Nonlinear mappings for generative kernels on latent variable models

A. Carli[1], M. Bicego[1,2], S. Baldo[1], V. Murino[1,2]

[1] University of Verona (Italy)
[2] Istituto Italiano di Tecnologia (Italy)

S+SSPR SIMBAD special session: 21st August 2010

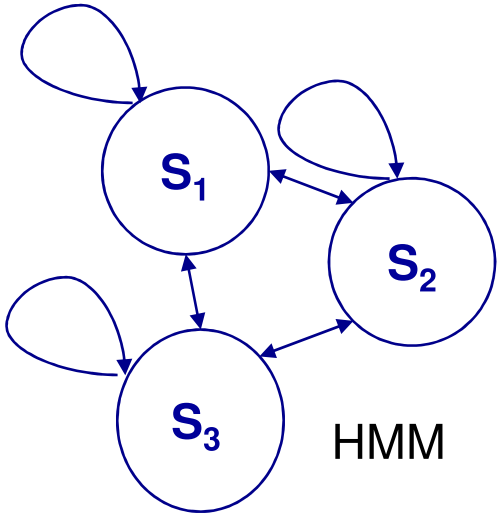
Summary

- The starting point: generative kernels
  - the generative embedding point of view
- The normalization problem
- Nonlinear normalization
- Results and findings
- Conclusions and open issues

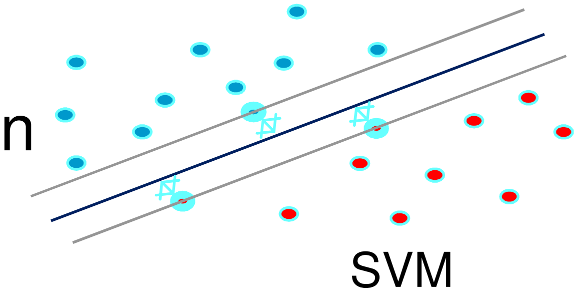
Background

- Two approaches to classification

- Generative models:
  - better description capabilities
  - ability to deal also with non vectorial (structural) representations (e.g. sequences)



- Discriminative methods:
  - typically have better classification performances



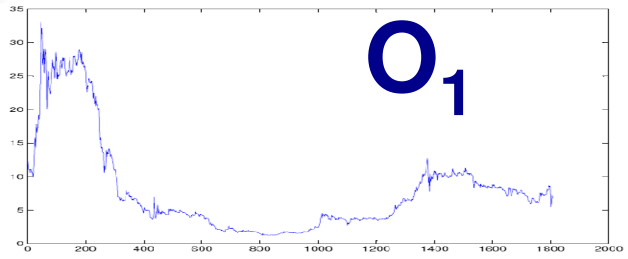
Generative kernels

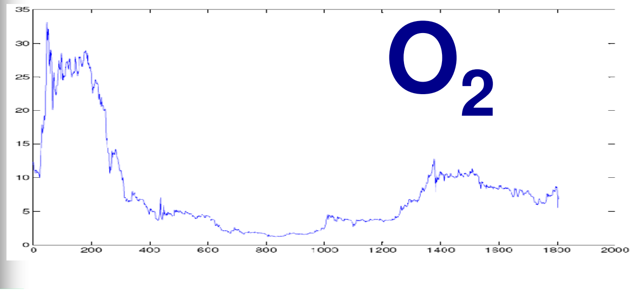
- Generative kernels are hybrid methods able to merge
  - description capabilities of generative models
  - classification skills of discriminative methods

Generative kernels

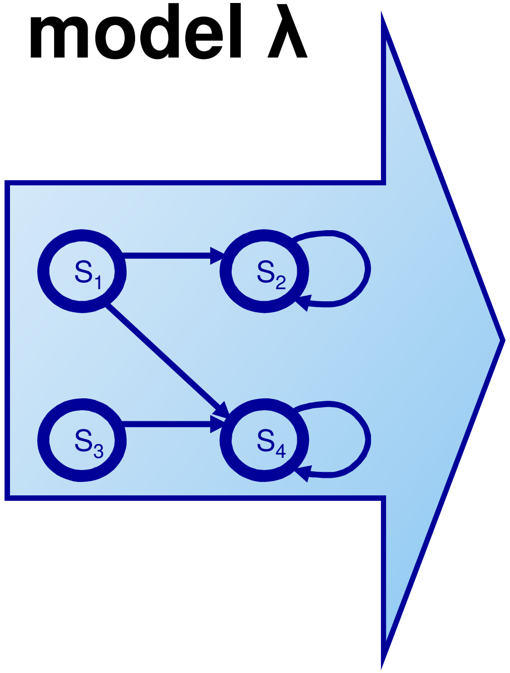
- IDEA: Exploit a generative model to compute a kernel between objects (to be used in a discriminative scenario)

