Information Theoretical Kernels for Generative Embeddings Based on HMM

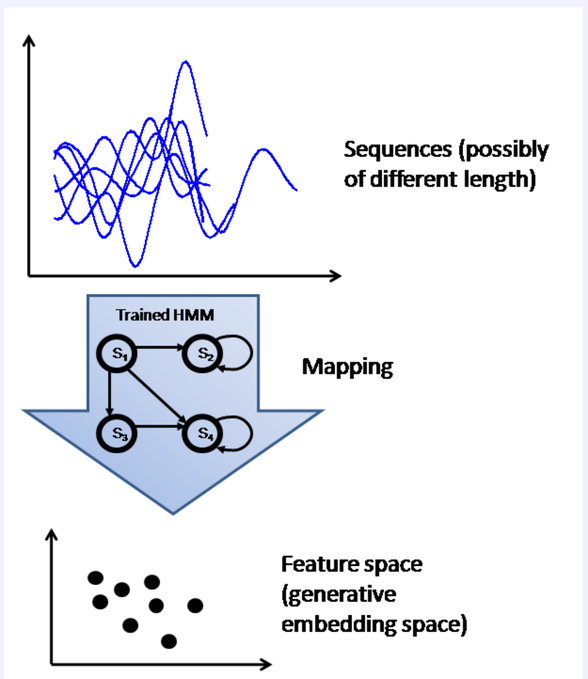
André F. T. Martins[3], Manuele Bicego[1,2], Vittorio Murino[1,2], Pedro M. Q. Aguiar[4], Mário A. T. Figueiredo[3]

[1] Computer Science Department, University of Verona - Verona, Italy – [2] Istituto Italiano di Tecnologia (IIT) - Genova, Italy
[3] Inst. de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal – [4] Inst. de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal

Motivations and Main Idea

THE CONTEXT: GENERATIVE EMBEDDINGS

- Classification of structured objects (e.g., shapes) is typically addressed with generative models (able to deal with non vectorial representations)
- Discriminative classifiers (e.g., SVM) typically outperform generative models, but prefer a vectorial representation.
- Generative embeddings represent hybrid generative-discriminative approaches, which exploit a learned generative model to map a possibly non vectorial object into a vector space, where discriminative classifiers can be used.



Using a generative embedding involves three steps:

(i) define and learn the generative model used to build the embedding;

(ii) define the mapping from object space to the generative embedding space;

(iii) discriminatively learn a (maybe kernel) classifier on the adopted feature space.

NOTE The literature on generative embeddings is essentially focused on step (i) and (ii), usually adopting some standard off-the-shelf tool (e.g., an SVM with a linear or RBF kernel) for step (iii).

THE PROPOSAL Here we follow a different route, testing the recently proposed non-extensive information theoretic kernels on several Hidden Markov Models-based generative embeddings.

The generative embeddings

Notation Components of a HMM with N states: $\boldsymbol{A} = (a_{ij})$ (the transition matrix), $\boldsymbol{\pi} = (\pi_i)$ (the initial state probability distribution), $\boldsymbol{B} = (\boldsymbol{b}_i)$ (the set of emission probability functions)

Generative embedding: it is defined by $\phi(\boldsymbol{o}, \boldsymbol{\lambda})$, which uses a trained HMM $\boldsymbol{\lambda}$ (or more than one) to map a sequence $\boldsymbol{o} = (o_1, ..., o_T)$ into a vector

1. The Classical Fisher Score Embedding (FSE) [Jaakkola *et al.* 99]

$$\phi^{\text{FSE}}(\boldsymbol{o}, \boldsymbol{\lambda}) = \left[\frac{\partial \log(P(\boldsymbol{O} = \boldsymbol{o}|\boldsymbol{\lambda})}{\partial \lambda_1}, \cdots, \frac{\partial \log(P(\boldsymbol{O} = \boldsymbol{o}|\boldsymbol{\lambda})}{\partial \lambda_L}\right]^\top$$

where λ_i represents one of the L parameters of the model $\boldsymbol{\lambda}$

2. The Marginalized Kernel Embedding (MKE) [Tsuda *et al.* 02]

$$\phi^{\text{MKE}}(\boldsymbol{o}, \boldsymbol{\lambda}) = [m_{si}(\boldsymbol{o}, \boldsymbol{\lambda})], \forall s = 1..S, i = 1..N$$

where

$$m_{si}(\boldsymbol{o}, \boldsymbol{\lambda}) = \frac{1}{T} \sum_{\boldsymbol{q} \in \{1,...,N\}^T} P(\boldsymbol{Q} = \boldsymbol{q}|\boldsymbol{O} = \boldsymbol{o}, \boldsymbol{\lambda}) \sum_{t=1}^{T} I(o_t = s \wedge q_t = i),$$

3. The State Space Embedding (SSE) [Bicego *et al.* 09]

$$\phi^{\text{SSE}}(\boldsymbol{o}, \boldsymbol{\lambda}) = \left[\sum_{t=1}^{T} P(Q_t = 1|\boldsymbol{o}, \boldsymbol{\lambda}), \cdots, \sum_{t=1}^{T} P(Q_t = N|\boldsymbol{o}, \boldsymbol{\lambda})\right]^\top$$

