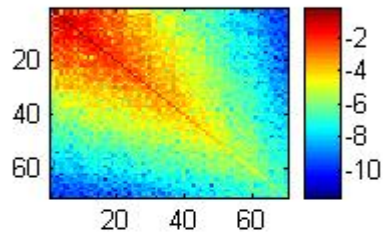


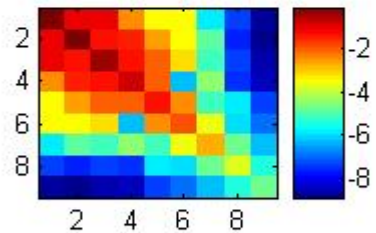
T



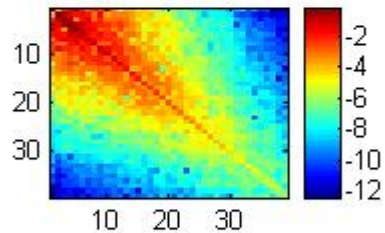


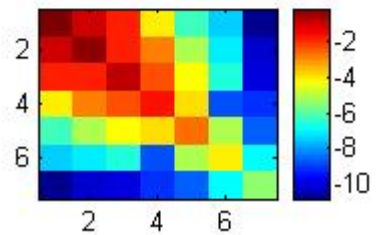
T^2



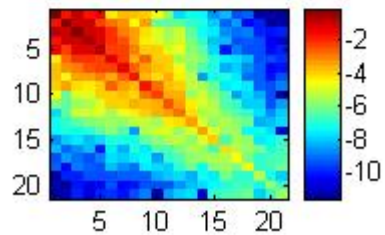


T^4



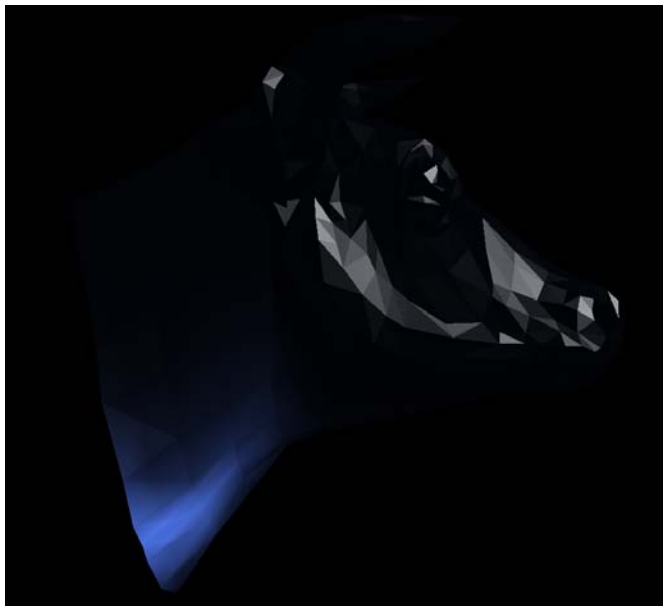


T^8



3D Mesh Compression using Diffusion Wavelets

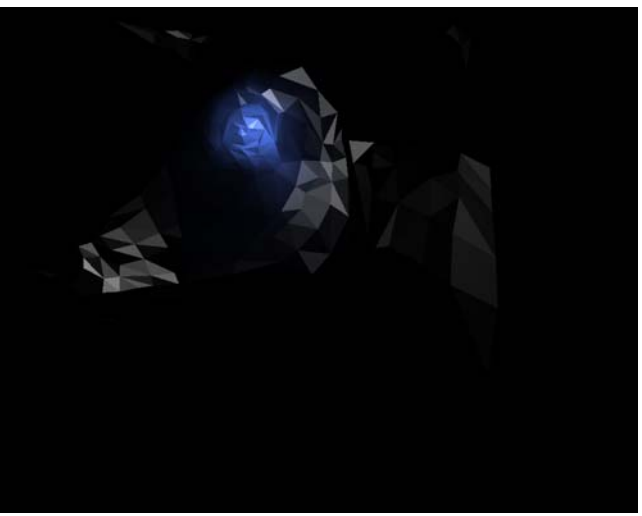
(Mahadevan, ICML 2007)



Level 4

Level 9

Level 5



Level 8

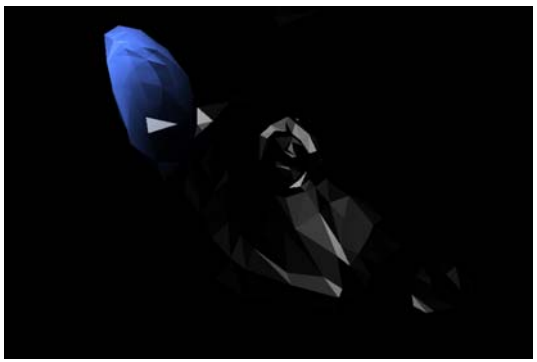
Level 10

Level 5 Basis Functions

“Horn”

“Eye”

“Chin”



“Ear”

“Cheek”

“Neck”

Diffusion Projection: Multi-scale Analysis on Directed Manifolds

(Chang, Maggioni and Mahadevan, 2007)

- ***Diffusion Projection (DP)*** is based on the diffusion scaling functions in Diffusion Wavelets.
- Diffusion Projections provide multi-scale embedding, which means they automatically reveal the geometric structure of the data at different scales
- They enable finding the best scale/dimension for a low dimensional embedding once a rough dimension range is given.

Diffusion Projections

- The scaling function $[\phi_j]_{\phi_0}$ provides a mapping between the data on large scale space and small scale space.

- The elements in $[\phi_j]_{\phi_0}$ are usually much coarser and smoother than the initial elements in $[\phi_0]_{\phi_0}$, which is why they can be represented in a compressed form.

- Given $[\phi_j]_{\phi_0}$, any function on the compressed large-scale space can be extended naturally to the original space or vice versa. The connection between any vector in the original space and its compressed representation at scale *j* is $v_{[\phi_j]} = ([\phi_j]_{\phi_0})' v_{[\phi_0]}$

Symmetric and Non-symmetric Embeddings

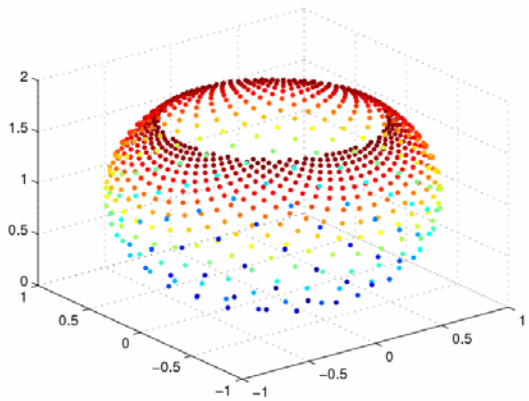
- Diffusion Projection can handle both symmetric and non-symmetric matrices.

- Symmetric case: the low dimensional embedding is the same as the embedding learned from Laplacian eigenmap, up to precision ε.

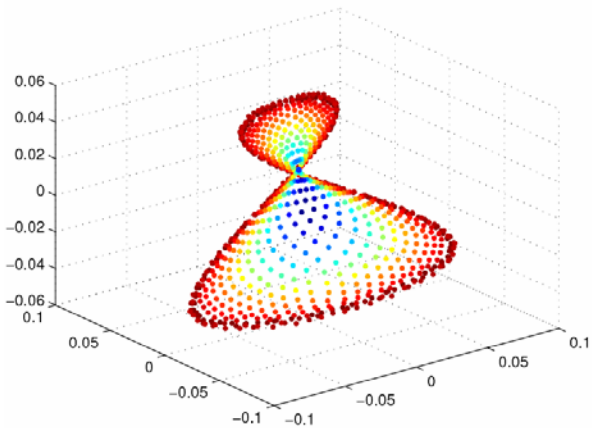
- Non-symmetric case spans the same subspace, up to a precision ε, spanned by the columns of $[\phi_t]_{\phi_0}$, which is the space of probability distributions of the random walk T^{2^t} at time 2^t.

Example

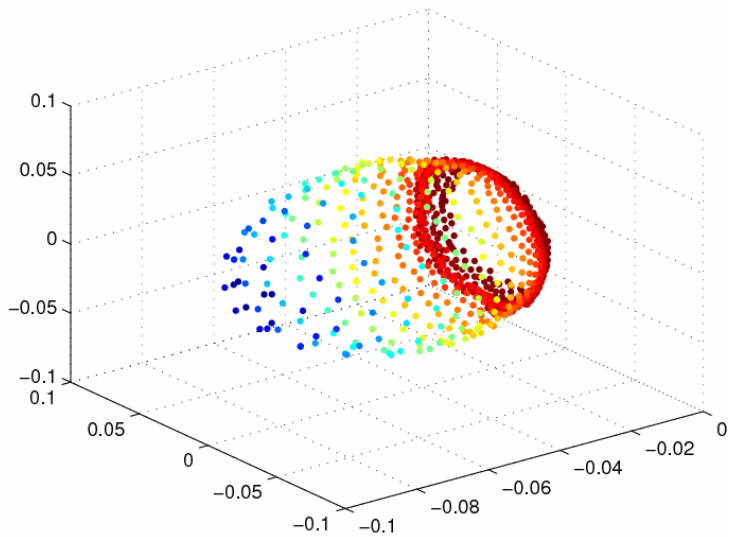
The punctured sphere in Figure (A) includes 800 points. We compare Laplacian eigenmap on *(W+W')/2* with Diffusion Projections on the random walk matrix of *W*.