Two objects





Generative model λ



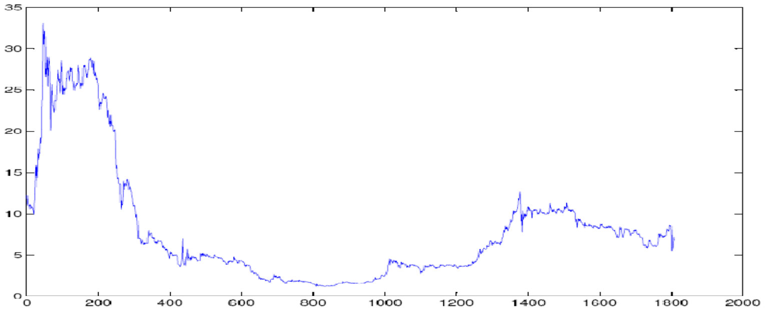
Kernel

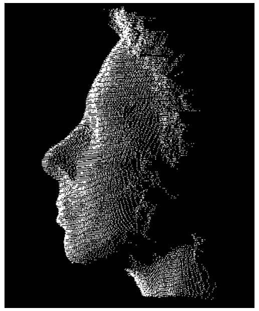
$K_\lambda(O_1,O_2)$

Generative kernels

- Main feature:
  - very suitable for structured (non vectorial) objects (sequences, graphs, sets, strings,...)

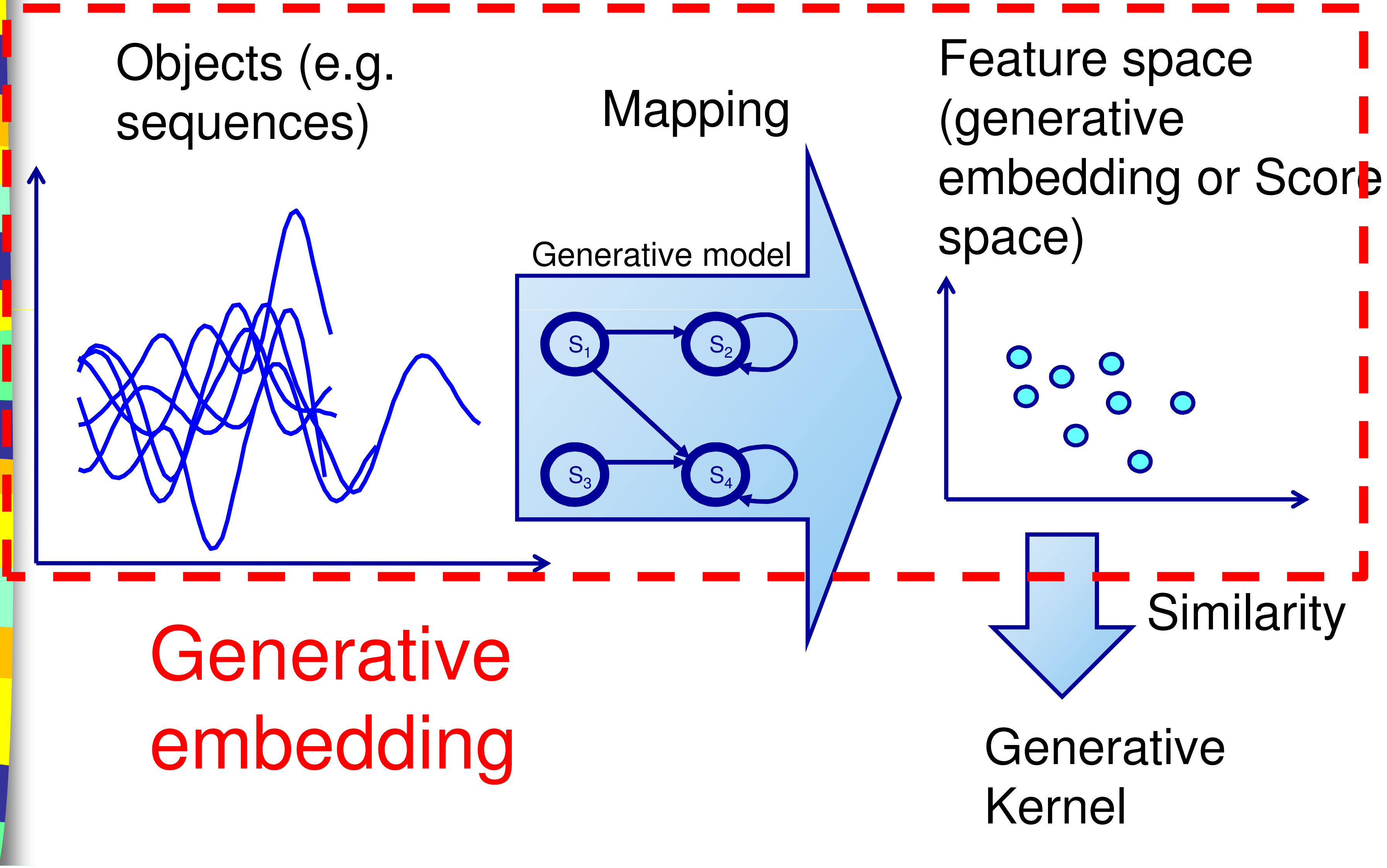
attcgatcgatcgatcgatcaggcg
cgctagagcggcgaggacctatccg





