Learning pullback HMM distances

(supplementary experiments on synthetic data)

Fabio Cuzzolin and Michael Sapienza

Abstract—We present supplementary experiments on artificial HMM sequence data as a proof of concept. Sequences of observations are synthetically generated from predefined HMMs, and our algorithm is applied to learn HMM models from the synthesized observations and to learn pullback distances in the space of learnt HMM models. Classification results compared to those of the base distance and the influence of parameter sampling density and number of states on performance are shown.

Index Terms—Distance learning, pullback metrics, hidden Markov models.

1 INTRODUCTION, TOY PROBLEM

We present experiments on a toy problem with synthetic data in order to show the merits of our pullback-HMM learning in a controlled environment. Since our method first learns an HMM from observation sequences, and measures distances between HMMs to make a classification decision, we outline a method to generate a synthetic dataset of HMM observation sequences which can then be fed to the pullback HMM learning pipeline.

Consider a scenario in which families or groups of friends go traveling to K countries drawn uniformly at random, in this case: $k \in \{$*Malta, Germany, Russia*$\}$. Each group (20 people) records the snack they eat at 4pm on each of the T days on holiday (the length of each observation sequence is $T = 14$), and the possible observations are drawn from a list of M snacks: $m \in \{$*hot chocolate, pretzels, ice cream, doughnut*$\}$. The overall group snacking distribution per day is captured by a histogram, counting how many snacks of each type have been consumed. Back home each group is instructed to learn an N-state HMM (e.g. $N = 3$) from the sequence of observations they collected. The objective is to automatically determine which country they have visited by classifying the HMM learnt on their return home.

2 SYNTHETIC HMM SEQUENCES

In order to generate the observation sequences, K transition and state-output matrices were predefined (see Fig. 1 & Fig. 2). The hidden states of each HMM represent possible weather states: Rainy, Cloudy, and Sunny (the possible states remain unknown to the travelers). The synthetic sequences' generation procedure is laid out in Algorithm 1.

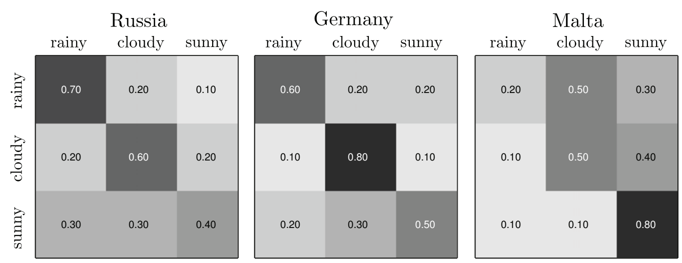

Fig. 1. *Transition matrices A based on fictional weather patterns from each country. Darker shades indicate higher probabilities.*

Algorithm 1 Generate synthetic HMM observations.

STEP 1: Initialise dataset parameters:
-classes ($k \in \{$*Malta, Germany, Russia*$\}$),
-states ($j \in \{$*Rainy, Cloudy, Sunny*$\}$)
-observations ($m \in \{$*hot-chocolate, pretzels, ice cream, doughnut*$\}$),
-length of trip ($T = 14$),
-group size ($G = 20$),
-examples per class ($L = 300$),
-transition matrices A (Fig. 1),
-state output observation matrices C (Fig. 2),
-initial state probabilities $\pi = \frac{1}{N}$
STEP 2: Monte-Carlo simulation
for $k = 1 : K$ **do**
  for $l = 1 : L$ **do**
    j = gen_rand_choice(π);
    while $length(observ) < T$ **do**
      observ = gen_rand_histogram(C, j, k, G);
      j = gen_rand_choice(A, j, k);
    end while
    Output: observ = $T \times M$-dim histograms.
  end for
end for

3 EXPERIMENTAL SETUP

To create the artificial toy dataset we generated 300 examples per class, using Algorithm 1. The obtained sequences were then averaged and plotted to manually verify that the histograms indeed matched the original discrete distributions they were sampled from. The HMM parameters were identified via the EM algorithm, applied 10 times for each observation sequence in order to select the parameters yielding the highest likelihood. The toy dataset was split into three randomly selected sets, and for each set, 2/3 of the data was placed in the