4. The Transition Embedding (TE) [Bicego *et al.* 09]

$$\phi^{\text{TE}}(\mathbf{O}, \boldsymbol{\lambda}) = \begin{bmatrix} \sum_{t=1}^{T-1} P(Q_t = 1, Q_{t+1} = 1|\boldsymbol{o}, \boldsymbol{\lambda}) \\ \sum_{t=1}^{T-1} P(Q_t = 1, Q_{t+1} = 2|\boldsymbol{o}, \boldsymbol{\lambda}) \\ \vdots \\ \sum_{t=1}^{T-1} P(Q_t = N, Q_{t+1} = N|\boldsymbol{o}, \boldsymbol{\lambda}) \end{bmatrix}$$

The information Theoretic Kernels

Given two probability measures p_1 and p_2, representing two objects, we tested several information theoretic kernels (ITKs) [Martins *et al.* 09]:

1. k^{JS} – Jensen-Shannon kernel:

$$k^{\text{JS}}(p_1, p_2) = \ln(2) - JS(p_1, p_2),$$

with $JS(p_1, p_2)$ being the Jensen-Shannon divergence

$$JS(p_1, p_2) = H\left(\frac{p_1 + p_2}{2}\right) - \frac{H(p_1) + H(p_2)}{2},$$

$H(p)$ is the usual Shannon entropy.

2. k_q^{JT} – Jensen-Tsallis (JT) kernel:

$$k_q^{\text{JT}}(p_1, p_2) = \ln_q(2) - T_q(p_1, p_2),$$

where $\ln_q(x) = (x^{1-q} - 1)/(1 - q)$ is the q-logarithm,

$$T_q(p_1, p_2) = S_q\left(\frac{p_1 + p_2}{2}\right) - \frac{S_q(p_1) + S_q(p_2)}{2^q}$$

is the Jensen-Tsallis q-difference, and $S_q(r)$ is the Jensen-Tsallis entropy, defined, for a multinomial $r = (r_1, ..., r_L)$, with $r_i \geq 0$ and $\sum_i r_i = 1$, as

$$S_q(r_1, ..., r_L) = \frac{1}{q - 1}\left(1 - \sum_{i=1}^{L} r_i^q\right).$$

3. k_q^A and k_q^B two versions of the Jensen-Tsallis kernel applicable to unnormalized measures (see the paper for more details)

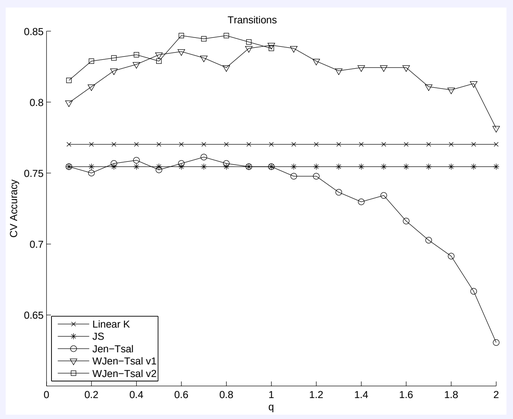
Experimental Evaluation

Details

- Tests on a 2D shape recognition task (shapes are characterized with a sequence of curvature values), with the Chicken Pieces Database (446 silhouettes of chicken pieces - 5 classes). Accuracies computed with averaged hold out CV (10 repetitions)
- 3-state HMMs with Gaussian emission densities
- SVM with IT kernels on generative embeddings
- C of SVMs and q of the information theoretic kernels were optimized by 10-fold cross validation (CV) on the training set

Results

Emb.	Linear	$k^{\text{JS}} = k_1^{\text{JT}}$	k_q^{JT}	k_q^A	k_q^B
ϕ^{SSE}	0.7387	0.7230	0.7095	0.7995	0.8221
ϕ^{SSE} (S)	0.7342	0.7230	0.7005	0.8086	0.7950
ϕ^{TE}	0.7703	0.7545	0.7545	0.8243	**0.8356**
ϕ^{TE} (S)	0.8311	0.7995	0.7973	0.8176	0.8198
ϕ^{FSE}	0.6171	0.6194	0.6261	0.7568	0.6689
ϕ^{FSE} (S)	0.8108	0.8243	0.8243	**0.8311**	0.8243
ϕ^{MKE}	0.6712	0.7095	0.7455	0.8243	0.8063
ϕ^{MKE} (S)	0.7477	0.6937	0.7162	0.7995	0.8063



(Left) Classification accuracies. "(S)" refer to experiments where the embeddings were standardized (centered and scaled to unit variance). (Right) SVM accuracies with several kernels for the Transition Embedding, as a function of q.

Comparative Analysis

Methodology	Accuracy (%)	Reference
1-NN + Levenshtein edit distance	≈ 0.67	[18]
1-NN + approximated cyclic distance	≈ 0.78	[18]
KNN + cyclic string edit distance	0.743	[19]
SVM + Edit distance-based kernel	0.811	[19]
1-NN + mBm-based features	0.765	[6]
1-NN + HMM-based distance	0.737	[6]
SVM + HMM-based entropic features	0.812	[21]
SVM + HMM-based Top Kernel	0.808	[22]
SVM + HMM-based FESS embedding + rbf	0.830	[22]
SVM + HMM-based non linear Marginalized Kernel	0.855	[8]
SVM + HMM-based clustered Fisher kernel	0.858	[3]

Comparative Results on the *Chicken* data.

Acknowledgements: We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (Contract 213250) and from Fundação para a Ciência e Tecnologia (FCT) (grant PTDC/EEA-TEL/72572/2006).