Original



Laplacian embedding



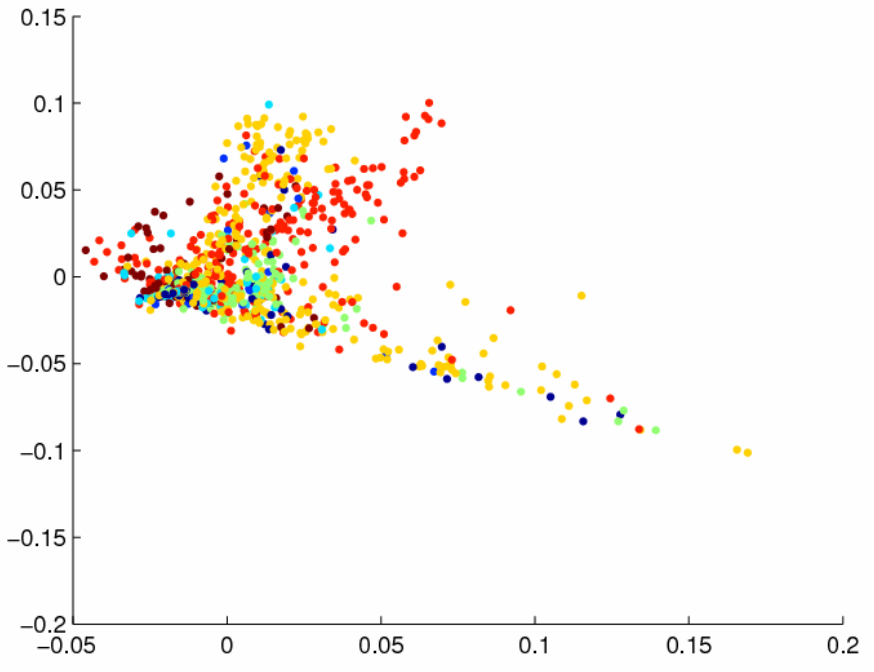
Diffusion Projection embedding

Multiscale Embedding of Citation Graphs

- The citation data set in KDD Cup 2003 are scientific papers from arXiv.org from the high-energy physics theory area.
- We sampled 1716 documents from the complete data set and created a citation graph.
- A citation relationship is directed, and a paper cited by many other papers should be more important compared to a paper that cites many others but is not cited by others.
- Symmetrizing such a relationship destroys much useful information.

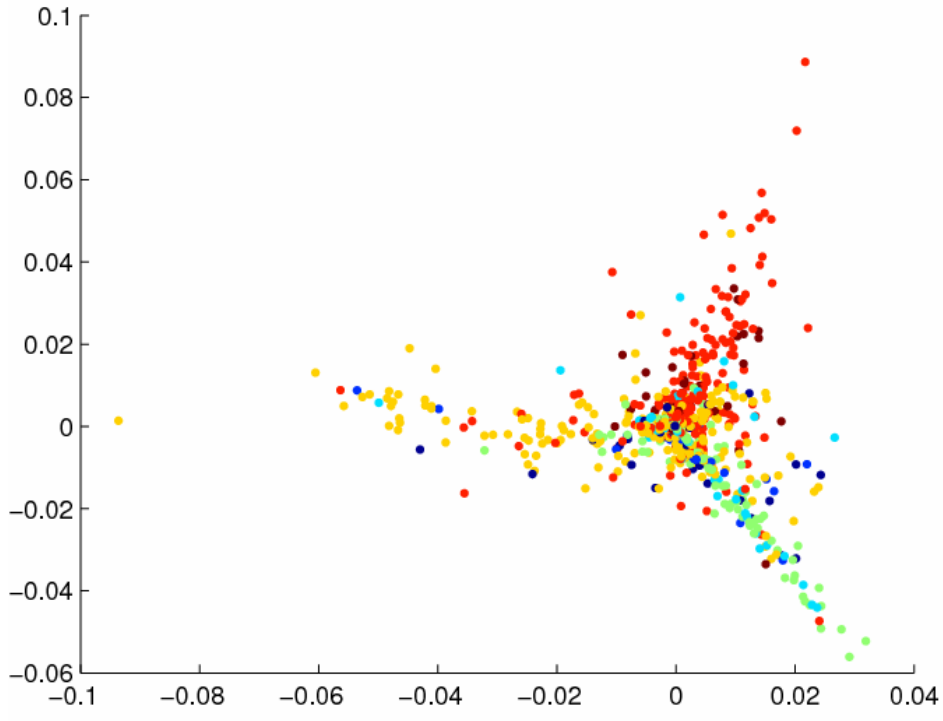
Citation Graphs: Two Embeddings

(High-energy physics papers, KDD 2003)



Laplacian Eigenmaps on Undirected Graph

(Belkin and Niyogi, MLJ 2005)



Diffusion Projection on Directed Graph

(Wang et al., 2007)

Diffusion Policy Evaluation: Fast Inversion

(Maggioni and Mahadevan, ICML 2006)

$$V^{\pi} = (I - \gamma P^{\pi})^{-1} R^{\pi} = (I + \gamma P^{\pi} + \gamma^2 (P^{\pi})^2 + \ldots) R^{\pi}$$

- Two approaches to policy evaluation:
  - DIRECT: inverting the matrix takes $O(|S|^3)$
  - ITERATIVE: successive approximation in $O(|S|^2)$
- New faster approach to policy evaluation:
  - Construct a diffusion wavelet tree from the transition matrix to invert the Green's function
  - Use the Schultz expansion to do the inversion
  - Results in a significantly faster method $\approx O(|S|)$ (for *diffusion* matrices, within a log |S| factor)

Schultz Expansion for Diffusion Semi-Groups

- The random walk operator T is a semi-group, where the powers T^k satisfy the following conditions
  - $T^0 = I$
  - $T^{k+l} = T^k T^l$
- To compute the Green's function $(I - T)^{-1}$, we use the Schultz expansion formula
- As a special case, we can apply this approach to policy evaluation, where T is represented by γP

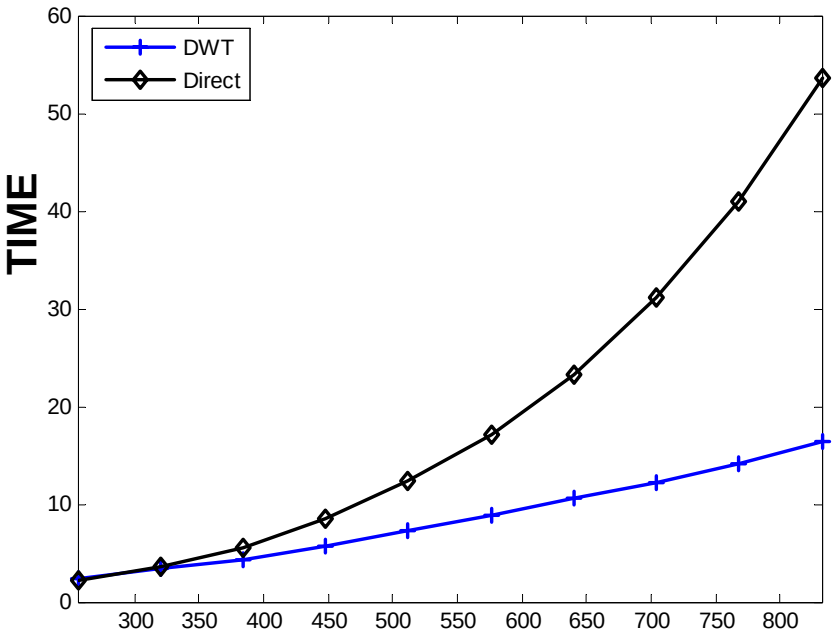
$$(I-T)^{-1} f = \sum_k T^k f$$

$$S_k = \sum_{k=1}^{2^k} T^k$$

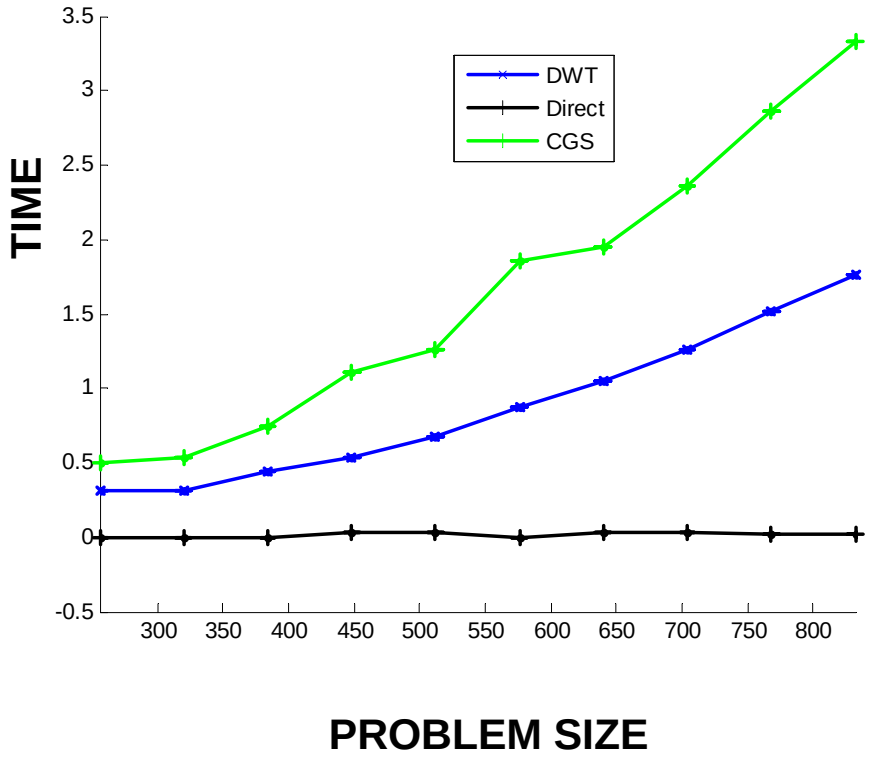
$$S_{k+1} = S_k + T^{2^k} S_k = \prod_{l=0}^{k} (I + T^{2^l})$$