Examples: Fisher Kernel, Marginalized Kernel, KL kernel, Product Probability kernel

An alternative point of view



Objects (e.g. sequences)

Mapping

Generative model

Feature space (generative embedding or Score space)

Similarity

Generative Kernel

Generative embedding

An alternative point of view

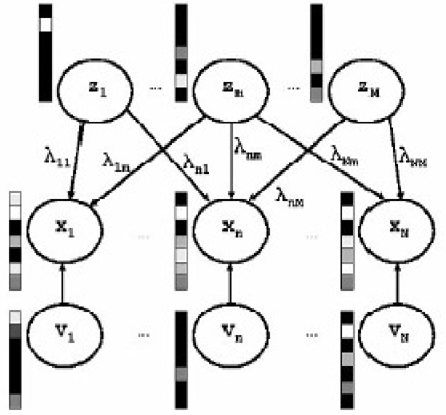
- Many generative kernels may be seen in this view

  Example: the Fisher Kernel
- The generative embedding space (called Fisher Score space)

$$\phi(O) = \nabla_{\theta} \log P(O \mid \theta)$$

- The similarity $K(O_1, O_2) = \phi(O_1) \cdot \phi(O_2)$

Generative kernels



- Different kernels may be defined depending on:
  - different generative models
  - different mappings
  - different similarities in the feature space

- HERE:
  - HMM-based generative kernels
  - the kernel is the inner product in the obtained generative embedding space

The normalization problem

- Observation: it has been shown in different cases that a proper normalization of the obtained generative embedding space is crucial
  - Fisher Score space – Smith Gales NIPS02
  - Marginalized Kernel – Tsuda et al Bioinformatics 2002
  - Other evidences: Generative embedding spaces proposed in Bicego, Pekalska, Tax, Duin, PR 09

The normalization problem (2)

- In all these cases the applied normalization is *linear*
  - e.g. standardization

$$x^{j}{}_{i}^{new} = \frac{x^{j}{}_{i} - \mu^{j}}{\sigma^{j}}$$

  - every direction *j* of the space has zero mean and unit variance

- QUESTION: may a *nonlinear* normalization be useful?

