Non-linear Generative Embeddings for Kernels on Latent Variable Models

Anna Carli [1], Manuele Bicego [1,2], Sisto Baldo [1], Vittorio Murino [1,2]
[1] Dept. of Computer Science, University of Verona, Strada le Grazie 15 – 37134 Verona, Italy
[2] Istituto Italiano di Tecnologia (IIT), Via Morego 30 – 16163 Genova, Italy
anna.carli|manuele.bicego|sisto.baldo|vittorio.murino|@univr.it

Abstract

Generative embeddings use generative probabilistic models to project objects into a vectorial space of reduced dimensionality – where the so-called generative kernels can be defined. Some of these approaches employ generative models on latent variables to project objects into a feature space where the dimensions are related to the latent variables. Here, we propose to enhance the discriminative power of such spaces by performing a non-linear mapping of space dimensions leading to the formulation of novel generative kernels. In this paper, we investigate one possible non-linear mapping, based on a powering operation, able to equilibrate the contributions of each latent variable of the model, thus augmenting the entropy of the latent variables vectors. The validity of the idea has been shown in the case of two generative kernels, which have been evaluated with tests on shape recognition and gesture classification, with really satisfying results that outperform state-of-the-art methods.

1. Introduction

Subspace methods [24, 14] allow to embed objects living in the problem space into a vectorial space of limited dimensionality. In the case when the original problem space is a vectorial space, many techniques have been proposed in the literature, like Principal Component Analysis, Independent Component Analysis, Non-Negative Matrix Factorization and others [21], each one characterized by different characteristics, like linearity, optimized criteria, computational effectiveness, and others. In some problems, the original space is not a vectorial space, for example when the objects have a structural form, like sequences (of different length), graphs, sets, and the like. In this case, the reduced vectorial space may be obtained by a different class of approaches, generally called embeddings.

Embeddings may be performed in different ways: the most used is to compute pairwise proximity measures between the objects and to create a vector space where these proximities are preserved. One widely applied example in this class is the so-called Multidimensional Scaling [22], and several generalizations have been proposed (*e.g.* Laplacian embeddings [2]). Another class of such approaches, whose interest has drastically grown in the last years, is represented by the so-called generative embeddings [4, 5, 19]. In this class, the idea is to employ a generative probabilistic model (like Hidden Markov Models or Bayesian Networks) to encode and model the structural information of the objects; then, a vector space is obtained by using features extracted from this model. The dimensionality of the resulting space is drastically reduced if compared with that of the original input space. When a classification problem is addressed, an effective solution is to define a similarity measure between points in this novel space, leading to the definition of a kernel between the original objects (called generative kernel [8, 9, 23]), to be employed in discriminative kernel-based classifiers like Support Vector Machines (SVMs).

Generative embeddings allow to mix generative methods (like Hidden Markov Models or Bayesian Networks) and discriminative methods (like SVMs), merging the description capabilities of the former class of approaches with the discriminative skills of the latter class [13].

Different generative kernels can be built starting from different vectorial spaces obtained through generative embeddings. Depending on which (dis)similarity measure between distributions is used, one obtains the Fisher Kernel [8], the Probability Product Kernel and the Bhattacharyya affinity kernel [9], and the Marginalized Kernel [23], among the others.

In this paper, we focus on a particular class of generative embeddings, namely embeddings defined on generative models with latent variables (for example, the states in a Hidden Markov Model), leading to generative kernels defined as inner product on the resulting vectorial space. A famous example of such kernels is the Marginalized Kernel [23] – even if in the original paper an explicit definition of the space and a derivation of the kernel as inner product are missing. Very recently, another kernel has been proposed

2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops
978-1-4244-4441-0/09/$25.00 ©2009 IEEE

to be used with Hidden Markov Models, called State-Space Kernel [5]. The idea at the basis of generative embeddings for this class of kernels is to map the objects of the problem in a space where each dimension (or a set of dimensions) describes the contribution of one of the latent variables of the model. For example, in the State-Space each direction represents the averaged probability of being in a particular state given the model and the observation. The inner product in such generative-derived space represents the kernel. Even if the success of these approaches has been proven on many applications there is still room for improvement. Actually, the different directions of the derived generative space (which are related to latent variables) could have different characteristics in terms of discriminative and descriptive power. These characteristics could not be completely highlighted by a simple inner product, and some space transformations may be useful. Actually, a space normalization step (centering and scaling) has been proposed in [20] that leads to an improved version of the well-known Fisher Kernel. Moreover, it should also be noted that the Marginalized Kernel, in its original formulation [23], needed a rescaling of the Kernel Matrix (centering and division by the Frobenius norm) for working properly. The common characteristic of these space transformations is however the linearity of the scaling function. Nevertheless, there are situations where the linearity assumption is too restrictive, and a benefit may be obtained from a non-linear scaling via a non-linear mapping.