Policy Evaluation Results

PRECOMPUTATION



VALUE DETERMINATION



Structure of Tutorial

PART 1	Motivation: Why automate representation discovery?
PART II	Representation Discovery using Fourier Manifold Learning
	COFFEE BREAK
PART III	*Multiscale* Representation Discovery using Wavelet Manifold Learning
PART IV	Advanced Topics and Challenges; Discussion

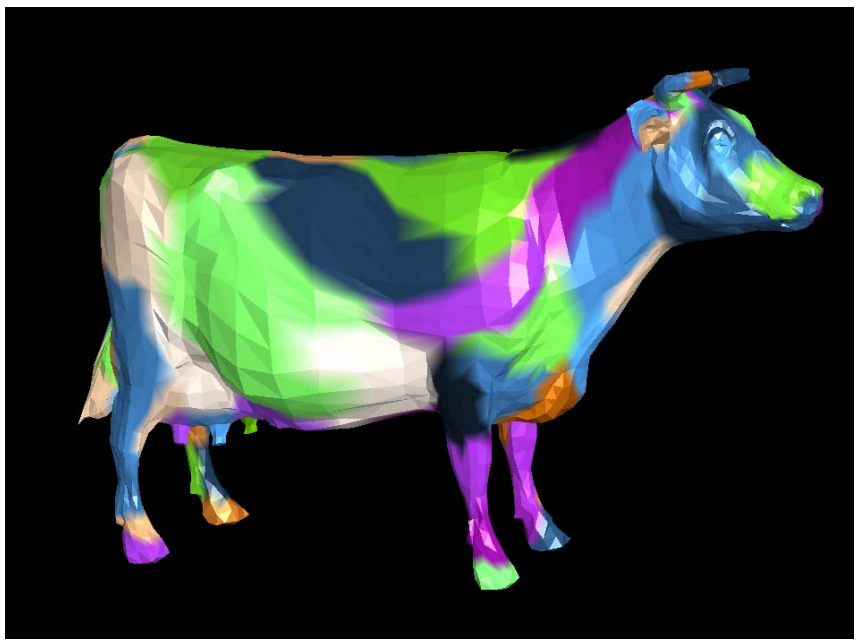
Representation Discovery in Large Spaces

- Divide-and-conquer:
  - Graph partitioning
- Factorization of product spaces:
  - It is possible to represent spectral bases compactly for large factored state spaces
- Kronecker Decomposition
  - Approximate matrix factorization using Kronecker products
- Harmonic analysis on groups

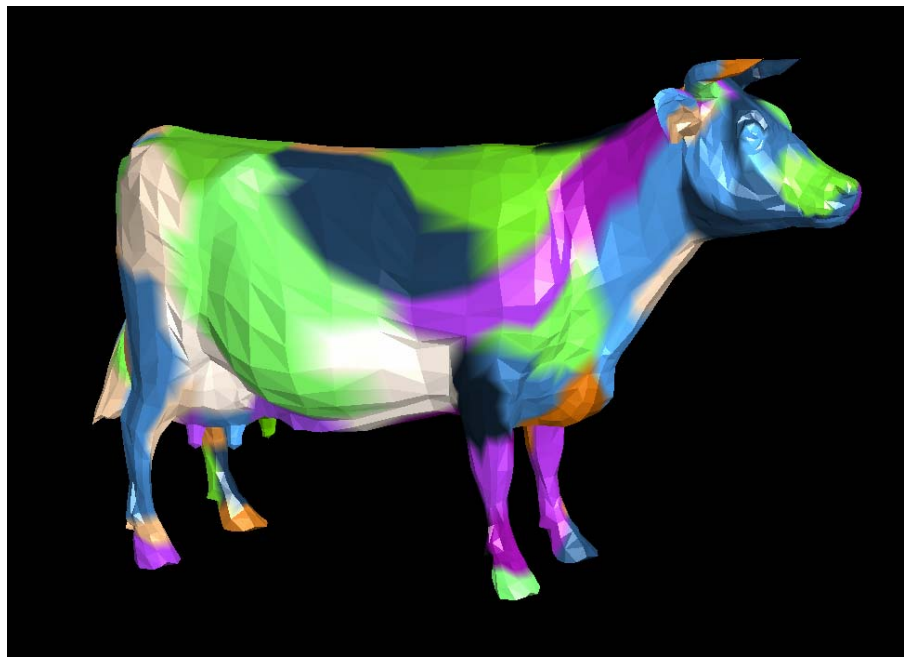
Representation Discovery in Graphics

- Graph partitioning (Karypis and Kumar, SIAM 1998)
  - Coarsening: $|V_0| > |V_1| > \ldots |V_m|$
  - Partitioning: Divide V_m into two parts
  - Uncoarsening: Project P_m onto original graph

Laplacian



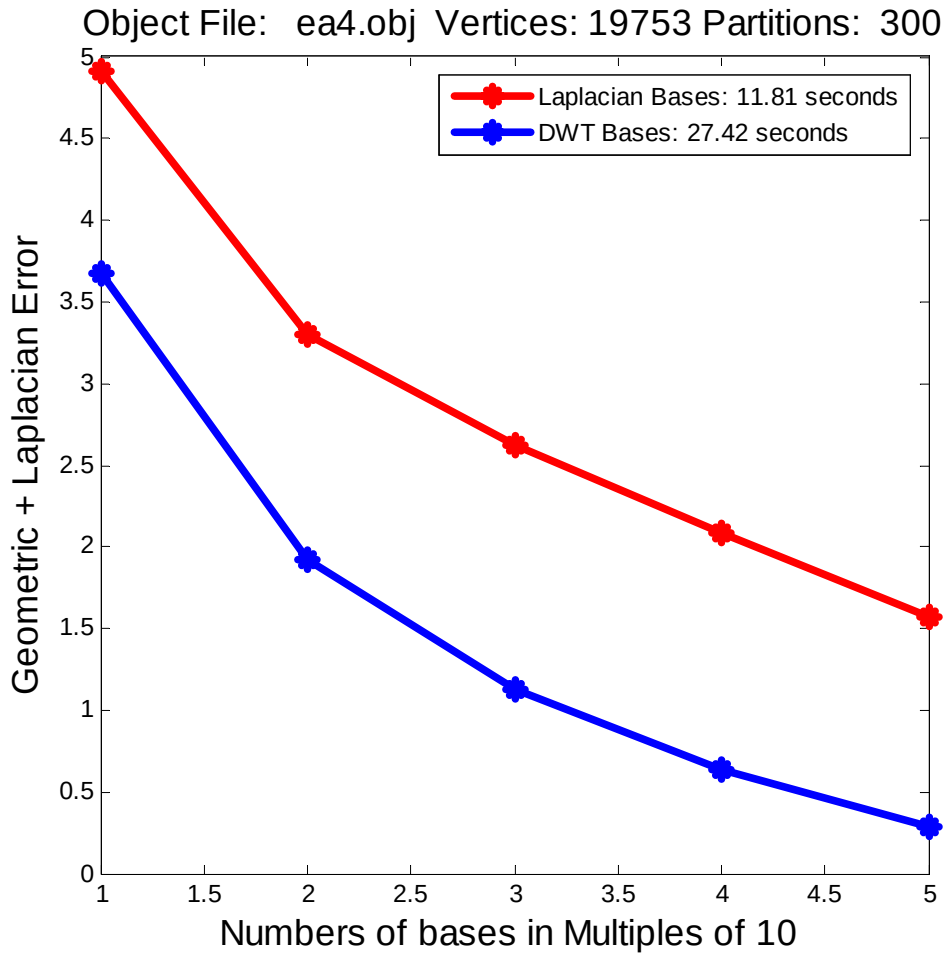
Wavelet



Fourier vs wavelet bases
(Mahadevan, ICML 2007)

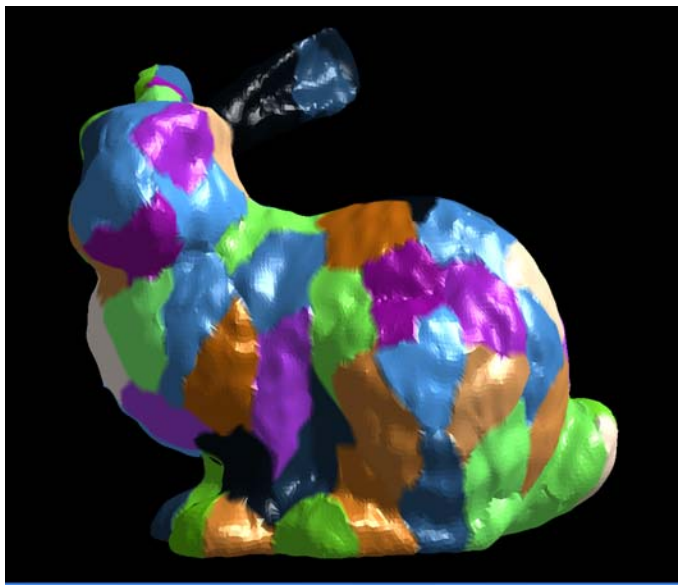
"Elephant"

