Temporal Data Classification by Imprecise Dynamical Models

Alessandro Antonucci
IDSIA (Switzerland)
alessandro@idsia.ch

Rocco de Rosa
Università di Milano (Italy)
rocco.derosa@unimi.it

Alessandro Giusti
IDSIA (Switzerland)
alessandrog@idsia.ch

Fabio Cuzzolin
Oxford Brookes (UK)
fabio.cuzzolin@brookes.ac.uk

Abstract

We propose a new methodology to classify temporal data with imprecise hidden Markov models. For each sequence we learn a different model by coupling the EM algorithm with the imprecise Dirichlet model. As a model descriptor, we consider the expected value of the observable variable in the limit of stationarity of the Markov chain. In the imprecise case, only the bounds of this descriptor can be evaluated. In practice the sequence, which can be regarded as a trajectory in the feature space, is summarized by a hyperbox in the same space. We classify these static but interval-valued data by a credal generalization of the k-nearest neighbors algorithm. Experiments on benchmark datasets for computer vision show that the method achieves the required robustness whilst outperforming other precise and imprecise methods.

Keywords. Time-series classification, credal sets, Markov chains, credal classification.

1 Introduction

The theory of imprecise probability (IP, [17]) extends Bayesian theory of subjective probability to cope with sets of distributions, this providing more general and robust models of uncertainty. These ideas have been applied to classification and a number of IP-based, so-called *credal*, classifiers for static data have been proposed (e.g., [19]). A key feature of these approaches is the ability of discriminating between hard-to-classify instances (e.g., for Bayesian-like approaches, those prior-dependent) for which multiple class labels are returned in output, and the others "easy" instances to which single labels are assigned. On the other side, dynamical models such as Markov chains and hidden Markov models (HMMs) have been also extended to IP in order to model the non-stationarity of a process (see e.g., [5, 6]). It seems therefore natural to merge these two lines of research and develop a credal classifier for temporal data based on imprecise HMMs, thus generalizing methods already developed for precise HMMs (e.g., [13]).

This is achieved as follows. First, from each sequence, we learn an imprecise HMM by means of a technique, already tested in [3] and [16], which combines the EM algorithm, commonly used to learn precise HMMs, with the *imprecise Dirichlet model* (IDM, [18]), a popular approach to learn IPs from (complete) data. After this step, each sequence is associated with an imprecise HMM. As a descriptor of this model (and hence of the sequence), we evaluate the lower and upper bounds of the expected values of the features in the limit of stationarity. This is based on a characterization of the limit behaviour of imprecise Markov chains provided in [6]. As a result, the sequence is associated with a hyperbox in the feature space. This represents a static, but interval-valued, datum which can be processed by a classifier. To achieve that, a generalization of the k-nearest neighbors algorithm to support interval data is proposed. Overall this corresponds to a *credal classifier* (i.e., a classifier which might return more than a single class) for temporal data. Its performances are tested on some of the most important computer vision benchmarks. The results are promising: the methods we propose achieve the required robustness in the evaluation of the class labels to be assigned to a sequence and outperform the competing imprecise method proposed in [3] with respect to state-of-the-art metrics [20] for performance evaluation. The performance is also good when comparing the algorithm with the *dynamic time warping*, a state-of-the-art approach to the classification of temporal sequences, whose performance degrades when coping with multidimensional data [14].

2 Temporal data

Let us introduce the key features of our approach and the necessary formalism for the precise case. Variables $O_1, O_2, \ldots, O_T$ denote the observations of a particular

phenomenon at T different (discrete) times. These are assumed to be observable, i.e., their actual (real) values are available and denoted by $o_1, o_2, \ldots, o_T$.

If the observations are all sampled from the same distribution, say $P(O)$, the empirical mean converges to its theoretical value (strong law of large numbers):

$$\lim_{T\to+\infty} \frac{\sum_{i=1}^{T} o_i}{T} = \int_{-\infty}^{+\infty} o \cdot P(o) \cdot \mathrm{d}o. \qquad (1)$$

Under the stationarity assumption, the empirical mean is therefore a sensible descriptor of the sequence. More generally, observations at different times can be sampled from different distributions (i.e., the process can be non-stationary). Such a situation can be modeled by pairing O_t with an auxiliary discrete variable X_t, for each $t = 1, \ldots, T$. Variables $\{X_t\}_{t=1}^{T}$ are in correspondence with the generating distributions: they all take values from the same set, say $\mathcal{X}$, whose M elements are in one-to-one correspondence with the different distributions. In other words, for each $t = 1, \ldots, T$, O_t is sampled from $P(O_t|X_t = x_t)$, and $P(O|x_{t'}) = P(O|x_{t''})$ if and only if $x_{t'} = x_{t''}$.