In this paper, we propose to investigate the latter alternative, proposing a non-linear transformation of the original vectorial space, obtained through generative embedding, into another vectorial space, namely a non-linear mapping of space dimensions able to highlight or exploit their discriminative characteristics. New kernels are then defined as inner products on the transformed space: the specific form of such kernels depends on the choice of the non-linear mapping and on the latent variable model it relies upon. Diverse non-linear mappings are indeed possible, and we propose one possible, very simple choice of a non-linear mapping, able to balance the contributions of each latent variable of the model, thus augmenting the entropy of the latent variables vectors. The basic tool is a powering operation, able to equilibrate the contributions of each latent variable. We apply this idea to the Marginalized Kernel and to the State-Space Kernel, giving the kernel formulation in closed form for a HMM generative model.

The effectiveness of the proposed non-linear mapping has been evaluated in a classification framework, comparing the performance of an SVM classifier based on kernels defined on the transformed space to its standard counterpart based on kernels defined on the original space. Two different sequence classification problems (2-D shape recognition and gesture recognition) are addressed with really satisfying results that outperform those presented in the literature reaching a mean classification accuracy of 85.52%, largely above the best results in the state of the art, for the 2-D shape recognition problem, and a mean classification accuracy of 93.24% for the gesture classification problem using raw sequences without extracting quite elaborated features. Moreover, it is important to note that the best results are obtained projecting the input objects in vectorial spaces of very low dimensionality, since the number of HMM states ranges from 3 to 8.

The remainder of this paper is organized as follows. In Section 2, we first analyze the problem in detail and the proposed methodology is described. In Section 3, the specific formulation in the case of HMM-based generative embeddings is presented. Section 4 presents the experimental results, while Section 5 concludes the paper with a discussion of future research.

2. The proposed methodology

We consider a particular class of generative embeddings relying on latent variables and propose a non-linear transformation of the resulting vectorial space into another one, where a more discriminative similarity measure can be defined. Given an object x in the input space $\mathcal{X}$, the generative embedding can be based on model components, i.e. latent variables. The objects x are then projected into a vectorial space of reduced dimensionality. Features of the resulting space $\mathcal{H}$ summarize information about how latent variables describe the observation x. We will call this information in the resulting space as $\boldsymbol{g_h}$ – the pedix $\boldsymbol{h}$ highlights the dependence of such information from the latent variables $\boldsymbol{h}$ – where $\boldsymbol{h} = \{h_1, \ldots, h_N\}$ denotes the set of hidden variables of generative latent model.

A kernel can be defined on the resulting space as a sum of inner products, each one related to a specific latent variable:

$$K\left(x, x'\right) = \left\langle \boldsymbol{g_h}\left(x\right), \boldsymbol{g_h}\left(x'\right)\right\rangle = \sum_{i=1}^{N} \left\langle \boldsymbol{g}_{h_i}\left(x\right), \boldsymbol{g}_{h_i}\left(x'\right)\right\rangle \tag{1}$$

where $\boldsymbol{g}_{h_i}\left(x\right) = \left[g_{h_i}^1\left(x\right), g_{h_i}^2\left(x\right), \ldots, g_{h_i}^S\left(x\right)\right]$ denotes a vector of features related to a particular hidden variable h_i. In other words, we are grouping together in the vector $\boldsymbol{g}_{h_i}$ all the directions $1, \ldots, S$ of the feature space relative to a particular latent variable h_i (all these grouped directions may for example be derived from other quantities of the model).

This formulation can in principle be applied to any generative model with latent variables which are used to form features. Examples of such kernels are the well-known Marginalized Kernel [23] and the recently proposed State-Space Kernel [5]. In the former case $\boldsymbol{g}_{h_i}$ is a vector of length equal to the number of symbols in the alphabet, while in the second case $\boldsymbol{g}_{h_i}$ is a single scalar value.

Even if their success has been proven on many applications there is still room for improvement. Actually, the different directions of the derived generative space could have different characteristics in terms of discriminative and descriptive power. These characteristics cannot be completely highlighted by a simple inner product, and some space transformations may be useful. Here we investigate the possibility of employing a non-linear scaling of space dimensions able to highlight their discriminative characteristics. This results in performing a non-linear mapping f of dimensions of the original Hilbert space: $f\left(\boldsymbol{g}_{h_i}\right)$, $\forall i = 1, \ldots, N$. The kernel formulation as inner product on this transformed space is then defined as:

$$NK\left(x, x'\right) = \sum_{i=1}^{N} \langle f\left(\boldsymbol{g}_{h_i}\left(x\right)\right), f\left(\boldsymbol{g}_{h_i}\left(x'\right)\right)\rangle \qquad (2)$$