~20,000 vertices

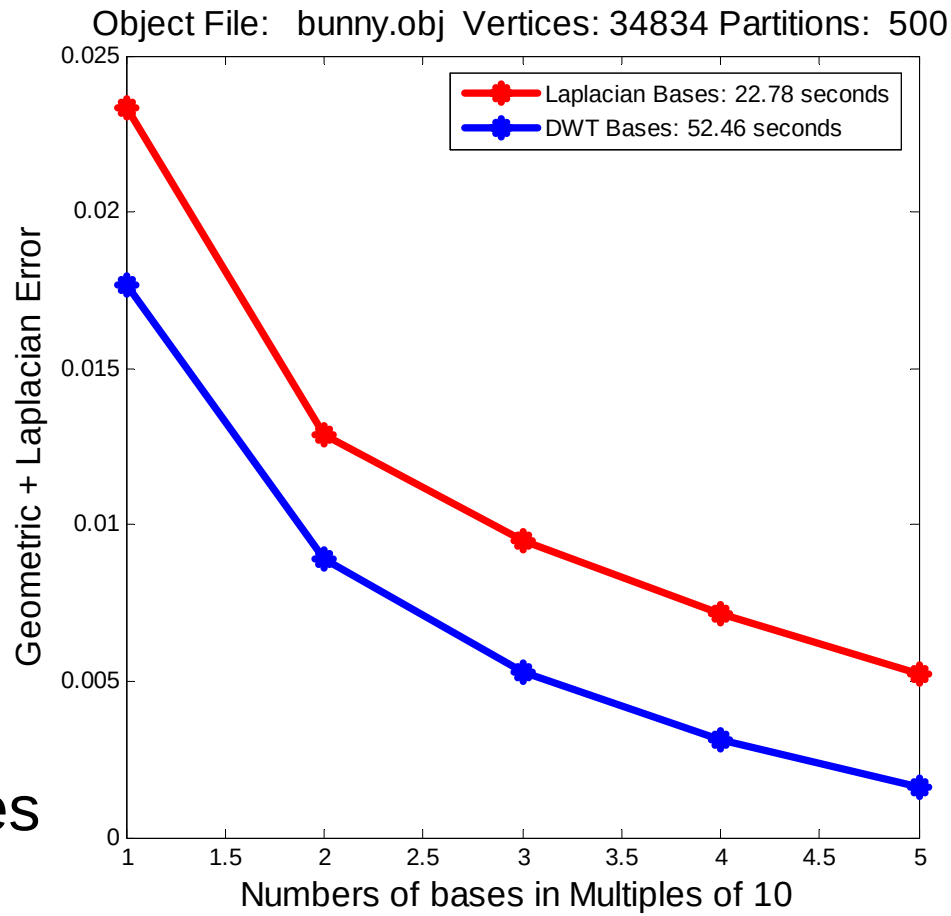


Scaling to Large 3D Objects

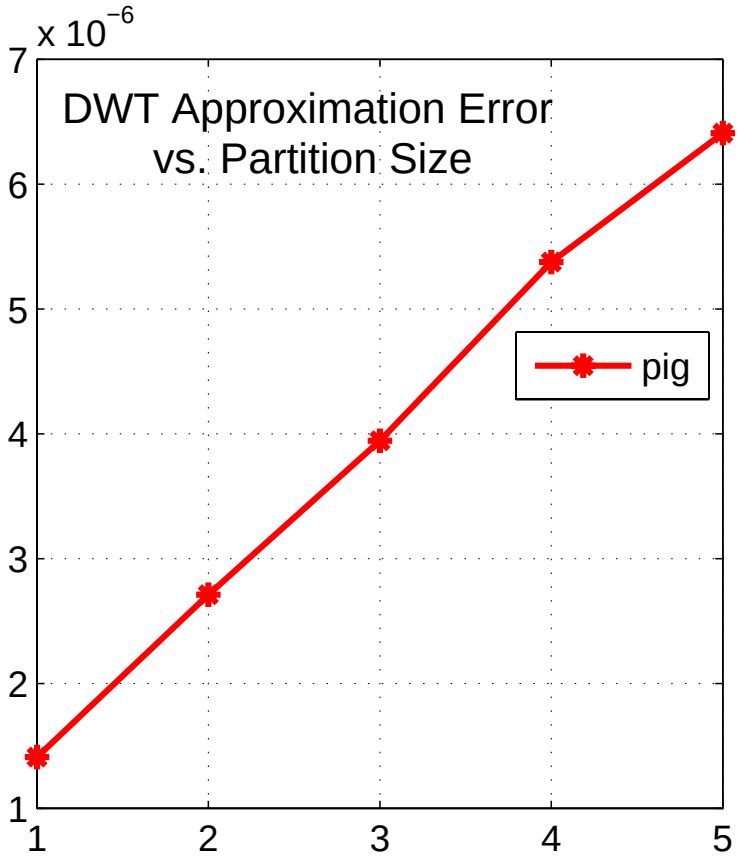
Stanford “Bunny”

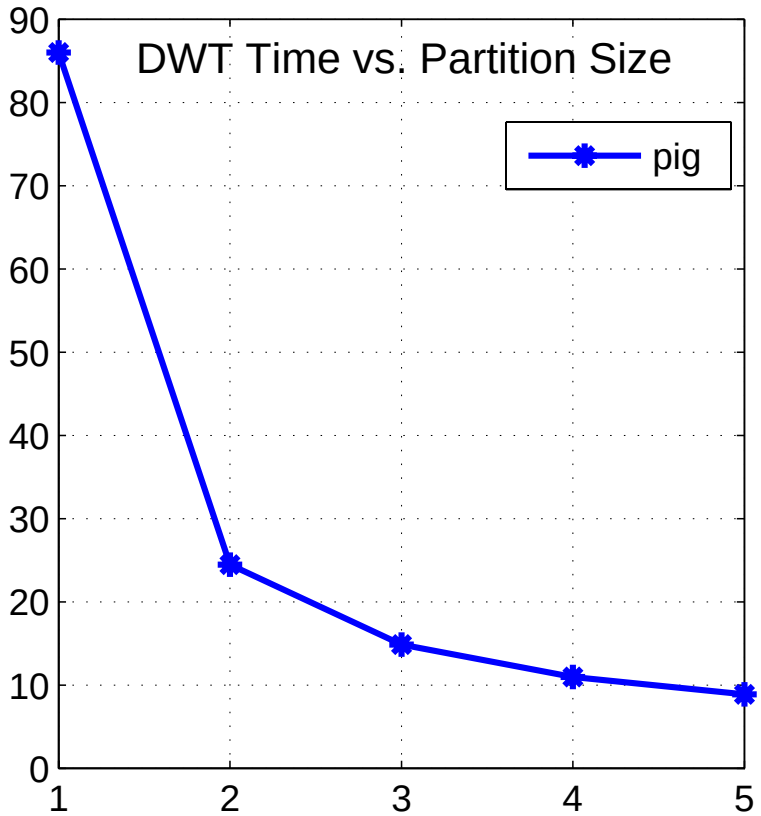


35000 vertices, 100,000 edges



Error vs. Number of Partitions





Kronecker Product Definition

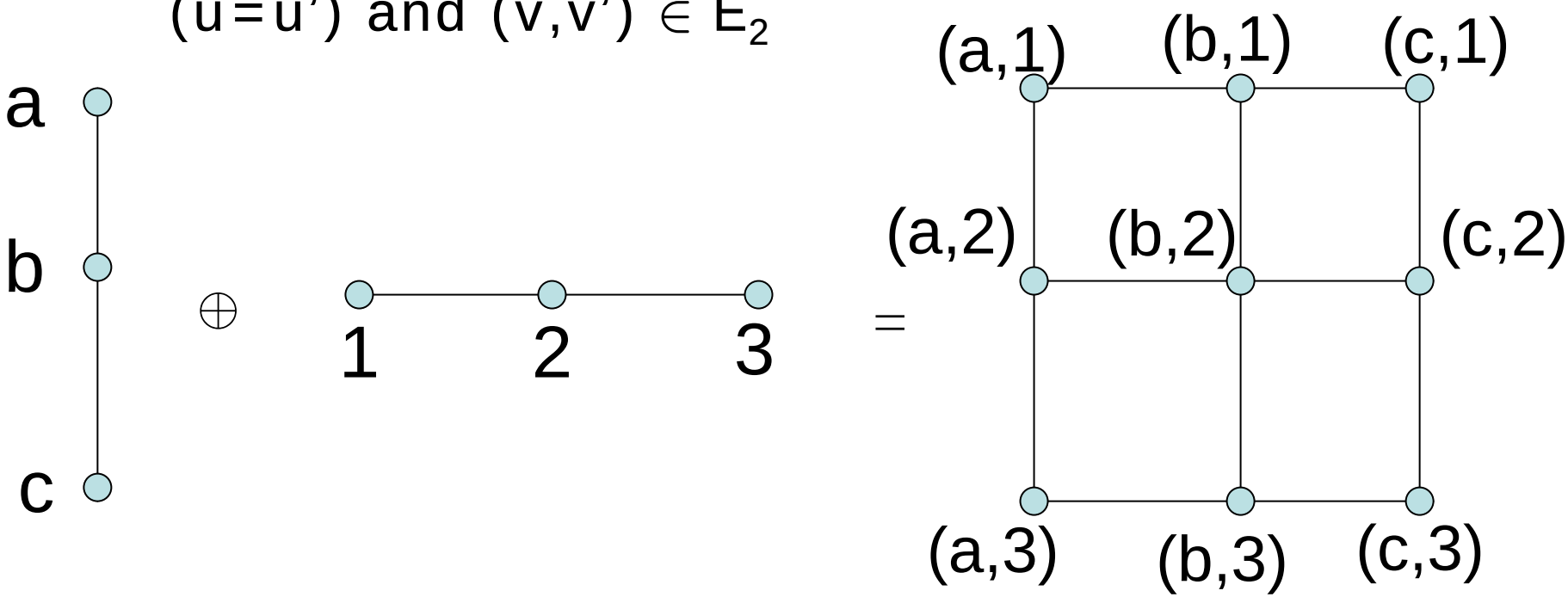
- Consider two matrices B and C. Let A equal the Kronecker product of B and C, written $A = B \otimes C$.

