Mining State-Space Graphs to Build Value-Function Representations

Bryan Silverthorn
Craig Corcoran

9 December 2011

Department of Computer Science
The University of Texas at Austin

Problem Description

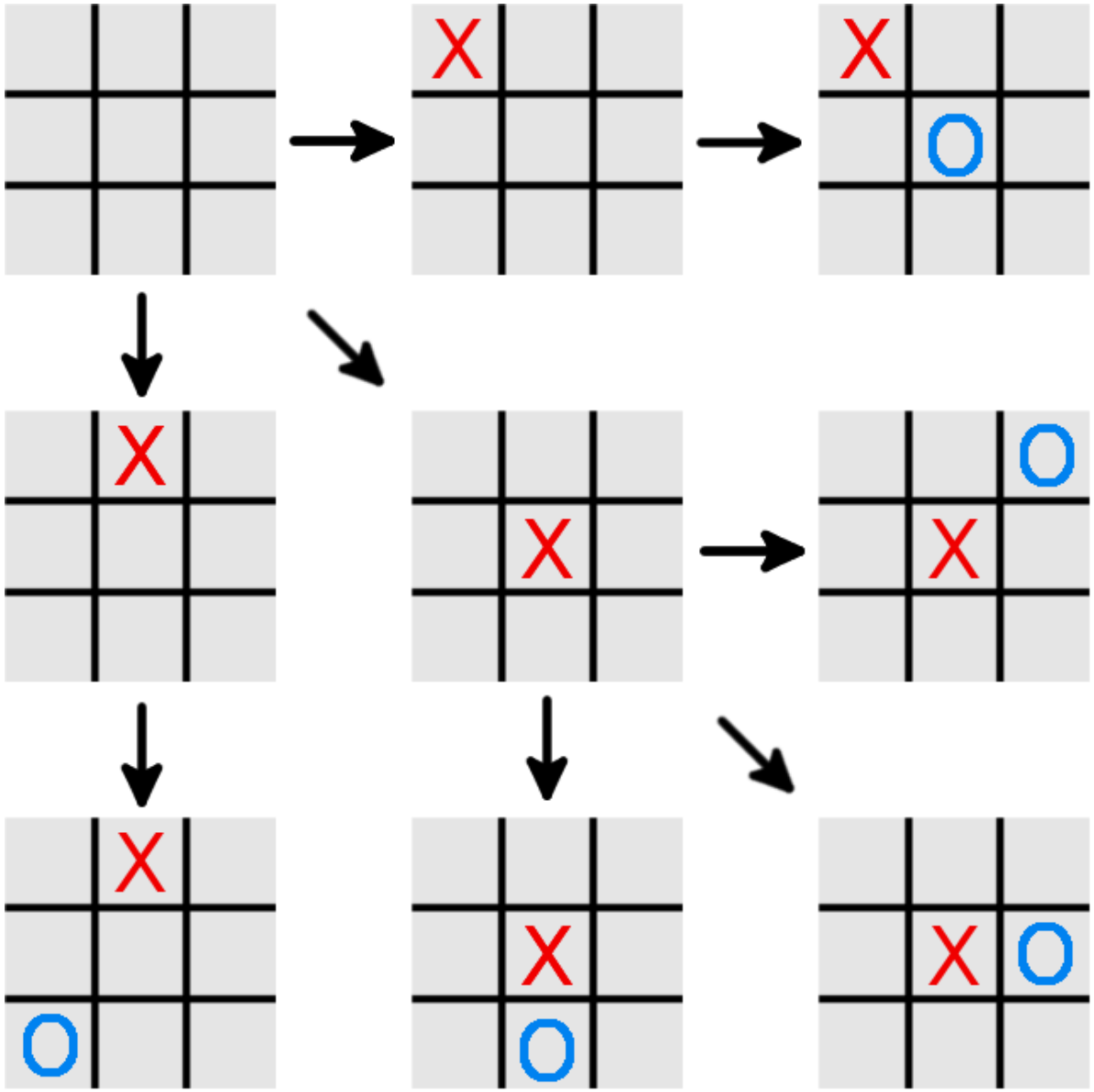
- Model board game as an MDP
- We want a linear estimate of the value function:

$$V^*(x) = \max_{a \in \mathcal{A}_x} [R(x) + \gamma V(T(x, a))]$$
$$V(s) = w^T \phi(s)$$

- Good estimate of $V(s) \rightarrow$ good greedy policy
- Generate $\phi(s)$ automatically

But how do we generate features?