NK represents an inner product in a new space whose dimensions are obtained from dimensions of the original latent variable space through a non-linear mapping f. If well designed, non-linear mapping can unravel hidden structures and balance for not equally important directions. In this paper we propose a non-linear mapping able to enhance the expressiveness of the kernel. In particular we adopt the following function f[1]:

$$f\left(\boldsymbol{g}_{h_i}\left(x\right)\right) = \left(\boldsymbol{g}_{h_i}\left(x\right)\right)^{\rho} \qquad \forall i = 1, \ldots, N \qquad (3)$$

where ρ is a positive real value. We notice that for $\rho = 1$ the original kernel is reobtained. The choice of ρ is clearly crucial. Different values assume different significance. In this study we do not propose any systematic method for the choice of the value ρ, considering it as a free parameter of the kernel (we performed an experimental study on finding the best value of ρ – see Section 4.4). In any case we assume $\rho \leq 1$, since this choice has some appealing characteristics. Actually, the effect of such powering is to raise the contribution of smaller components of $\boldsymbol{g}_{h_i}$ and to reduce the contribution of larger components (see Fig. 1), thus re-equilibrating the contributions of each latent variable. This may be seen as a way of augmenting the entropy of the contributions of latent variables. On the contrary, assuming $\rho > 1$ may have the opposite behavior, sparsifying the contributions of the latent variables.

A final remark: it should be noted that the powering operation is not new in the kernel scenario, even if our use is innovative. The most famous example is the polynomial kernel [18] where the powering of the inner product is considered – while here we propose the powering of each single component of the vectors involved in the inner products. Further, another example can be found in [9], where the definition of the so-called Probability Product Kernel is given – which implies the powering of probability products. Also in this case there is a remarkable difference, since the definition in [9] relies on powering the components of an integral over the observation space of two known probability distributions – whereas our approach considers the integration (summation) in the latent variable space, namely the integration is over the model components.

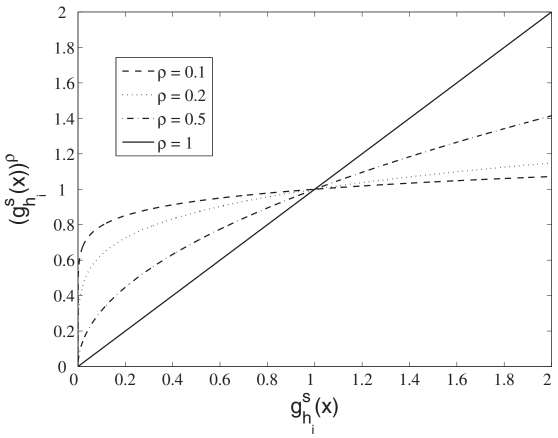

Figure 1. Shape of the $f\left(g^s_{h_i}\right) = \left(g^s_{h_i}\right)^{\rho}$ function when $\rho < 1$.

3. Non-linear mapping of HMM-based generative embeddings

In this section, we will provide details about the non-linear mapping of spaces obtained through generative embeddings based on a HMM modeling of the input data. The counterparts of two kernels, namely the Marginalized Kernel and the State-Space Kernel, will be defined in these transformed spaces.

Some basics about HMM, Marginalized Kernel and State-Space Kernel will first be reviewed, mainly to fix the notation.

3.1. Basics

3.1.1 Hidden Markov Models

A discrete-time first order Hidden Markov Model [16] is a stochastic finite state machine defined over a set of N states $\boldsymbol{h} = \{h_1, h_2, \ldots, h_N\}$. The states are hidden, i.e. not directly observable. Each state has an associated probability density function encoding the probability of observing a certain symbol being output from that state. Let $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ be a fixed state sequence of length T with the corresponding observations $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$. A HMM is described by a model $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ where $\boldsymbol{A} = (a_{ij})$ is a matrix of transition probabilities, in which $a_{ij} = P(q_t = h_j \mid q_{t-1} = h_i)$ denotes the probability of state h_j following state h_i, $\boldsymbol{B} = (b_j(s))$ consists of emission

[1] If $\boldsymbol{g}_{h_i}$ is a vector, we consider the element-wise powering operation.

probabilities, in which $b_j(s) = P(x_t = s \,|\, q_t = h_j)$ is the probability of emitting the symbol s when being in state h_j, and $\boldsymbol{\pi} = (\pi_i)$ is the initial state probability distribution, i.e. $\pi_i = P(q_1 = h_i)$.