- Matrix A's size is $[row_B * row_C]$ x $[col_B * col_C]$

$$A = B \otimes C = \begin{bmatrix} b_{11}C & b_{12}C & \cdots & b_{1n}C \\ b_{21}C & b_{22}C & \cdots & b_{2n}C \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1}C & b_{m2}C & \cdots & b_{mn}C \end{bmatrix}$$

Kronecker Sum Graphs

- The **Kronecker sum** of two graphs $G = G_1 \oplus G_2$ is the graph with vertex set $V = V_1 \times V_2$ and adjacency matrix $A = A_1 \otimes I_2 + I_2 \otimes A_1$
  - Alternative definition: The Kronecker sum graph G has an edge between vertices (u,v) and (u',v') if and only if $(u,u') \in E_1$ and v=v' or (u=u') and $(v,v') \in E_2$



Spectral Theory of Tensor Products

- Let $A_{r \times r}$ and $B_{s \times s}$ be two matrices of full rank
- Let (λ_i, u_i) and (μ_j, v_j) be the i^{th} eigenvalue and eigenvector of graph A and B, respectively
- Spectra of tensor sum and products:
  - $(A \otimes B)\ (u_i \otimes v_j) = \lambda_i\ \mu_j\ (u_i \otimes v_j)$
  - $(A \otimes I_s + I_r \otimes B)\ (u_i \otimes v_j) = (\lambda_i + \mu_j)\ (u_i \otimes v_j)$
- This result is based on the following identity
  - $(A\ C) \otimes (B\ D) = (A \otimes B)\ (C \otimes D)$ (if AC and BD are well-defined)

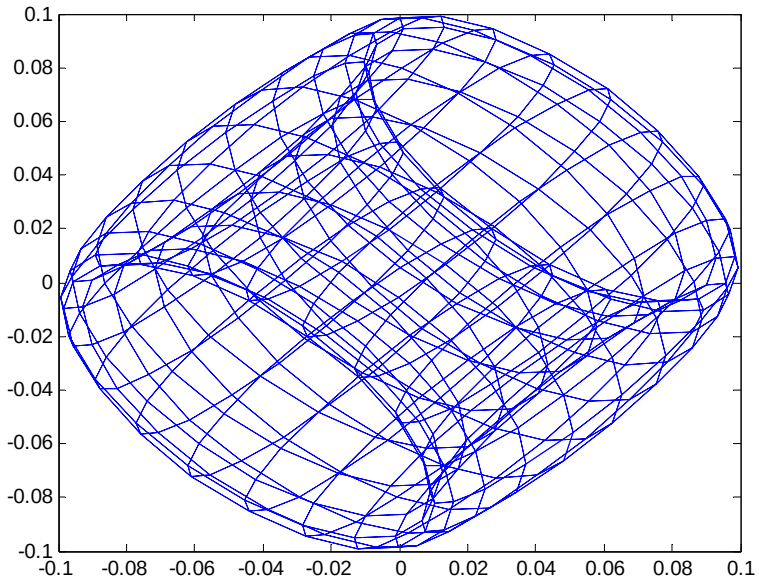
Laplacian of Kronecker Sum Graphs

- If L_1, L_2 be the combinatorial Laplacians of graphs G_1, G_2, then the spectral structure of the combinatorial Laplacian of the Kronecker sum of these graphs $G = G_1 \oplus G_2$ is specified as

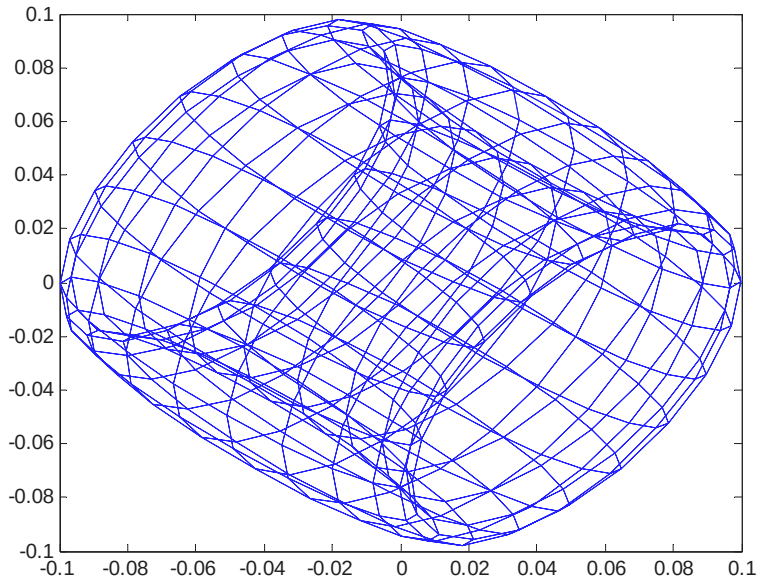
$$\sigma(L), \mathcal{X}(L)) = \{\lambda_i + \mu_j, l_i \otimes k_j\}$$

- where λ_i is the i^{th} eigenvalue of $L(G_1)$ with associated eigenvector l_i and μ_j is the j^{th} eigenvalue of $L(G_2)$ with associated eigenvector k_j.

Embedding of Structured Spaces



Torus

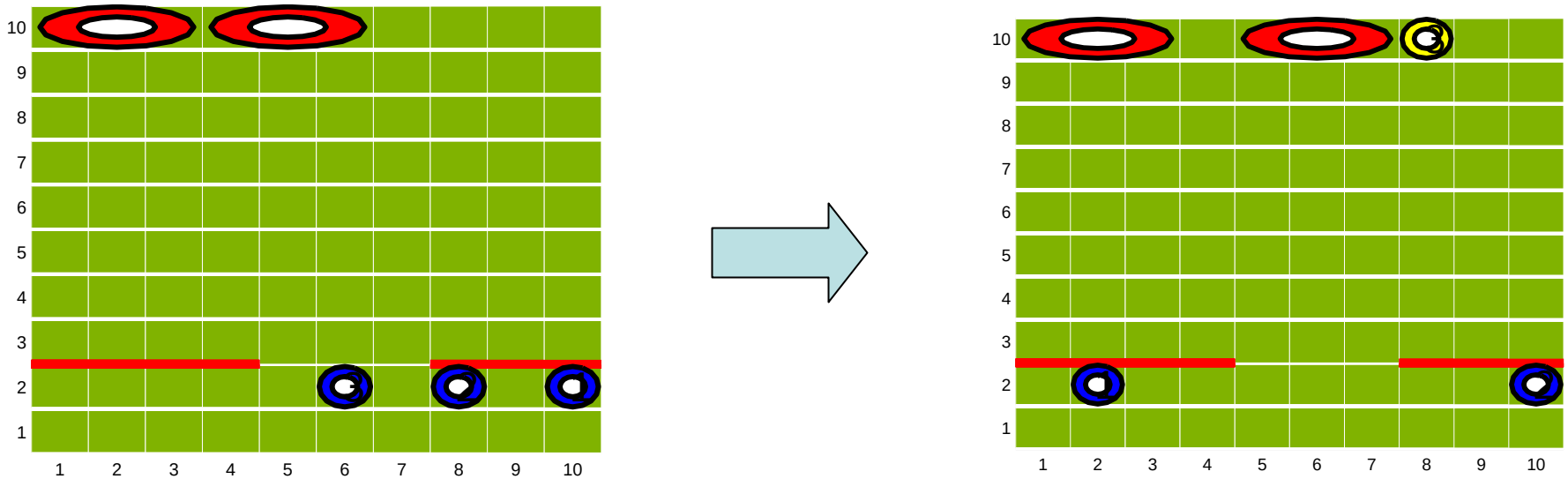


2nd and 3rd eigenvectors
of combinatorial Laplacian

Blockers Domain

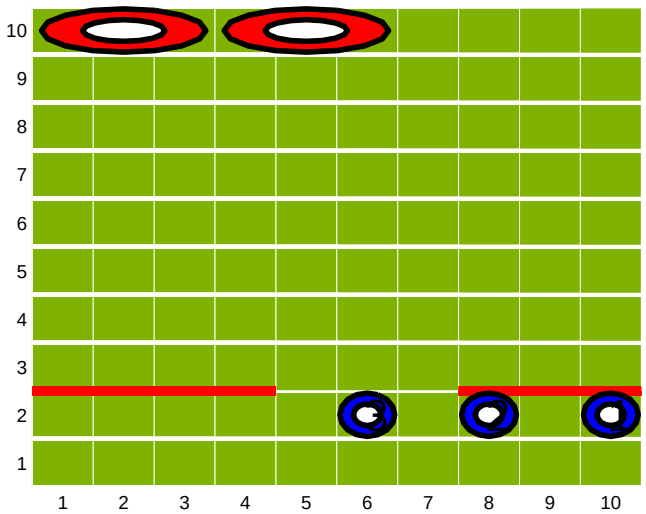
(Sallans and Hinton, JMLR 2003)