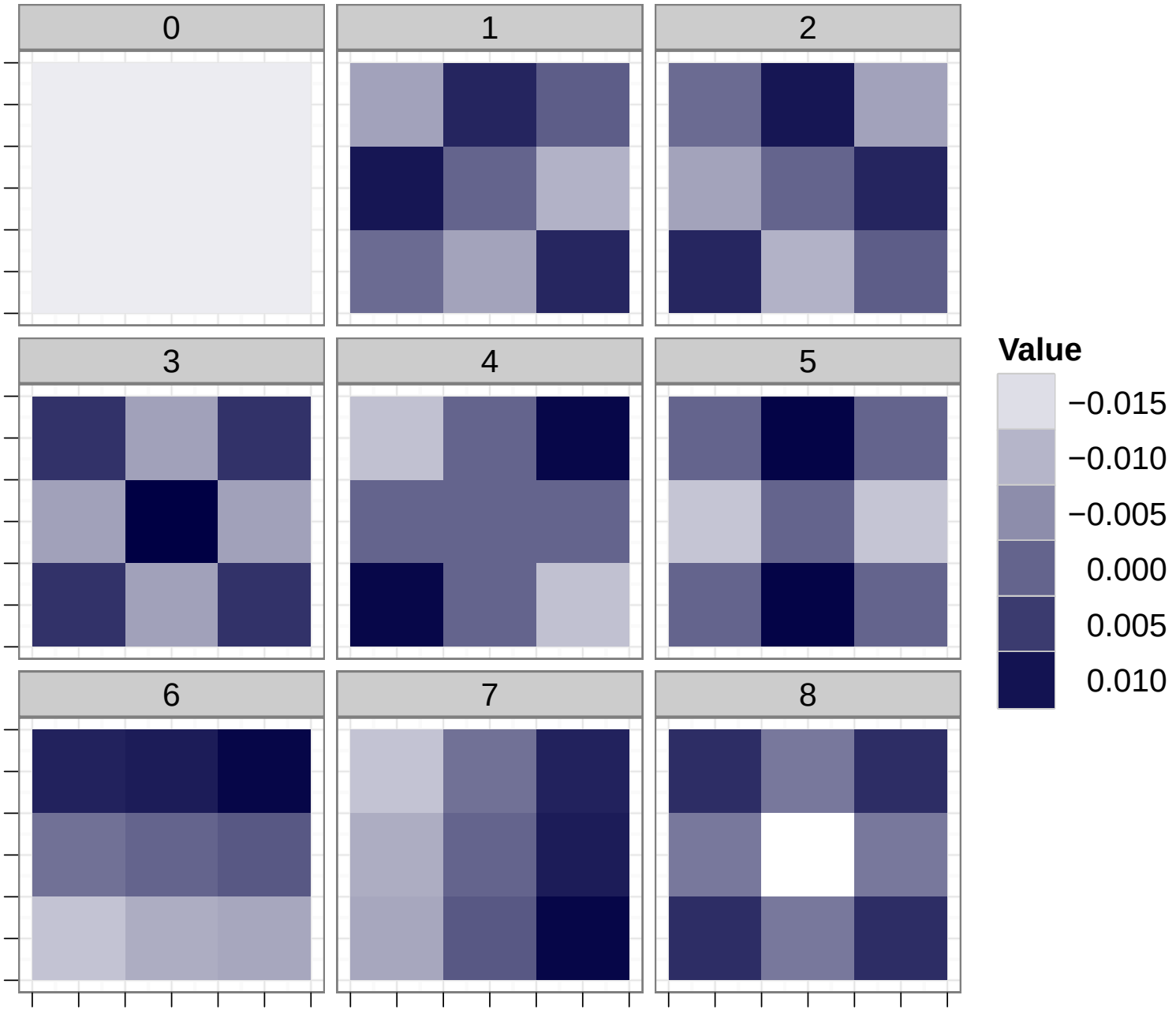
. . . mine the state-space graph:



What to do with the graph?