A crucial procedure is the so-called *forward-backward* procedure [16], used to recursively compute the probability $P(\boldsymbol{x}|\boldsymbol{\lambda})$ for a test sequence $\boldsymbol{x}$, i.e. the probability of generating $\boldsymbol{x}$ by model $\boldsymbol{\lambda}$. This algorithm is used multiple times to derive the quantities needed in our proposal.

3.1.2 Marginalized Kernel

The Marginalized Kernel was defined for discrete HMM in [23], without explicitly relying on a vectorial space, as:

$$MK(\boldsymbol{x}, \boldsymbol{x}') = \sum_{s=1}^{S} \sum_{i=1}^{N} \gamma_{sh_i}(\boldsymbol{x})\, \gamma_{sh_i}(\boldsymbol{x}') \tag{4}$$

with

$$\gamma_{sh_i}(\boldsymbol{x}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{q_t=h_1}^{h_N} P(q_t|\boldsymbol{x})\, I(x_t = s, q_t = h_i) \tag{5}$$

where $\{1, \ldots, S\}$ is the discrete alphabet of emitted symbols and the indicator function $I(\alpha = \bar{\alpha})$ is 1 if the condition $\alpha = \bar{\alpha}$ is true, 0 otherwise.

3.1.3 State-Space Kernel

In [5], a finite-dimensional space is derived from the latent variables of a generative HMM trained on data. This HMM-induced vector space, called *State-Space* is equipped with the traditional Euclidean metric. Here we define an inner product on this space and take it as a similarity measure (kernel) between sequences $\boldsymbol{x}$ and $\boldsymbol{x}'$. Given one trained HMM, the kernel is defined as:

$$\begin{aligned} SK(\boldsymbol{x}, \boldsymbol{x}') \;=\; & \sum_{i=1}^{N} \left(\sum_{t=1}^{T} P(q_t = h_i|\boldsymbol{x}, \boldsymbol{\lambda}) \right) \\ & \cdot \left(\sum_{t'=1}^{T'} P(q_{t'} = h_i|\boldsymbol{x}', \boldsymbol{\lambda}) \right) \end{aligned} \tag{6}$$

3.2. Non-linear mapping of generative embeddings

In Section 2, we have explicitly defined the vectorial spaces on which generative kernels based on latent variables, like Marginalized Kernel and State-Space Kernel, rely. In such a way we have proposed an unified notation for them that could also be used for other kernels on latent variables. Indeed, it is straightforward to show that they adhere to our general formulation in Eq. (1) by rewriting the Marginalized Kernel defined in Eq. (4) as:

$$\begin{aligned} MK(\boldsymbol{x}, \boldsymbol{x}') = & \sum_{i=1}^{N} \sum_{s=1}^{S} \left(\frac{1}{T} \sum_{t=1}^{T} P(q_t = h_i|\boldsymbol{x}) \cdot I(x_t = s) \right) \\ & \cdot \left(\frac{1}{T'} \sum_{t'=1}^{T'} P(q_{t'} = h_i|\boldsymbol{x}') \cdot I(x_{t'} = s) \right) \end{aligned} \tag{7}$$

This is exactly in the form of Eq. (1), with

$$\boldsymbol{g}_{h_i}(\boldsymbol{x}) = [g^1_{h_i}(\boldsymbol{x}), g^2_{h_i}(\boldsymbol{x}), \ldots, g^S_{h_i}(\boldsymbol{x})] \tag{8}$$

and

$$g^s_{h_i}(\boldsymbol{x}) = \frac{1}{T} \sum_{t=1}^{T} P(q_t = h_i|\boldsymbol{x}) \cdot I(x_t = s) \tag{9}$$

For the State-Space Kernel, $\boldsymbol{g}_{h_i}(\boldsymbol{x})$ contains only one component $g^1_{h_i}(\boldsymbol{x})$, which is defined as:

$$g^1_{h_i}(\boldsymbol{x}) = \sum_{t=1}^{T} P(q_t = h_i|\boldsymbol{x}) \tag{10}$$

More generally, we can write

$$g^s_{h_i}(\boldsymbol{x}) = \sum_{t=1}^{T} \eta^s_{th_i}(\boldsymbol{x}) \tag{11}$$

with

$$\eta^s_{th_i}(\boldsymbol{x}) = \frac{1}{T} P(q_t = h_i|\boldsymbol{x}) \cdot I(x_t = s) \tag{12}$$

for the Marginalized Kernel and

$$\eta^1_{th_i}(\boldsymbol{x}) = P(q_t = h_i|\boldsymbol{x}) \tag{13}$$

for the State-Space Kernel.