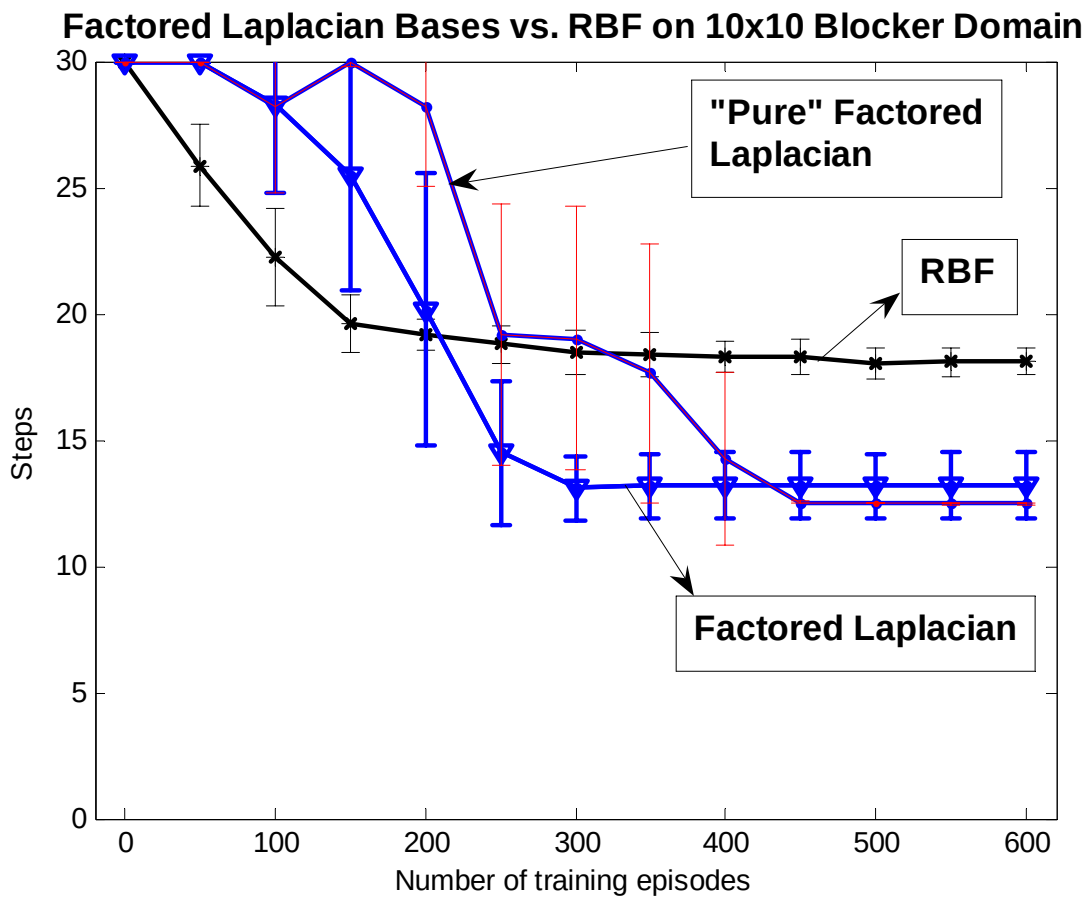
Large state space of > 10^6 states



Large Factored MDP: Blockers Domain

Topologically, this space is the tensor product of three “irregular” cylinders





Discovery of Factored Representations for Markov Decision Processes

(Johns and Mahadevan, AAAI 07)

- Goal: Given basis matrix A, find the best matrices B and C such that $A \approx B \otimes C$

- Create approx. eigenvectors of A by computing the eigenvectors of B and C and combining them via the Kronecker product

- Benefits for Proto-value functions:
  - Computing eigenvectors of much smaller matrices
  - Storing eigenvectors in more compact form
  - Benefits are magnified if factorization is done recusively!

Automatic Kronecker Factorization
(Pitsianis and van Loan, 1993)

- Given: matrix A
  dimensions for matrix B

- Return: B and C such that $\min_{B,C} \| A - B \otimes C \|_F$

- Frobenius norm:

$$\| X \|_F = \sqrt{\sum_i \sum_j x_{i,j}^2}$$

Rank-One Problem in Disguise

- Problem: $\min_{B,C} \left\| A - B \otimes C \right\|_F$
- Pitsianis ('97) solved this problem for *arbitrary* A

$$\left\| A - B \otimes C \right\|_F = \left\| \tilde{A} - \mathrm{vec}(B) \otimes \mathrm{vec}(C)^T \right\|_F$$

$$= \left\| \tilde{A} - \underbrace{\mathrm{vec}(B) \cdot \mathrm{vec}(C)^T}_{\text{rank-1 matrix}} \right\|_F$$

Rank-One Solution

- Optimal solution is to take the SVD of $\tilde{A}$ and use the first columns and first eigenvalue

$$\min_{x,y} \left\| \tilde{A} - x \cdot y^T \right\|_F \quad \text{has same minimizer as}$$

$$\min_{x,y} \left\| \tilde{A} - x \cdot y^T \right\|_2 = \left\| \tilde{A} - \sigma_1 u_1 v_1^T \right\|_2$$