Variables $\{X_t\}_{t=1}^{T}$ are, generally speaking, *hidden* (i.e., their values are not directly observable). The modeling of the generative process requires therefore the assessment of the joint mass function $P(X_1, \ldots, X_T)$. This becomes particularly simple under the *Markovian assumption*: given X_{t-1}, all previous values of X are irrelevant to X_t, i.e., $P(X_t|x_{t-1}, x_{t-2}, \ldots, x_1) = P(X_t|x_{t-1})$. Together with chain rule, this implies the factorization:[1]

$$P(x_1, \ldots, x_T) := P(x_1) \cdot \prod_{t=2}^{T} P(x_t|x_{t-1}), \qquad (2)$$

for each $(x_1, \ldots, x_T) \in \mathcal{X}^T$. If the transition probabilities among the hidden variables are time-homogeneous, the specification of the joint model reduces to the assessment of $P(X_1)$ and $P(X_t|X_{t-1})$, i.e., $M^2 + M$ parameters. A model of this kind is called a *Markov chain* and, in the time-homogeneous case, it is known to assume a stationary behaviour on long sequences, i.e., the following limit exists:

$$\tilde{P}(x) := \lim_{T\to\infty} P(X_T = x), \qquad (3)$$

where the probability on the right-hand side is obtained by marginalizing out all the variables in the joint in Eq. (2) apart from X_T. The marginal probability mass function $\tilde{P}$ over $\mathcal{X}$ is called the *stationary mass function* of the chain and it can be computed by standard algorithms.

[1] We use P for both probability mass functions and densities.

In this limit, also the generation of the observations becomes stationary, i.e.,

$$\tilde{P}(O) = \sum_{x\in\mathcal{X}} P(O|x) \cdot \tilde{P}(x). \qquad (4)$$

Again, as in Eq. (1), the empirical mean converges to the theoretical value, which is now:

$$\lim_{T\to+\infty} \frac{\sum_{i=1}^{T} o_i}{T} = \sum_{x\in\mathcal{X}} \tilde{P}(x) \cdot \int_{-\infty}^{+\infty} o \cdot P(o|x) \cdot \mathrm{d}o. \qquad (5)$$

The two key points of this paper are the following: (i) emphasize the fact that, although coincident in the limit of infinite sequences, the weighted average of the means on the right-hand side of Eq. (5) provides a better descriptor than the empirical mean on the left-hand side; (ii) extend Eq. (5) to the imprecise-probabilistic framework and then use the new descriptor for robust classification of temporal data.

Concerning (i), the important remark is that the arithmetic mean does not take into account the temporal correlation of the data, while the learning of the transition probabilities $P(X_t|X_{t-1})$ and hence the corresponding value of the stationary mass function takes that into account. An empirical validation of this point is reported in Section 5. A discussion of point (ii) is in the next two sections.

3 Imprecise hidden Markov models

By merging the Markov chain defined in the previous section together with the time-homogeneous emission terms $P(O_t|X_t)$, we define a probabilistic model over the whole set of variables $X_1, O_1, \ldots, X_T, O_T$ which is called a *hidden Markov model* (HMM). An imprecise HMM is obtained by simply replacing with *credal sets*, i.e., convex sets of probability mass functions over the same variables, the precise local models $P(X_1)$, $\{P(X_{t+1}|x_t)\}_{x_t\in\mathcal{X}}$ and $\{P(O_t|x_t)\}_{x_t\in\mathcal{X}}$. While a precise HMM defines a single distribution over its whole set of variables, an imprecise HMM defines a joint credal set, which is the convex closure of the whole set of joint distributions obtained when each local model takes its values in the corresponding credal set. In the following we explain, respectively: (i) how to learn an imprecise HMM from a sequence; (ii) how to extend Eq. (5) to the case of imprecise HMMs; (iii) how to perform classification with these models.

3.1 Learning

The hidden variables $X_1, \ldots, X_T$ of a HMM, no matter whether precise or imprecise, are by definition directly unobservable. Algorithms to learn model parameters from incomplete data in HMMs are therefore

needed. A typical choice in the precise case is the EM algorithm, which finds a local optimum of the likelihood by an iterative procedure. Extending EM to IP is not trivial: credal sets can be described by a variable number of parameters (e.g., its extreme points), which cannot be easily tracked during the iteration.[2]

