Family Discovery

Stephen M. Omohundro
NEC Research Institute, Inc.
4 Independence Way
Princeton, New Jersey 08540
om@research.nj.nec.com

Poster presented by Stephen M. Omohundro at:
Neural Information Processing Systems, Denver, Colorado, November 1995.

The Standard Learning Paradigm

- An underlying stochastic model $p(x)$ representing a density, classifier, mapping, stochastic grammar, etc.
- Training samples x_i drawn from the model.
- A learning algorithm chooses a model $\hat{p}$ using the training data.
- Use $\hat{p}$ to make predictions.

Family Discovery

- A parameterized family of stochastic models $p_\gamma(x)$ representing densities, classifiers, mappings, stochastic grammars, etc.
- A training set partitioned into N episodes.
  Episode i is N_i samples drawn from p_{γ_i}.
- Family discovery estimates the dimension and structure of the parameterization $\hat{p}_\gamma$ from the training episodes.
- Test episodes first estimate the parameter value γ^*, then make predictions from $\hat{p}_{\gamma^*}$.
- Potential benefits: Better episode models, better interpretation of novel examples.

Example: Multi-speaker Speech Recognition

- Training examples are labelled segments of speech.
- Speech model is parameterized by accent.
- Training episodes correspond to different speakers.
- Family discovery would find a parameterized model for accent.
- For recognition, first recognize the "accent parameters", then use a speech model (eg. HMM) with those parameters.

Poster presented by Stephen M. Omohundro at:
Neural Information Processing Systems, Denver, Colorado, November 1995.

Example: Omni-font Character Recognition

- Training examples are labelled images of text.
- Text model is parameterized by font.
- Training episodes correspond to text in different fonts.
- Family discovery would find a parameterized model for font.
- For recognition, first recognize the "font parameters", then use a character model with those parameters.

Alternative 1: Separate Models

- Train separate models for each parameter setting.
- Eg. Speaker-dependent speech recognition.
- Eg. Font-dependent character recognition.
- Disadvantage: Don't know how to deal with new parameter values.
- Disadvantage: No transfer of learning between parameter values.

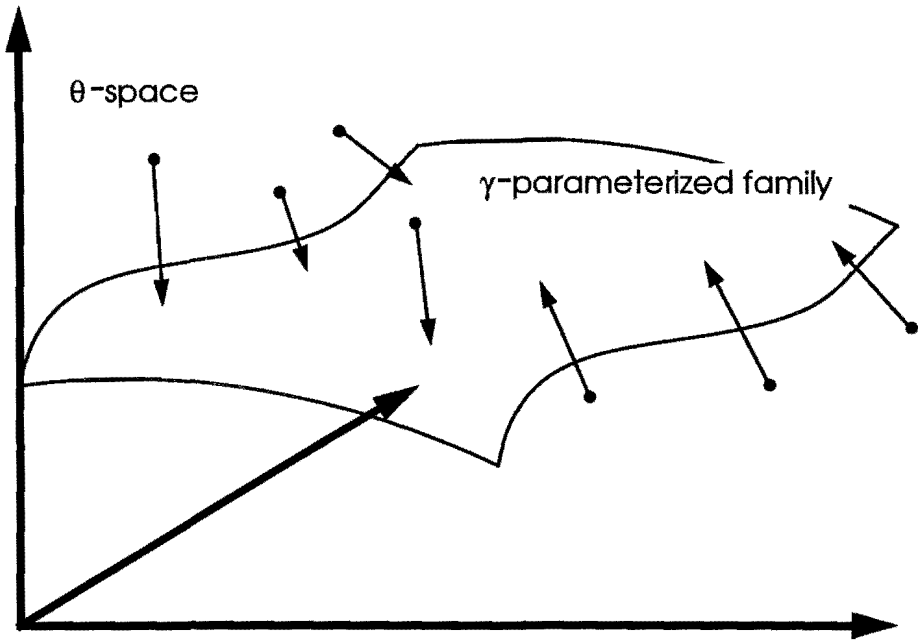
Alternative 2: A Single Non-parameterized Model

- Train one model on all the data.
- Eg. Single HMM for all speakers.
- Eg. Single character model for all fonts.
- Disadvantage: Confusion between samples from different parameter settings.
- Disadvantage: Ignores episode information.