The general formulation in this case is then

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{x}') &= \langle \boldsymbol{g_h}(\boldsymbol{x}), \boldsymbol{g_h}(\boldsymbol{x}') \rangle \\ &= \sum_{i=1}^{N} \sum_{s=1}^{S} g^s_{h_i}(\boldsymbol{x}) \cdot g^s_{h_i}(\boldsymbol{x}') \\ &= \sum_{i=1}^{N} \sum_{s=1}^{S} \sum_{t=1}^{T} \eta^s_{th_i}(\boldsymbol{x}) \cdot \sum_{t'=1}^{T'} \eta^s_{t'h_i}(\boldsymbol{x}') \end{aligned} \tag{14}$$

When applying the non-linear mapping proposed in Eq. (3), we obtain

$$\begin{aligned} NK(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{i=1}^{N} \sum_{s=1}^{S} \left(g^s_{h_i}(\boldsymbol{x}) \right)^{\rho} \cdot \left(g^s_{h_i}(\boldsymbol{x}') \right)^{\rho} \\ &= \sum_{i=1}^{N} \sum_{s=1}^{S} \left(\sum_{t=1}^{T} \eta^s_{th_i}(\boldsymbol{x}) \right)^{\rho} \cdot \left(\sum_{t'=1}^{T'} \eta^s_{t'h_i}(\boldsymbol{x}') \right)^{\rho} \end{aligned} \tag{15}$$

Following [9], we also test the following variant:

$$\widetilde{NK}\left(\boldsymbol{x},\boldsymbol{x}'\right) = \sum_{i=1}^{N}\sum_{s=1}^{S}\sum_{t=1}^{T}\left(\eta^{s}_{th_i}\left(\boldsymbol{x}\right)\right)^{\rho} \cdot \sum_{t'=1}^{T'}\left(\eta^{s}_{t'h_i}\left(\boldsymbol{x}'\right)\right)^{\rho} \tag{16}$$

This is exactly the L^{ρ}-quasinorm sometimes introduced in Functional Analysis for L^{ρ} spaces with $0 < \rho < 1$ [17]. We will show in the experimental part that in some cases this modification may improve the performance of the method.

A final remark: the generative embeddings presented in this section assume one HMM modeling the whole problem. Clearly, in a C-class problem, more information may be extracted if more than one generative model is established, each one representing one single class (see [7] in the Fisher Kernel case). Here we adopt the generalization proposed in [7]: one HMM per class is built, and the final kernel is then defined as the inner product in the space obtained as cartesian product of the spaces resulting from each model (namely concatenating all the spaces of all models). More in detail, given a C-class problem, C HMMs $\boldsymbol{\lambda}_c$, where $c = 1, \ldots, C$, are trained. Then all the kernels are re-formulated by adding an external summation over the models; for example, Eq. (14) becomes

$$\begin{aligned} K\left(\boldsymbol{x},\boldsymbol{x}'\right) &= \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{s=1}^{S} g^{s}_{h_{ic}}\left(\boldsymbol{x},\boldsymbol{\lambda}_c\right) \cdot g^{s}_{h_{ic}}\left(\boldsymbol{x}',\boldsymbol{\lambda}_c\right) \\ &= \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{s=1}^{S}\sum_{t=1}^{T} \eta^{s}_{th_{ic}}\left(\boldsymbol{x},\boldsymbol{\lambda}_c\right) \cdot \sum_{t'=1}^{T'} \eta^{s}_{t'h_{ic}}\left(\boldsymbol{x}',\boldsymbol{\lambda}_c\right) \end{aligned} \tag{17}$$

where h_{ic} denotes the i^{th} latent variable of the model trained on the c^{th} class.

4. Experiments

In this section, the HMM-based non-linear generative embeddings proposed in the previous sections are evaluated in a Support Vector Machine (SVM) classification framework based on kernels defined on the transformed space, and its performances are compared with their standard counterpart based on kernels defined on the original space.

We addressed two different applications involving sequences, namely, 2-D shape recognition and gesture recognition.

Data sets and implementation issues are described in detail in Sections 4.1 and 4.2, respectively. In Sections 4.3 and 4.4 a discussion on results and on the choice of the parameter ρ is provided.

4.1. Data sets

4.1.1 2-D shape recognition

Here we choose to study the Chicken Pieces Database, denoted also as *Chicken* data [1]. This set consists of 446 binary images of chicken pieces (with five classes). The shapes are usually first described by contours, which are further encoded by suitable sequences. This poses a difficult classification task.

In our experiments, two different sequence representations are used to model contours, chain codes and curvature angles. In the first case, a standard 8-direction chain encoding procedure is applied to each image. Then, discrete HMMs are used to model these classes of symbol sequences. In the second case, we derive curvature sequences as in [3, 12]. Classes of curvature sequences are finally modeled by continuous Gaussian HMMs.

The original set is split into the training and test sets, in the ratio of $50\% - 50\%$. The classification runs are averaged over 20 hold-out experiments.