The proposed approach

- Here we try to answer to the previous question.
- **Nonlinear normalization**: apply to every component of the feature vector in the generative embedding space a *nonlinear* mapping (like powering, logarithm, logistic)
- We applied different nonlinear mappings to different HMM-based generative kernels in three applications

Details

- ***O*** is a generic object (e.g. a sequence), *λ* is the generative model (or a set of)
- Generative embedding:

$$\mathbf{O} \rightarrow [g_1(\mathbf{O},\lambda), g_2(\mathbf{O},\lambda),\ldots, g_N(\mathbf{O},\lambda)]^T$$

we assume $g_i(\boldsymbol{O},\lambda) > 0$

- Nonlinear normalization: we applied a non linear function *f* to every direction of the space

$$\mathbf{O} \rightarrow [f(g_1(\mathbf{O},\lambda)), f(g_2(\mathbf{O},\lambda)),\ldots, f(g_N(\mathbf{O},\lambda))]^T$$

Details: the nonlinear mappings

- Powering function

$$f(g_i(\mathbf{O},\lambda)) = g_i(\mathbf{O},\lambda)^{\rho} \qquad \rho > 0$$

- Natural logarithm (no parameters)

$$f(g_i(\mathbf{O},\lambda)) = \log(1 + g_i(\mathbf{O},\lambda))$$

- Logistic function

$$f(g_i(\mathbf{O},\lambda)) = \tanh\left(\rho\, g_i(\mathbf{O},\lambda)\right) \qquad 0 < \rho < 1$$

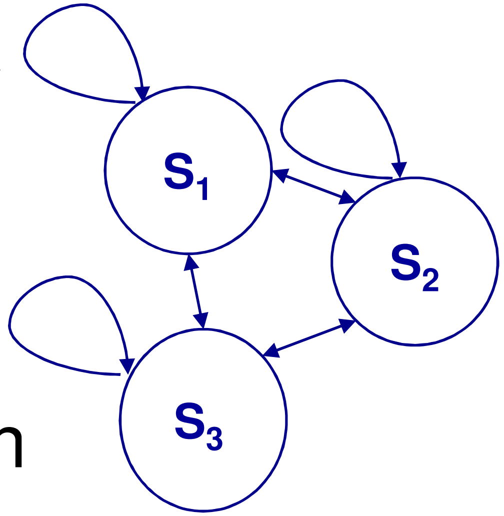
The experimental evaluation

- We tested the different nonlinear mappings with different embeddings and different applications

DETAILS

Generative model: Hidden Markov Model

- fully ergodic, trained with Baum Welch
- number of states is application dependent



The experimental evaluation

- Studied generative embeddings:
  - Fisher Score [Jaakkola et al., 1999]:

    $g_i(\boldsymbol{O},\lambda)$ is the derivative of the log likelihood of the HMM w.r.t. to a given parameter, evaluated in $\boldsymbol{O}$

  - State Space [Bicego et al., 2009]:

    $g_i(\boldsymbol{O},\lambda)$ is the averaged frequency of passing through a certain state of the HMM while observing $\boldsymbol{O}$

The experimental evaluation

- Marginalized Kernel Space [Tsuda et al., 2002]: very similar to the State Space

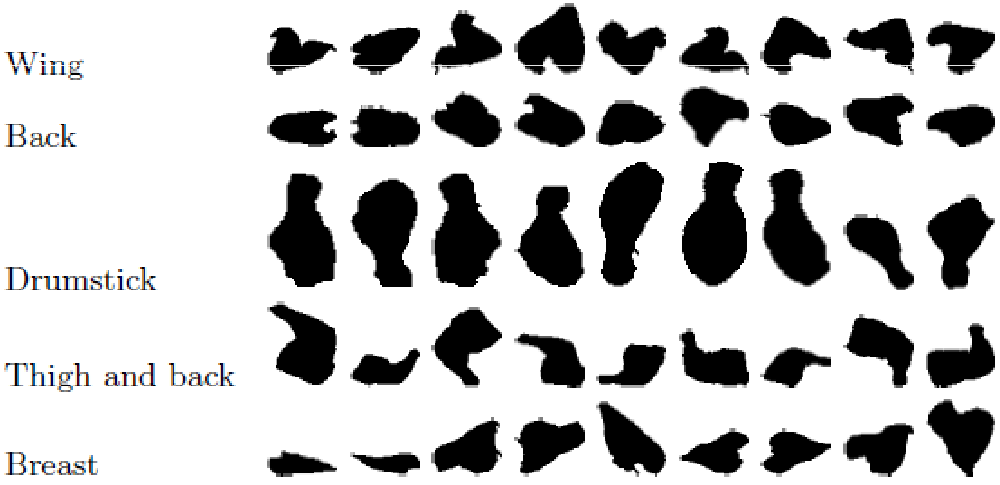
- Transition Space [Bicego et al., 2009]: *g_i(**O**,λ)* is the averaged frequency of passing through a given transition of the HMM while observing ***O***