- Build the weighted adjacency matrix W
- Form the graph Laplacian: $L = D - W$
- Take k "smallest" eigenvectors: $Lv_i = \lambda_i v_i, \lambda_i <= \lambda_{i+1}$
  - (as in spectral clustering)
  - Good basis for smooth functions on graph $\rightarrow$ features

Eigenvectors of Tic-Tac-Toe



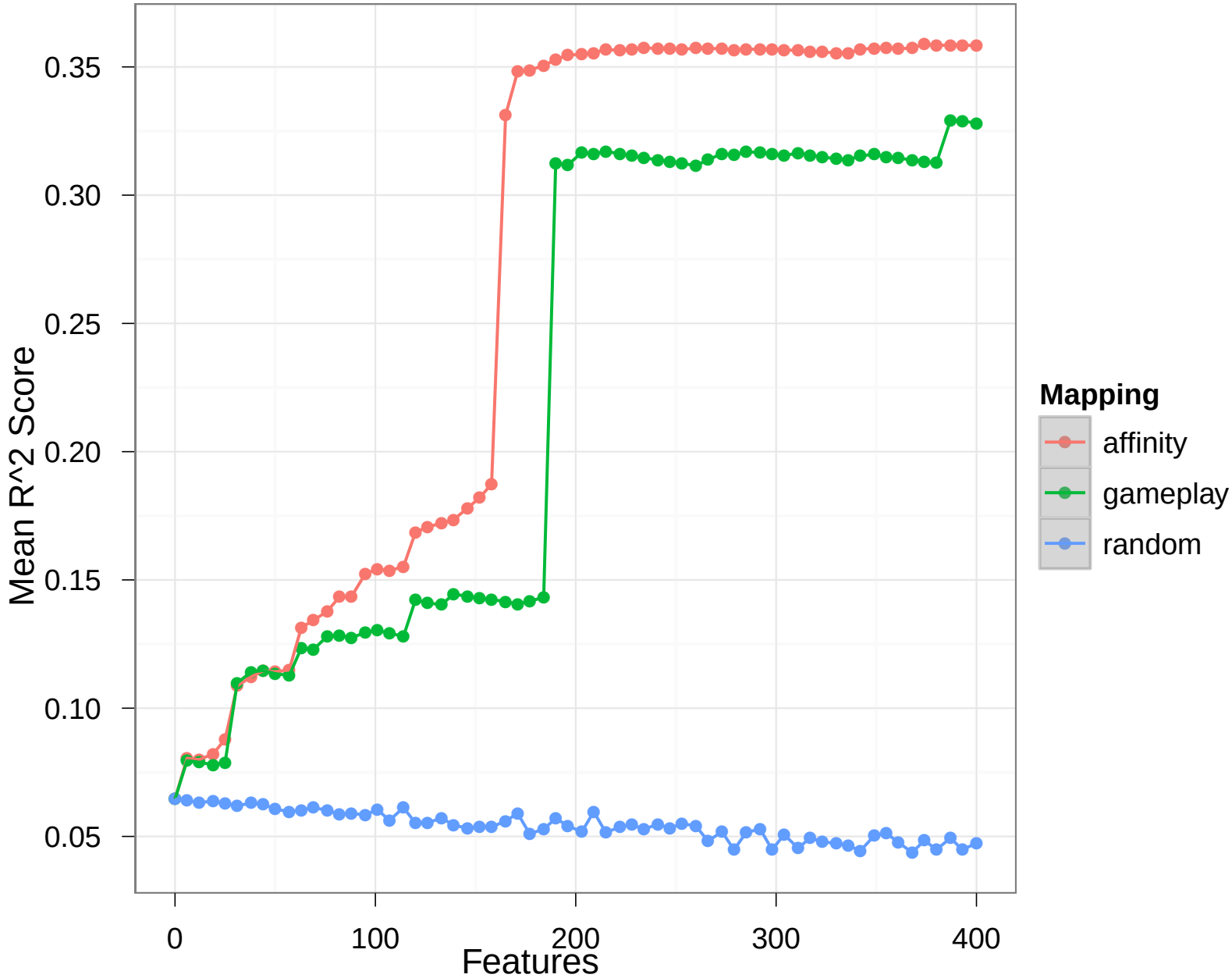
Affinity Graph for Large Domains

In large games, using full state-space graph is intractable.