4.1.2 Gesture recognition

We study here high-quality recordings of Australian sign language signs. This data set consists of samples of Australian signs [10]. We will denote it *Auslan* data. Samples from a single native signer were collected over a period of nine weeks, using high-quality position trackers and instrumented gloves (resulting in 22-D observations). 27 samples per sign were collected, the average recording length of each sign is approximately 57 frames. In the reference paper [10], two different scenarios are considered: (1) 95 sign-classes, with 2565 signs in total, and (2) 10 sign-classes. We follow the second scenario here, *i.e.* $C = 10$.

Continuous Gaussian HMMs are employed, directly modeling the signals acquired from the sensors. In order to get comparable results to the ones presented in [10], the performance of our classification schemes is computed by using 20 repetitions of a 5-fold cross-validation.

4.2. Experimental details

In all our experiments we assume fully ergodic HMMs. HMMs training has been performed using Baum-Welch reestimation procedure, stopping it at the likelihood convergence. Initialization is random both for the transition probabilities and initial state probabilities. In case of continuous signals, the emission probability models are initialized by a Gaussian Mixture clustering. In case of discrete symbol sequences, 20 independent training runs are performed, starting from a random initialization, picking the best likelihood model as the representative.

The number of states is fixed for all classes in each problem. Different numbers of states were tried, with results reported. In particular a number of states ranging from 3 to 8 were tested for *Chicken* data and for *Auslan* data. The HMM implementation relies on the Murphy's Hidden Markov Model Toolbox for Matlab [2]. The SVM classifier,

[2] http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

as implemented in LibSvm [3], is used to estimate classification accuracy for each problem.

The proposed non-linear mapping has been applied to State-Space Kernel and to Marginalized Kernel[4], both in the original and in the modified formulation of Eqs. (15) and (16). Different values of ρ, in the range $(0, 1]$, have been tested in order to study the influence of the parameter ρ over the classification accuracies – more details in Section 4.4.

4.3. Experimental results and discussion

Classification accuracies of State-Space Kernel, Marginalized Kernel and their extensions are reported in Tables 1 and 2 for 2-D shape recognition (*Chicken* data) and gesture recognition (*Auslan* data) applications. For each number of states, the best performance is shown. The corresponding ρ value the best performance has been reached with varies depending on data. In Section 4.4, details will be provided on experimental evaluation of the optimal ρ value. Considering all the experiments, the standard errors of the mean for the best ρ value are lower than 0.79 for the SK extensions and than 0.69 for the MK extensions in the case of *Chicken* with curvature, while they are lower than 0.8 for the SK extensions and than 0.73 for the MK extensions in the case of *Chicken* with chain codes and lower than 0.4 for the MK extensions and than 0.6 for the SK extensions in the case of *Auslan* data set.

From the tables it is evident that the proposed non-linear mapping has a beneficial impact on the performances of both the State-Space Kernel and the Marginalized Kernel for all data sets and independently from the number of HMM states. Moreover, in the *Chicken* case, the obtained results are really competitive with the state of the art, considering the difficulty of the data set. Our proposed approach performs better than techniques at the state of the art, as can be seen from Table 3, where results obtained in the literature for the same data set are shown. Also in the *Auslan* case our proposed approach is really competitive among techniques employing raw sequences – for example, in [5], authors provided a 87.2% of accuracy.

Concerning the two versions of the non-linear mapping (Eqs. (15) and (16)), it is worth to notice that they differ in performances dependently to the considered kernel, the number of states and the examined problem. In general, it seems that the State-Space Kernel performances are better enhanced when applying the approximated extension of Eq. (16), whereas for the Marginalized Kernel applying the original version in Eq. (15) seems to be more appropriate.

4.4. Study of ρ

The choice of ρ constituted a crucial point of our study. In Section 2 we justified the choice of $0 < \rho \leq 1$. In our experiments we tested different ρ values starting from 10^{-5} up to 1. Curves of the mean classification accuracy when ρ varies from 10^{-5} to 1 are displayed in Fig. 2 for Marginalized Kernel and in Fig. 3 for State-Space Kernel applied to *Chicken* data and to *Auslan* data. As can be seen from the figures, the curves follow a typical behavior, reaching a maximum for a ρ value < 1, independently from the kernel, the model (number of HMM states) and the problem. This suggests that there exists an optimal value for ρ that can unravel hidden structures of the latent variable (state) space leading to a more effective similarity measure (kernel) between objects. We are currently investigating how to discover this value automatically from the data.