where $U^T \tilde{A} V = \text{diag}([\sigma_1 \sigma_2 \ldots \sigma_q])$

$A \in R^{mxn}$

$x \in R^m$

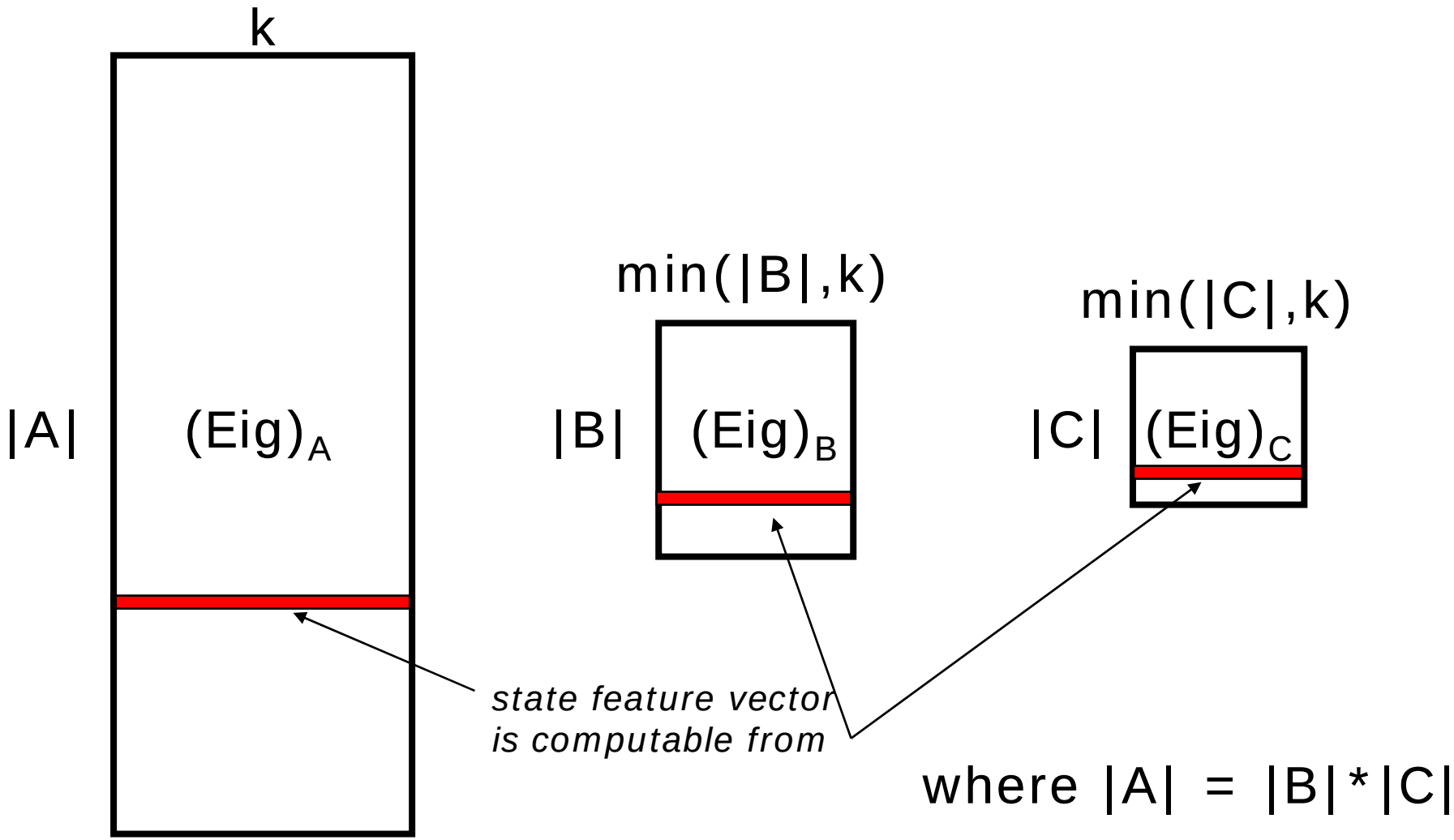
$y \in R^n$

$q = \min(m,n)$

$\text{vec}(B) = \sigma_1 u_1$

$\text{vec}(C) = v_1$

Basis Compression

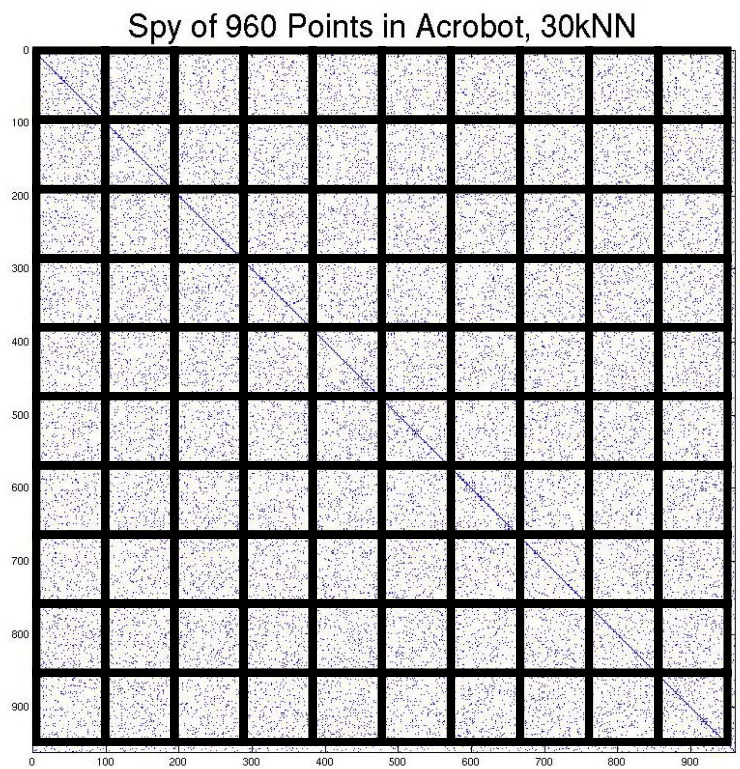


Overall Method

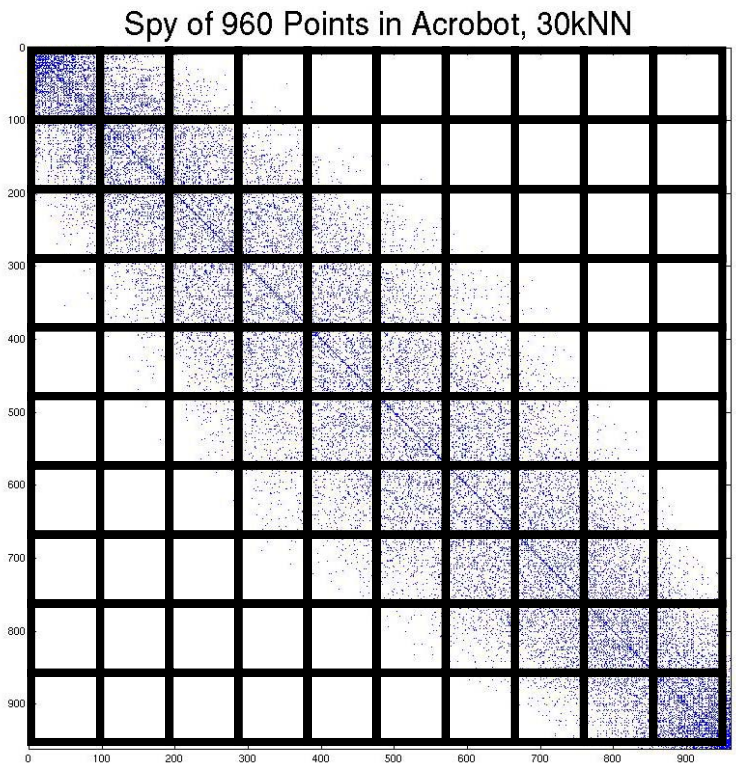
- Form a graph with weight matrix W by connecting states to nearby neighbors
- Form the random walk matrix $P = D^{-1} W$
- Represent P with two smaller stochastic matrices B and C such that $B \otimes C \approx P$
- Compute eigenvectors of B and C
- When we need an embedding (feature vector) for a state ϕ_P during learning, compute it from ϕ_B and ϕ_C

Discovery of Factorized Structure

Original Matrix

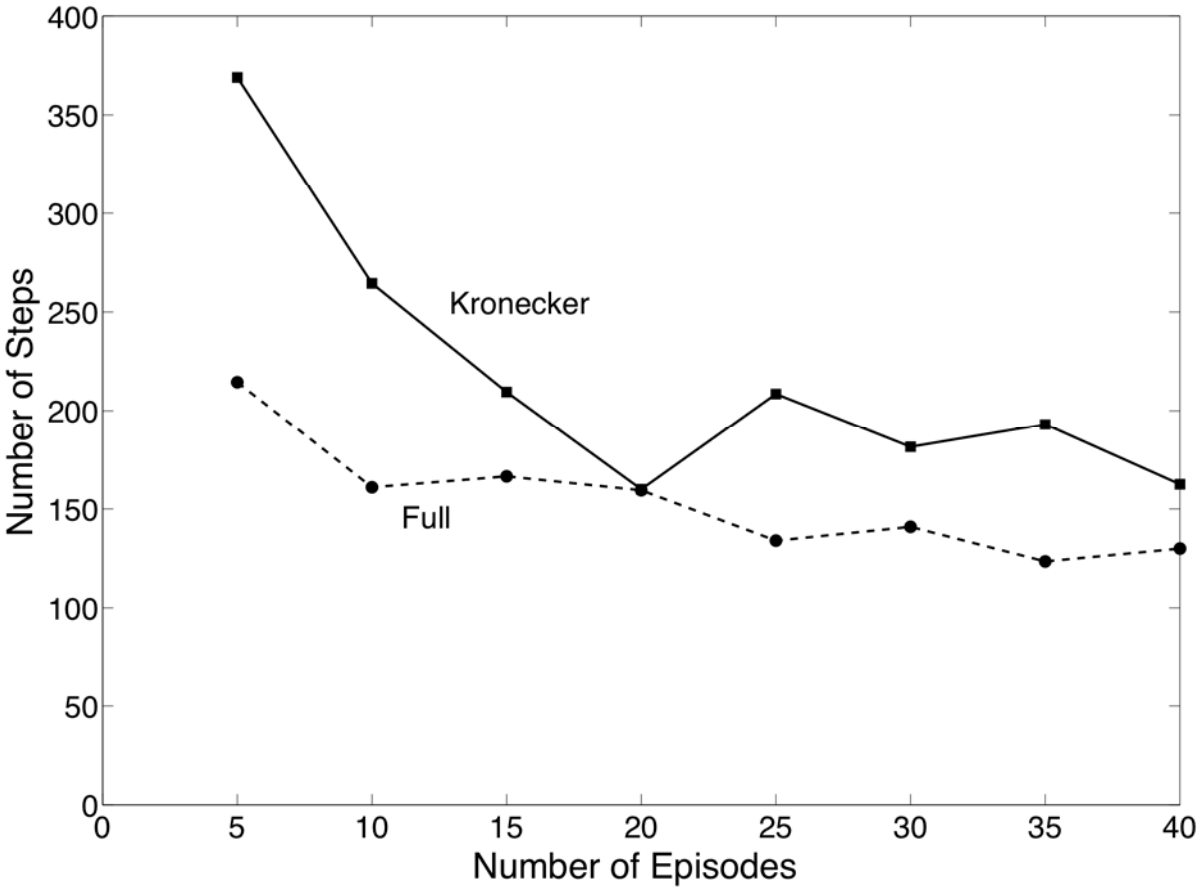


Reordered Matrix

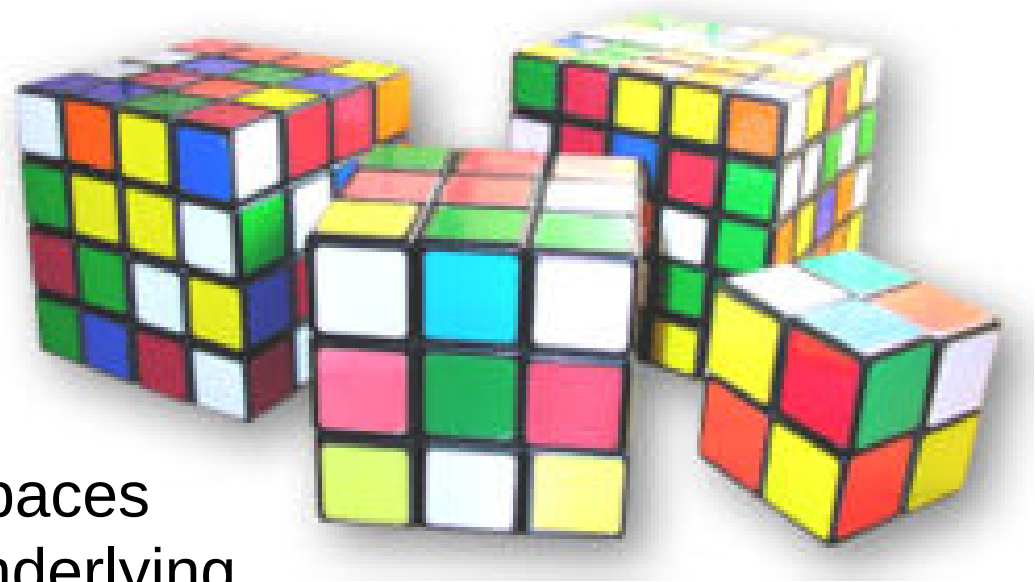


Acrobot (36x compression)