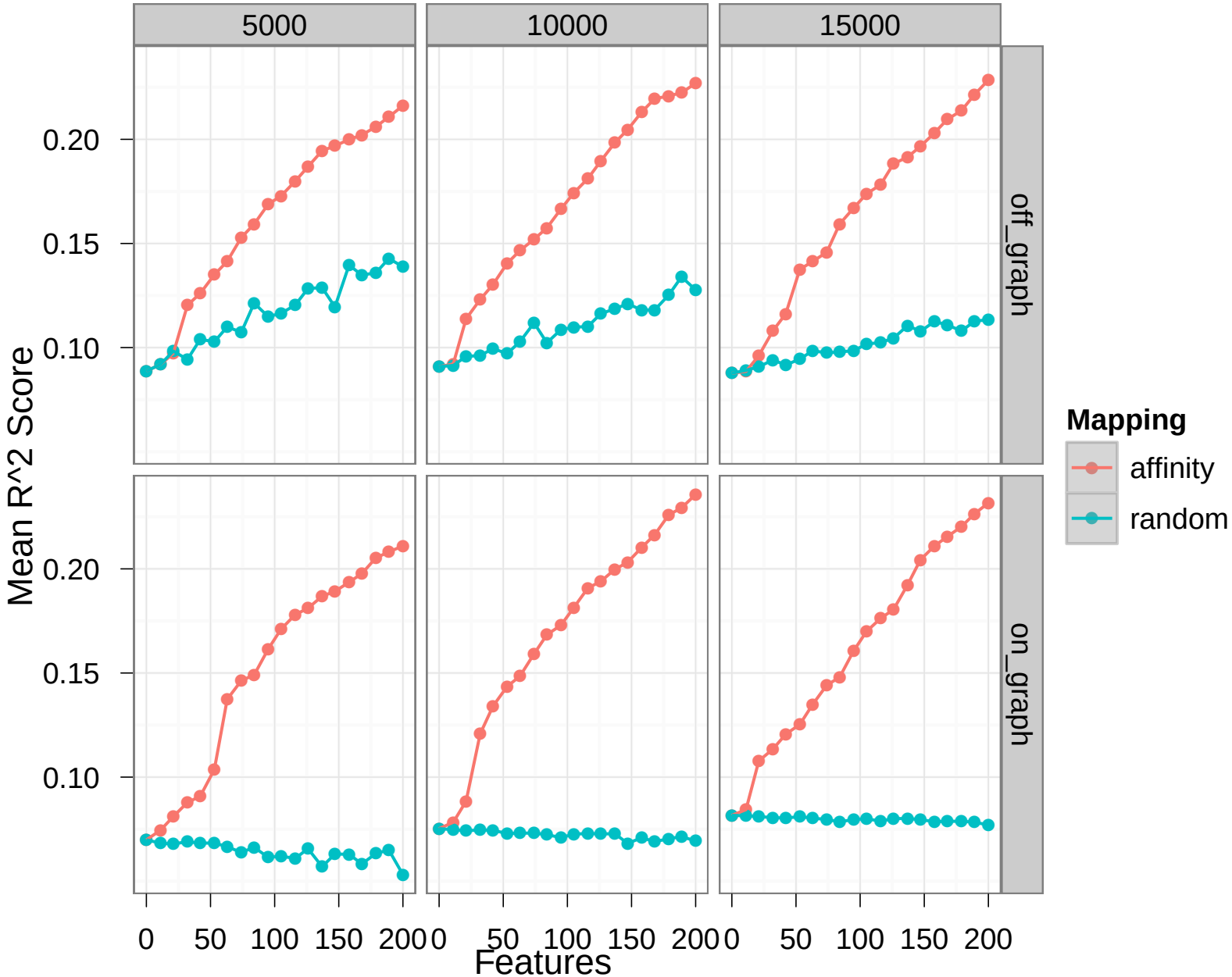
- Sample from recorded expert games
  - Reveals relevant region of state space
- Form k-NN graph in board space $\rightarrow W$
- Interpolate to new states (off-graph)

$$W_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$$

TTT Value-Function Prediction Error



Results in Go



Demo!

`http://www.cs.utexas.edu/~bsilvert/ttt.html`

Future Work

Obstacles to scaling

- Large number of samples
  - Constructing k-NN graph
  - Eigenvalue computation

Work in Progress

- Using k-means to cluster samples
  - Perform feature generation on subgraphs
- Use a better affinity space representation
  - Use hand-crafted, symmetry-invariant features
    - Feature amplification
  - Learning