The Family Discovery Algorithms

- Model space parameterized by θ.
- Family is a surface in θ-space parameterized by γ.
- Projection operator P maps a model to the closest model in the family.
- Family defines a prior for recognition: $m_{patch}(\theta) = e^{-(\theta - P(\theta))^2}$.
- Training alternates between fitting the best episode models and fitting the parameterized family.

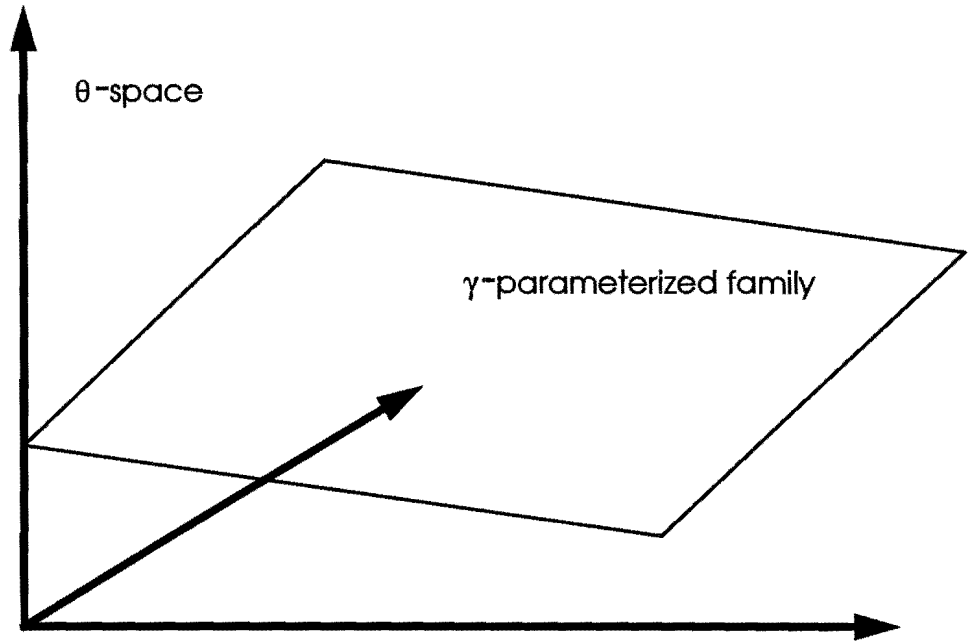
Projection onto the Family



Affine Family

- Affine family: P_{affine} projects θ orthogonally onto an affine subspace.
- Given a set of episode models, use principal components analysis to find the dimension and parameterization of the best fitting subspace.
- Dimension is signified by a gap in the principal values.
- Top principal vectors span the subspace.

Affine Family



Affine Patch Family

Affine patch family: Projection is a smooth convex combination of projections onto affine patches:

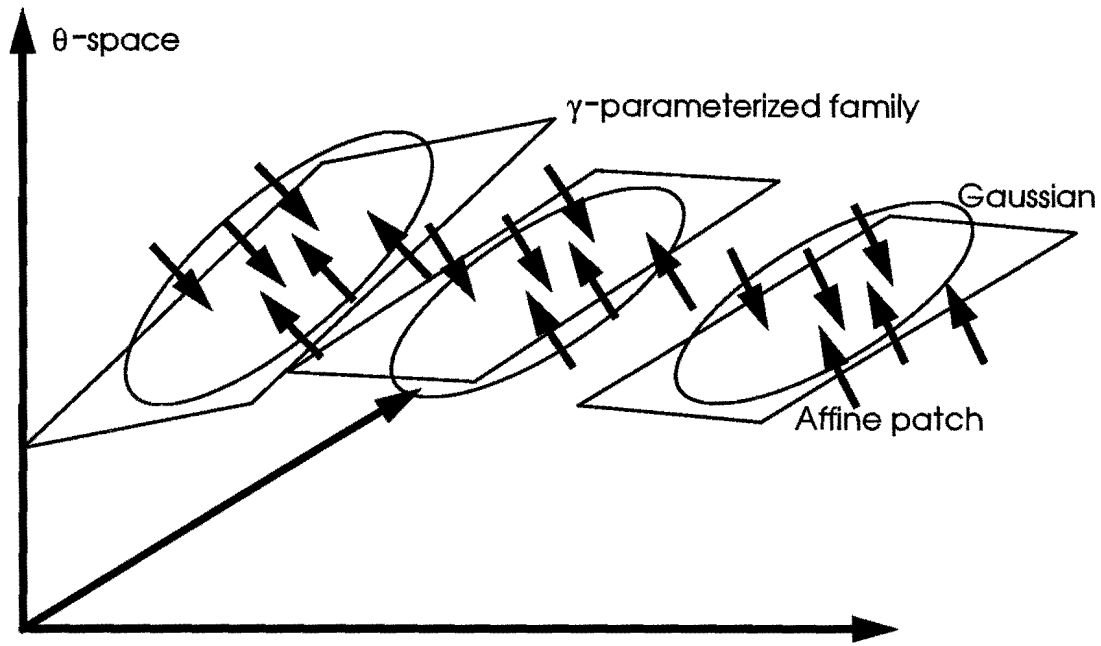
$$P_{patch}(\theta) = \sum_{\alpha=1}^{m} I_\alpha(\theta) A_\alpha(\theta)$$