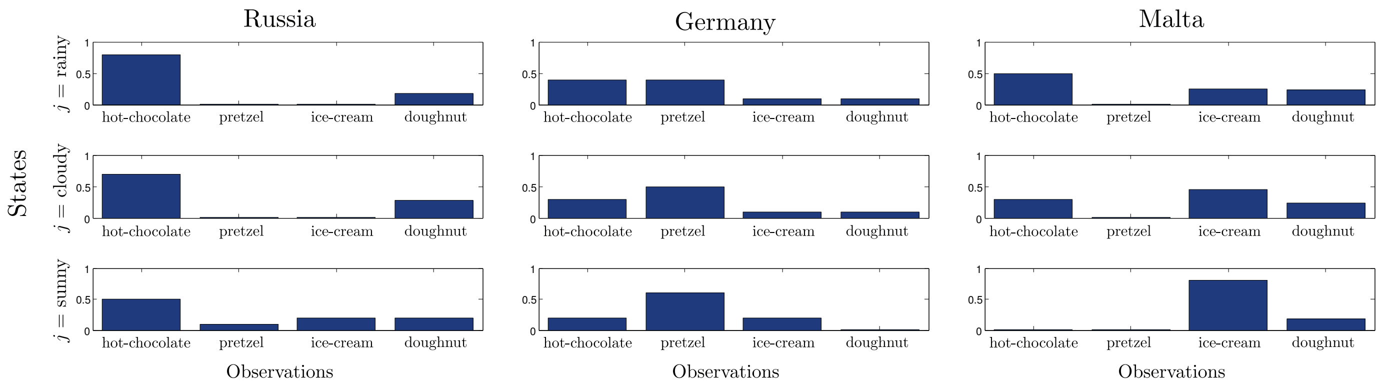

Fig. 2. *State-output probability matrices C based on fictional snacking patterns for each country under different weather conditions. The possible classes are the countries the travellers went to on holiday (Germany, Malta, Russia), the hidden states correspond to the weather conditions (sunny, cloudy and rainy), and the discrete observations are drawn from a set of snacks, in this case: hot-chocolate, pretzel, ice-cream and doughnut.*

training set, and the remainder in the test set. For each set, the parameters of the pullback learning algorithm were optimised using 5-fold cross validation on the training set.

4 Results

The results over each of the three sets were plotted using the mean accuracy and one standard deviation from the mean, as shown in Fig. 3 & Fig. 4.

In the first experiment, the sampling densities of the transition diffeomorphism F_A and output diffeomorphism F_C were varied, keeping the number of HMM states fixed at 3. To calculate the HMM approximate observation space we applied LLE with an embedding space dimensionality of $d = 3$ and number of neighbours equal to 40. In Fig. 3 the classification accuracy is plotted against the sampling densities in both Λ_A and Λ_C. It can be clearly seen that the pullback-Frobenius performance improves steadily as the sampling density in the observation automorphism's parameter space increases.

In the second experiment, the sampling density was kept constant at 8 samples in Λ_C and 216 samples in Λ_A, and the learnt number of states was varied from 2 to 4. All other parameters were kept constant. Interestingly, even though the predefined HMMs have 3 states, the HMM-pullback method was still able to discriminate between HMM models of 2 and 4 states, and even benefited from an increased number of states (see the minor improvement in accuracy and a reduced standard deviation in Fig. 4).

5 Conclusion

The supplementary experiments on artificial HMM data demonstrated the higher performance achievable using pullback distance learning for hidden Markov models, especially when compared to that of the base distance, the positive effect of higher sampling densities of the parameter space, and the robustness of the method to the number of states of the models.

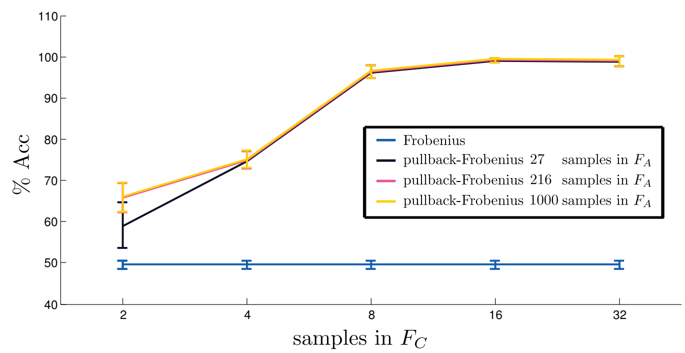

Fig. 3. *The % Accuracy plotted against the sampling density in the parameter spaces of F_A and F_C. Whereas the Accuracy of the base Frobenius distance remains constant, the Accuracy of the pullback-learning method increases steadily as the number of samples in F_C increases. The sampling density in Λ_A has a lesser effect in this case, with similar performances recorded for 216 or 1000 samples.*

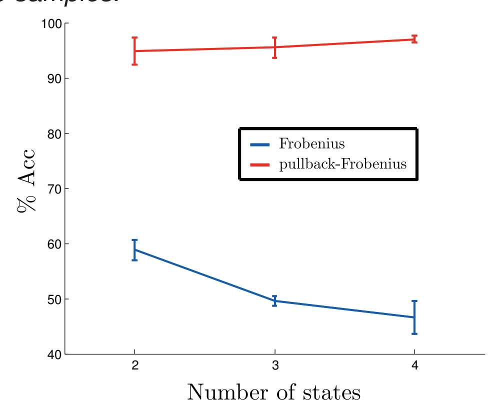

Fig. 4. *The % Accuracy plotted against the learnt number of states in the HMM. With the baseline Frobenius distance, the accuracy decreases as the number of states increases from 2 to 4. On the contrary, when using our pullback learning method the accuracy increases slightly (and the standard deviation is reduced) as the number of states increases. Also note the significant difference in accuracy in all cases.*