$|P| \approx 1800$
$|B| \approx 60$
$|C| = 30$



Rubiks Cube



Large state spaces have many underlying symmetries

Groups

- A group G is a set, along with an operation . : $G \times G \rightarrow G$
- In Abelian groups, a.b = b.a
  - Real numbers under addition
- Non-Abelian groups:
  - Let G be any graph, and consider a mapping ϕ:V $\rightarrow$ V that respects adjacency
  - If (u,v) $\in$ E, then (ϕ(u), ϕ(v)) $\in$ E
  - The set of all automorphisms forms a group

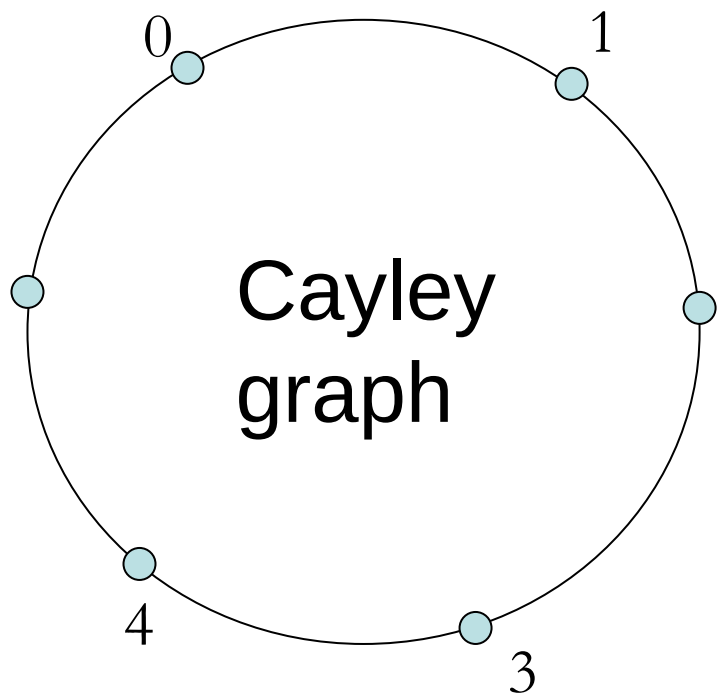
Fourier Analysis on Groups

- Given a group G, and a representation ρ, the Fourier transform of any function f on G is given by

$$\mathbf{f}_\rho = \sum_g f(g)\, \rho(g)$$

Graph generator:
i +/- 1 mod 6

$$\rho_j(k) = e^{-(2\pi i\, j\, k)/N}$$



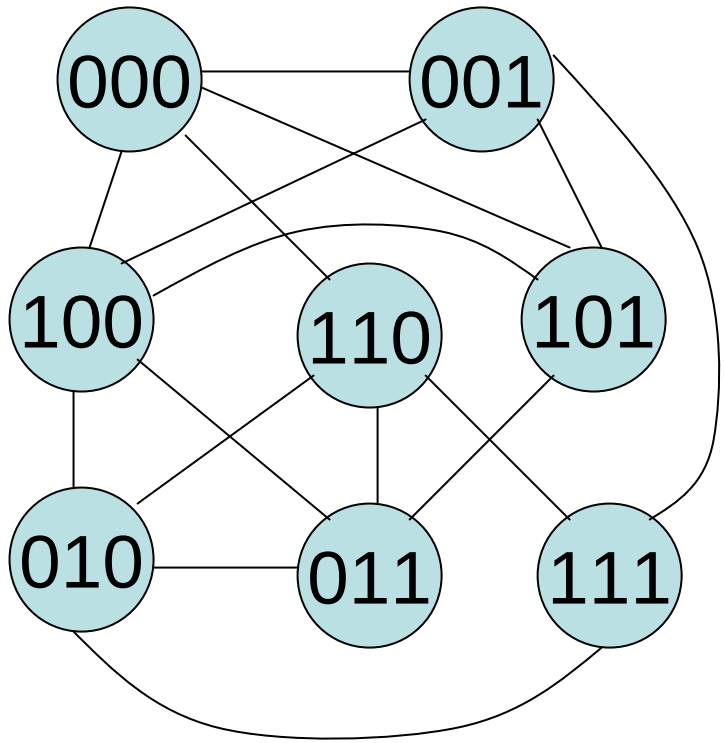
Abelian group

Cayley Group Representation of Boolean Functions

$f = x_1 \neg x_3 \vee \neg x_2 x_3$

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$f(m_1 \oplus m_2) = 1$



Cayley graph group under the operator $\oplus$

Matrix Groups

- Many problems in robotics generate manifolds that can be modeled as continuous matrix (Lie) groups
- GL(n) is the group of all (real or complex) invertible matrices (under matrix multiplication)
- The set of matrices A such that $A^T A = 1$ forms a subgroup of GL(n) and is called the orthogonal group O(n)
- Length-preserving transformations (e.g. rotations) form a subgroup of O(n) called SO(n) (here, the matrices have determinant = 1)

SO(2)

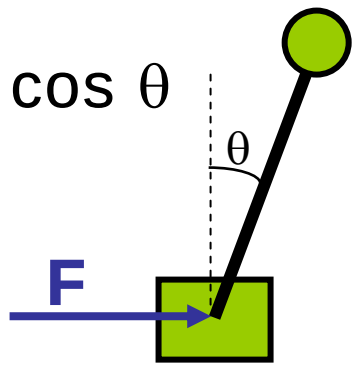
- SO(2) is the group defined by all rotations on the plane
- This group can be represented in several different ways
  - As a set of orthogonal rotation matrices

$$\begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$$

  - The new coordinates are given by

$$x \cos\theta - y \sin\theta,\ x \sin\theta + y \cos\theta$$

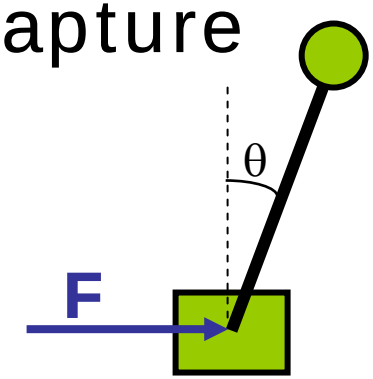
  - As a set of complex numbers $e^{i\theta}$



SE(2)

- If you include translations, it is easy to show that no 2x2 matrix representation exists
- However, it is possible to augment the representation to a 3x3 matrix to capture rotations and translations

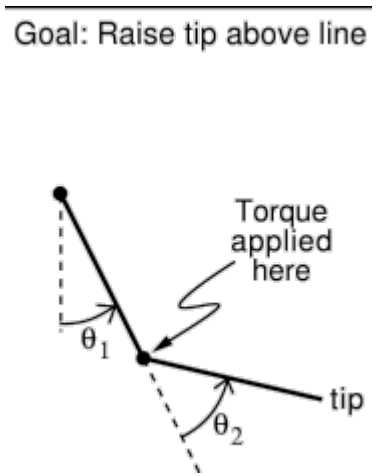
$$\begin{vmatrix} \cos\theta & -\sin\theta & x_1 \\ \sin\theta & \cos\theta & y_1 \\ 0 & 0 & 1 \end{vmatrix}$$



$x \cos\theta - y \sin\theta + x_1,\ x \sin\theta + y \cos\theta + y_1$

Kinematic Chains

- Consider the Acrobot task, defined by motions of a 2-link robot arm
- What manifold does this define?
- The manifold can be represented by products of SE(2) matrices



$$\begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 1 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Group Representations
(Serre)

- A linear representation of a group G is a mapping from elements of G to invertible matrices M such that

$$\rho(a.b) = \rho(a) \,.\, \rho(b)$$

- Group theory provides a powerful framework to study many problems in machine learning
  - Compact representations of eigenvectors and wavelet representations
  - Applicable to all the domains described here
  - See tutorial by Risi Kondor at ICML 2007

Noncommutative harmonic analysis

- NHA is a generalization of spectral learning
- In spectral analysis, data is projected onto eigenvectors, which is a 1-dimensional *invariant* subspace
- This is an example of commutative harmonic analysis (Fourier)
- There are many problems which cannot be reduced down to 1-dimensional subspaces.
- For example, consider the problem of finding structure in voting data

Group-Theoretic Analysis of Ranking
(Diaconis and Rockmore, 1990)

- The set of rankings of n objects is a subgroup of the symmetric group S_n
- A representation of the symmetric group is the *permutation* representation: $\rho(\pi)_{ij}$ = 1 if permutation $\pi(i) = j$.
- $\mathbf{f}_\rho = \sum_\pi f(\pi)\ \rho(\pi)$ computes the first-order summary statistics (the (i,j) entry counts the number of times object i is ranked j)
- Other representations reveal higher-order structure such as *coalitions*

Parametric Manifold Analysis

- Can knowledge of the underlying manifold be used to scale representation learning?
- How can invariants be exploited?
  - Rigid body actions preserve lengths
  - Graph automorphisms can be exploited
  - Learn Lie group generator (Rao and Ruderman, NIPS '98)
- How can AI systems discover abstract properties of the state space?

Future Challenges

- Scaling to large state spaces
  - Backgammon, chess, humanoid robot control, natural language, and web structure analysis
- Transfer learning
  - Can representations learned in one domain be mapped to a new domain?
- First-order representation discovery
  - How can these methods be extended to richer representations?
- Can we get deeper insight into human representation discovery?