(All details in the S+SSPR10 paper)

The experimental evaluation

- Applications:
  - 2D shape recognition using the Chicken Pieces Database

Shapes are described with chain codes (discrete HMM) and curvature (continuous Gaussian HMM)



  - gesture recognition using the AUSLAN dataset (sign language)

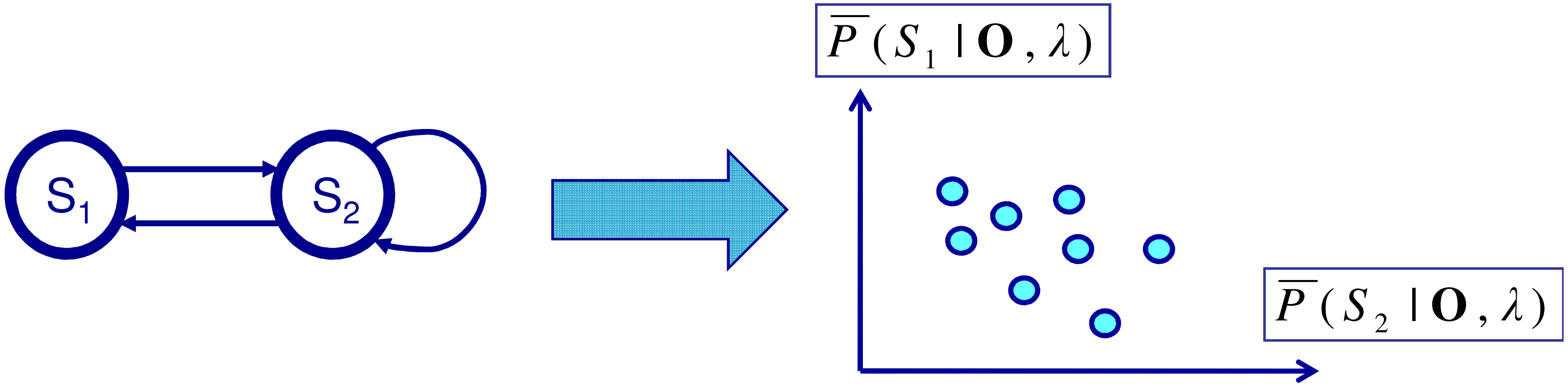
Experimental evaluation

- Classification is performed with SVM
  - the kernel: the inner product in the new space
  - C optimized with cross validation
- Accuracies computed with K-fold cross validation (results averaged over 20 repetitions)
- Different values for the parameters of nonlinear mappings (only best results are reported)

Findings: when it works (1)

Observations from the results

1. it works for generative embeddings in which each direction summarizes information related to a single HMM state
   - YES: State Space, Marginalized kernel space
   - NO: Fisher Score Space, Transitions space

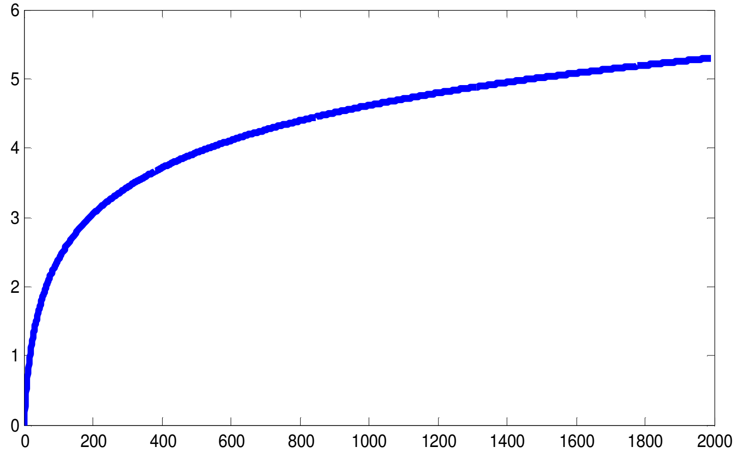

S_1
S_2
$\overline{P}(S_1 \mid \mathbf{O}, \lambda)$
$\overline{P}(S_2 \mid \mathbf{O}, \lambda)$