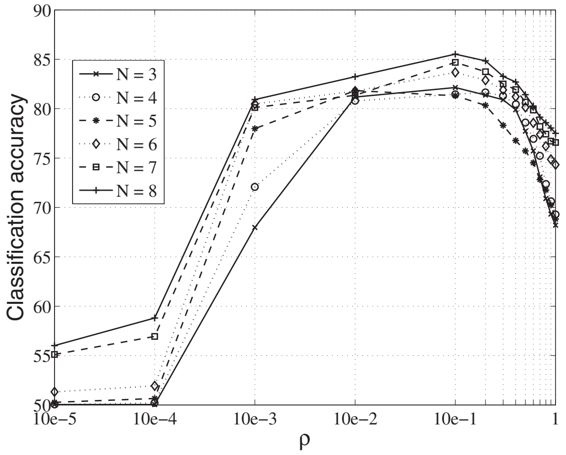

Figure 2. Marginalized NK – *Chicken* data (chain-code sequences).

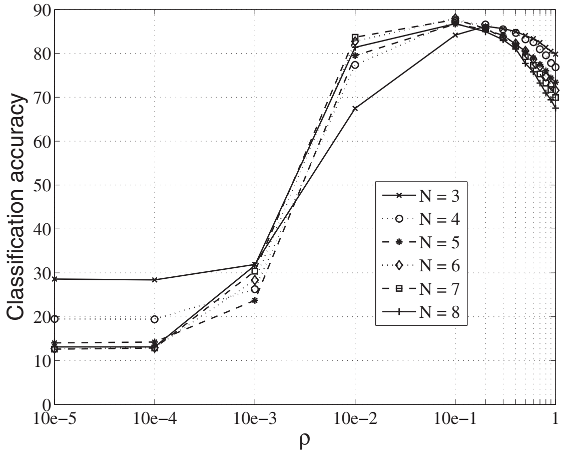

Figure 3. State-Space NK – *Auslan* data.

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] We extended in a straightforward way the Marginalized Kernel, proposed in [23] in the discrete case, to the continuous case.

Table 1. Comparison on the classification accuracy (in %) for the *Chicken* data obtained by standard SVM using Marginalized Kernel, State-Space Kernel and our new kernels derived from them by non-linear mapping. In the table, K refers to the original kernel (Eq. (14)), NK to the proposed extension (Eq. (15)) and $\widetilde{NK}$ to its variant given in Eq. (16). For each kernel, the best classification accuracy over the number of states is in boldface. The standard errors of the mean for the best ρ value are lower than 0.79 for the SK extensions and than 0.69 for the MK extensions in the case of *Chicken* with curvature, while they are lower than 0.8 for the SK extensions and than 0.73 for the MK extensions in the case of *Chicken* with chain codes.

Chicken data						
No. of HMM states	3	4	5	6	7	8
Chain-code sequences						
State-Space K	72.05	75.16	74.68	75.99	$\mathbf{76.28}$	75.11
State-Space NK	73.2	75.72	75.52	75.86	$\mathbf{76.28}$	76.15
State-Space $\widetilde{NK}$	76.19	76.64	79.64	$\mathbf{81.44}$	80.20	81.26
Marginalized K	68.2	69.28	68.83	74.32	76.58	$\mathbf{77.5}$
Marginalized NK	82.14	81.64	81.82	83.67	84.68	$\mathbf{85.52}$
Marginalized $\widetilde{NK}$	76.85	77.39	77.79	78.60	78.38	$\mathbf{79.46}$
Curvature sequences						
State-Space K	73.92	$\mathbf{75.11}$	73.58	71.37	71.33	72.05
State-Space NK	$\mathbf{75.7}$	75.59	75.52	74.75	75.05	74.71
State-Space $\widetilde{NK}$	78.9	$\mathbf{80.9}$	80.7	80.11	80.32	79.84
Marginalized K	76.22	76.46	$\mathbf{76.73}$	76.22	75.7	74.77
Marginalized NK	75.83	76.10	$\mathbf{76.55}$	76.46	75.25	74.91
Marginalized $\widetilde{NK}$	76.33	76.94	$\mathbf{78.02}$	77.75	77.25	77.84

Table 2. Comparison on the classification accuracy (in %) for the *Auslan* data obtained by standard SVM using Marginalized Kernel, State-Space Kernel and our new kernels derived from them by non-linear mapping. In the table, K refers to the original kernel (Eq. (14)), NK to the proposed extension (Eq. (15)) and $\widetilde{NK}$ to its variant given in Eq. (16). For each kernel, the best classification accuracy over the number of states is in boldface. The standard errors of the mean for the best ρ value are lower than 0.4 for the MK extensions and than 0.6 for the SK extensions.