where A_α is the projection operator for an affine patch and

$$I_\alpha(\theta) = \frac{G_\alpha(\theta)}{\sum_\alpha G_\alpha(\theta)}$$

is a normalized Gaussian blending function.

Affine Patch Family



Learning the Affine Patch Family

- Patches initialized by k-means clustering of the episode models to choose k patch centers.
- Do local principal components analysis on the episode models which are closest to each center.
- The Gaussian influence functions and the affine patches are updated by the EM algorithm.

Coupled Map Family

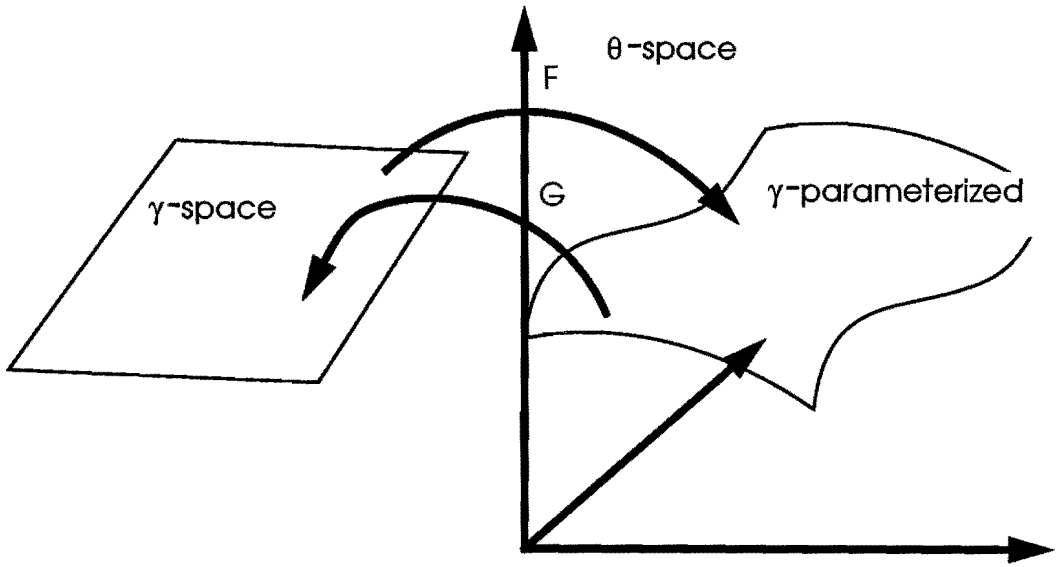
Coupled Map family: The projection P_{map} is a composition:

$$P_{map}(\theta) = F(G(\theta))$$

where G is a mapping from θ-space to γ-space and F is a mapping from γ-space to θ-space.

$G(\theta)$ defines the family parameter γ on θ-space.

Coupled Map Family



Learning the Coupled Map Family

- Choose $G(\theta)$ to be an affine map determined by global PCA.
- F could be any adaptive nonlinear mapping. We use a mixture of experts where each expert is an affine mapping and the mixture coefficients are Gaussians.
- Given G, F is chosen to minimize the difference between $F(G(\theta_i))$ and θ_i for each best-fit episode parameter θ_i.

Classification Task

Two classes with unit-variance normal class-conditional densities on a 5-dimensional feature space.