Findings: when it works (2)

2. It works when the nonlinear mapping has two characteristics:

- concave, with vanishing derivative at +∞

- asymptotically nonexpansive: it reduces distances, provided that $g_i(\mathbf{O},\lambda)$ are large enough



NOTE: powering with ρ>1 does not work

How it works

Classification accuracies for State Space embedding

Normalization	2D shape recognition (chain codes)	2D shape recognition (curvature)	Gesture classification
Linear	0.751	0.736	0.798
powering ($\rho<1$)	0.813	0.807	0.904
logarithm	0.753	0.755	0.838
logistic	0.770	0.780	0.826

The standard errors of the mean are all less than 0.007

How it works (2)

Classification accuracies for Marginalized kernel embedding

Normalization	2D shape recognition (chain codes)	2D shape recognition (curvature)	Gesture classification
Linear	0.775	0.767	0.533
powering (ρ<1)	0.855	0.780	0.932
logarithm	0.829	0.776	0.901
logistic	0.817	0.776	0.856

The standard errors of the mean are all less than 0.007

In some cases the improvements are impressive

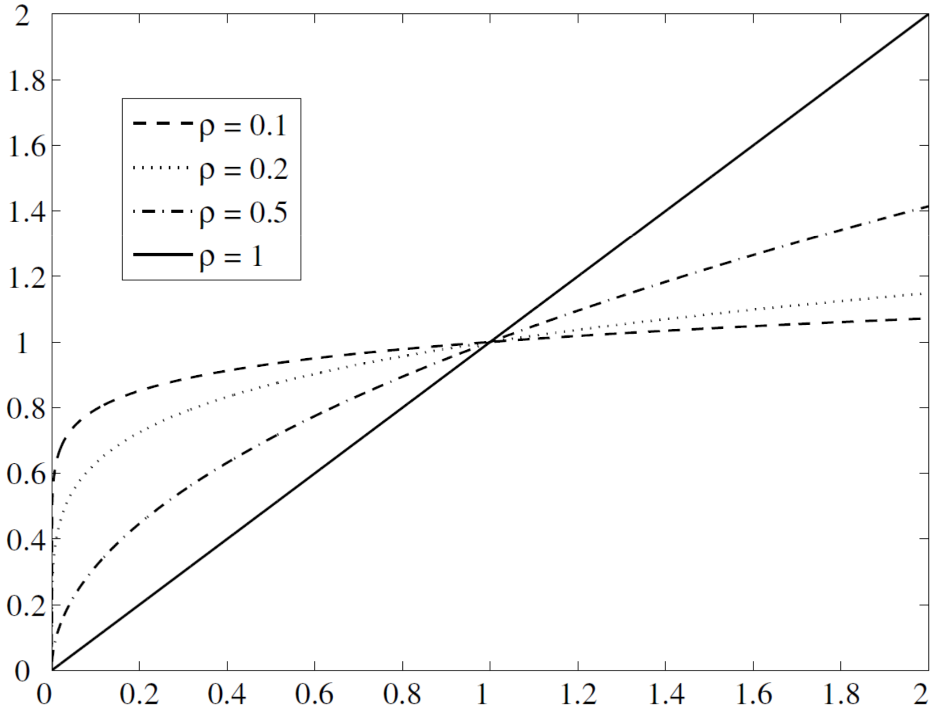
Best nonlinear mapping

- The best is the powering operation (with 0<ρ<1)

it reduces the contribution of larger components

and

it raises the contribution of smaller components



The effect is to re-equilibrate the contributions of each state of the HMM

Conclusions & future work

- Non linear normalization of generative embedding spaces may be very useful, but
  - not in all cases
  - not for all nonlinear mappings
- Why it works is still an issue
  - A direction we are investigating
    - effect of de-diagonalizing the kernel matrix (as in Schölkopf et al., ECML 2002)
- Choice of parameters is of course crucial

THANK YOU!

QUESTIONS?