Auslan data						
No. of HMM states	3	4	5	6	7	8
State-Space K	$\mathbf{79.82}$	76.84	73.42	71.56	69.93	67.52
State-Space NK	86.15	87.27	86.67	$\mathbf{87.98}$	87.7	86.74
State-Space $\widetilde{NK}$	$\mathbf{90.4}$	87.65	85.69	83.87	81.6	80.5
Marginalized K	$\mathbf{53.34}$	52.16	50.04	50.76	50.03	50.04
Marginalized NK	$\mathbf{93.24}$	91.76	90.57	90.46	90.18	86.77
Marginalized $\widetilde{NK}$	$\mathbf{92.69}$	89.72	88.14	86.23	85.02	83.56

Table 3. Comparative Results on the *Chicken* data.

Methodology	Protocol	Accuracy (%)	Reference
1-NN + Levenshtein edit distance	Leave One Out	≈ 67	[11]
1-NN + approximated cyclic distance	Leave One Out	≈ 78	[11]
KNN + cyclic string edit distance	Train/Test/Valid	74.3	[12]
SVM + Edit distance-based kernel	Train/Test/Valid	81.1	[12]
1-NN + mBm-based features	Leave One Out	76.5	[6]
1-NN + Hmm-based distance	Leave One Out	73.77	[6]
SVM + Hmm-based entropic features	Leave One Out	81.21	[15]

5. Conclusions

In this paper, a novel class of generative embeddings has been proposed, representing an extension of the embeddings based on generative models with latent variables, and able to project structural objects in a space with limited dimensionality. These embeddings have been obtained through a non-linear mapping, namely, a non-linear scaling of space dimensions able to highlight discriminative characteristics. We proposed one specific non-linear mapping whose basic tool is a powering operation, able to balance the contributions of each latent variable of the model, thus augmenting the entropy of the latent variables vectors. The proposed non-linear mapping has been applied to the well-known Marginalized Kernel and to the recently introduced State-Space Kernel, presenting the kernel formulation in closed form for the HMM case, in both the original and an approximated formulation.

The proposed kernels have been evaluated and compared to their standard counterparts in a classification framework involving two different sequence classification problems (2-D shape recognition and gesture recognition), with very satisfying results that outperform state-of-the-art methods.

Future research directions include the study of the optimal value for the power parameter, as well the analysis of alternative non-linear mapping functions.

Acknowledgements

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

[1] G. Andreu, A. Crespo, and J. Valiente. Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In *IEEE International Conference on Neural Networks*, volume 2, pages 1341–1346, 1997.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] M. Bicego and V. Murino. Investigating Hidden Markov Models' capabilities in 2D shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):281–286, 2004.

[4] M. Bicego, V. Murino, and M. Figueiredo. Similarity-based classification of sequences using Hidden Markov Models. *Pattern Recognition*, 37(12):2281–2291, 2004.

[5] M. Bicego, E. Pekalska, D. Tax, and R. Duin. Component–based discriminative classification for Hidden Markov Models. *Pattern Recognition*, 42(11):2637–2648, 2009.

[6] M. Bicego and A. Trudda. 2D shape classification using multifractional brownian motion. In *Joint International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, volume LNCS 5342, pages 906–916. Springer, 2008.

[7] L. Chen, H. Man, and A. Nefian. Face recognition based on multi-class mapping of Fisher scores. *Pattern Recognition*, 38(6):799–811, 2005.

[8] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, volume 11, pages 487–493, 1999.

[9] T. Jebara, I. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[10] M. Kadous. Learning comprehensible descriptions of multivariate time series. In *International Conference on Machine Learning*, pages 454–463, 1999.

[11] R. Mollineda, E. Vidal, and F. Casacuberta. Cyclic sequence alignments: Approximate versus optimal techniques. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3):291–299, 2002.

[12] M. Neuhaus and H. Bunke. Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.

[13] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, 2002.

[14] E. Oja. *Subspace Methods of Pattern Recognition*. New York: J. Wiley, 1983.

[15] A. Perina, M. Cristani, U. Castellani, and V. Murino. A new generative feature set based on entropy distance for discriminative classification. In *International Conference on Image Analysis and Processing*, 2009.

[16] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[17] W. Rudin. *Functional Analysis*. McGraw-Hill New York, 1973.

[18] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[19] N. Smith. *Using augmented statistical models and score spaces for classification*. PhD thesis, Engineering Department, Cambridge University, 2003.

[20] N. Smith and M. Gales. Speech recognition using SVMs. In *Advances in Neural Information Processing Systems*, volume 14, pages 1197–1204, 2002.

[21] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. 4th edition, Academic Press, 2009.

[22] W. Torgerson. *Theory and Methods of Scaling*. New York: Wiley, 1958.

[23] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18:S268–S275, 2002.

[24] S. Watanabe and N. Pakvasa. Subspace method in pattern recognition. In *1st. International J. Conference on Pattern Recognition*, pages 25–32, 1973.