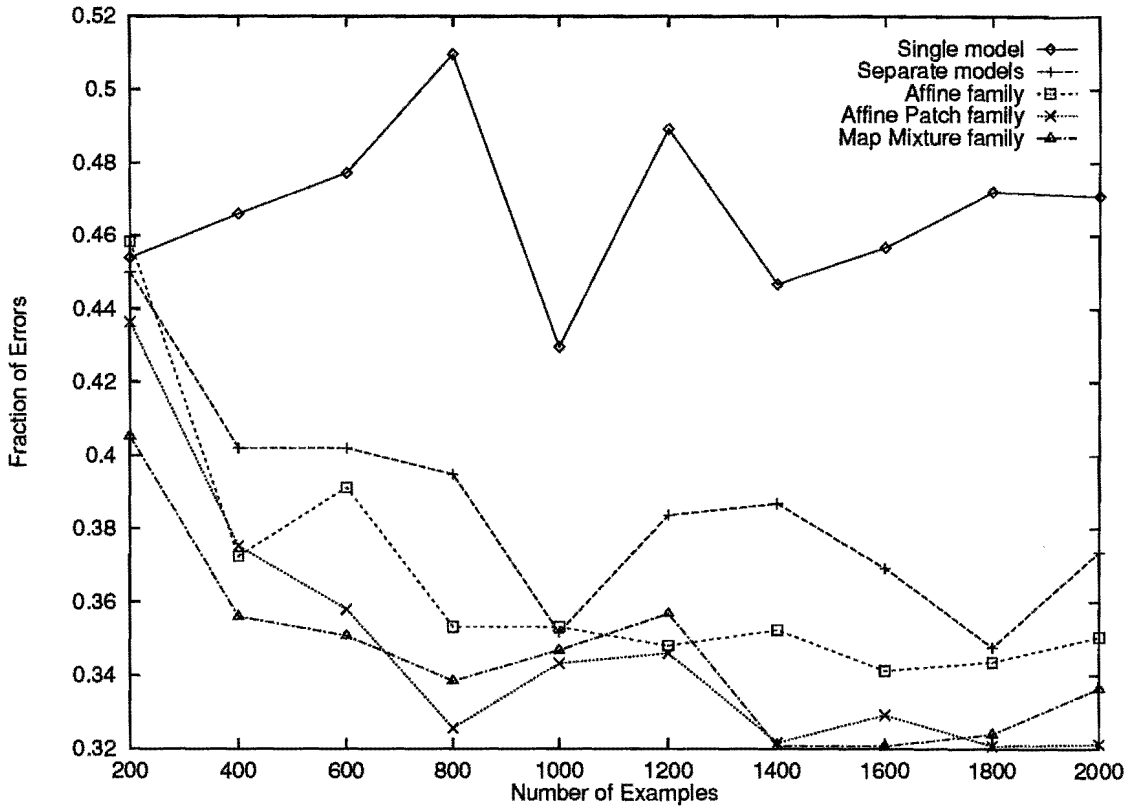
Means are parameterized by a nonlinear two-parameter family:

$$
\begin{aligned}
m_1 &= (\gamma_1 + \tfrac{1}{2}\cos\phi)\hat{e_1} + (\gamma_2 + \tfrac{1}{2}\sin\phi)\hat{e_2} \\
m_2 &= (\gamma_1 - \tfrac{1}{2}\cos\phi)\hat{e_1} + (\gamma_2 - \tfrac{1}{2}\sin\phi)\hat{e_2}.
\end{aligned}
$$

where $0 \leq \gamma_1, \gamma_2 \leq 10$ and $\phi = (\gamma_1 + \gamma_2)/3$.

For N samples, use $N = \sqrt{x}$ episodes of size $N_i = \sqrt{x}$. Choose episode parameters uniformly from the classifier family. For each episode, sample from the classifier distribution. Test set: 50 episodes of 50 examples each.

Comparison of the Five Algorithms



Training Set Discovery

- What about training sets not broken into episodes? eg. Speech recognizer not informed when switched to a new speaker. Character recognizer not informed when switched to a new font.
- Learner can sometimes use the data itself to detect episode changes.
- Temporal coherence prior: successive events are likely to come from the same model with occasional changes.
- Use EM algorithm to segment the data while fitting the models and learning the parameterization.
- Similar approach to slowly changing parameters.

Conclusions

- Family discovery is a practically important learning paradigm.
- Presented three algorithms for doing it.
- Significant performance improvement on test problem.