Despite the lack of a sound version of EM for IP, a simple heuristic approach based on the IDM has been shown to provide reasonable estimates [3]. In practice the counts required by the IDM to learn IPs, which are not available for incomplete data, are just replaced by the expectations provided by the standard EM. For the first variable in the chain, this corresponds to the following constraints:

$$\frac{E[n(x_1)]}{\sum_{x_1} E[n(x_1)] + s} \leq P(x_1) \leq \frac{E[n(x_1)] + s}{\sum_{x_1} E[n(x_1)] + s}, \tag{6}$$

for each $x_1 \in \mathcal{X}$, where $E[n(x_1)]$ is the EM expectation, after convergence, for $X_1 = x_1$, the sum is over all the elements of $\mathcal{X}$ and s is a nonnegative real parameter which describes the level of cautiousness in the learning process. Intervals in Eq. (6) are used to compute the credal set $K(X_1)$ made of the probability mass functions consistent with these (linear) constraints. We similarly proceed for the transition credal sets $\{K(X_t|x_{t-1})\}_{x_{t-1}\in\mathcal{X}}$. Considering the freedom in the choice of the number of hidden states M, it is worth noticing that the above IDM-based probability intervals are invariant with respect to that number.

Regarding the *emission* part of the model (i.e., the relation between hidden and observable variables), note that the discussion was introduced in the case of a scalar observable O just for sake of simplicity. In real-world problems, we often need to cope with sequences of arrays of $F > 1$ features, say $\mathbf{o}_1, \ldots, \mathbf{o}_T$, with $\mathbf{o}_t \in \mathbb{R}^F$ for each $t = 1, \ldots, T$. To define a joint model over the features we assume their conditional independence given the corresponding hidden variable. A Gaussian distribution is indeed used, for each feature, to model the relation between hidden and observable variables:

$$P(\mathbf{o}_t|x_t) \cdot \mathrm{d}\mathbf{o}_t = \prod_{f=1}^{F} \mathcal{N}_{\sigma_f(x_t)}^{\mu_f(x_t)}(o_t^f) \cdot \mathrm{d}o_t^f, \tag{7}$$

where o_t^f is the f-th component of the array $\mathbf{o}_t$, $\mathcal{N}_\sigma^\mu$ is a Gaussian density with mean μ and standard deviation σ, and $\mu_f(x_t)$ and $\sigma_f(x_t)$ are the EM estimates for the mean and standard deviation of the Gaussian over O_t^f given that $X_t = x_t$.[3]

Regarding the choice of the number of hidden states $M := |\mathcal{X}|$, with Gaussian emission terms the clustering method in [12] provides an optimal criterion to assess this value. The cluster information (means and standard deviations) also defines a possible initialization of for the emission terms in the EM, while uniform choices are adopted for the transition and the prior. Overall, after this learning step, the sequence of observations in the F-dimensional space is associated with a time-homogeneous imprecise HMM, with imprecise specification of the transition and prior probabilities and precise specification of the (Gaussian) emission terms.

3.2 An interval-valued descriptor for imprecise HMMs

In this section we show how the descriptor proposed in Eq. (5) for precise HMMs can be generalized to the case of the imprecise HMM we learn from a sequence of feature vectors. In the imprecise case the stationary mass function of a Markov chain is replaced by a *stationary credal set*, say $\tilde{K}(X)$. Its computation, which is briefly summarized in Appendix A, can be obtained by Choquet integration [6]. Thus, in this generalized setup, distribution $\tilde{P}(X)$ in Eq. (5) is only required to belong to $\tilde{K}(X)$. Note that $\tilde{K}$ is a finitely generated credal set which can be equivalently characterized by (a finite number of) linear constraints. Regarding the emission terms, nothing changes as they are assumed to be precise. Thus, for each feature o_f, with $f = 1, \ldots, F$, we evaluate the bounds of the expectation as

$$\underline{o}^f := \min_{\tilde{P}(X)\in\tilde{K}(X)} \sum_{x\in\mathcal{X}} \tilde{P}(x) \cdot \mu_f(x), \tag{8}$$

$$\overline{o}^f := \max_{\tilde{P}(X)\in\tilde{K}(X)} \sum_{x\in\mathcal{X}} \tilde{P}(x) \cdot \mu_f(x). \tag{9}$$

Both $\underline{o}^f$ and $\overline{o}^f$ are solutions of linear programs with $|\mathcal{X}|$ optimization variables and an equal number of linear constraints (see Appendix A). The interval $[\underline{o}^f, \overline{o}^f]$ represents therefore the range of the descriptor in Eq. (5) in the case of imprecise HMMs.

The lower and upper vectors $\underline{\mathbf{o}}, \overline{\mathbf{o}} \in \mathbb{R}^F$ are indeed obtained by applying the optimization is Eqs. (8) and (9) to each feature. They define a hyperbox in the feature space, which can be regarded as the range of the F-dimensional version of the descriptor in Eq. (5) when

[2] An exception is the EM for belief functions proposed in [7]. Yet, belief functions correspond to a special class of credal sets parametrized by a fixed number of elements.

[3] The choice of using a single Gaussian, separately for each feature, is just for the sake of simplicity. An extension of the methods proposed in this paper to a single multivariate Gaussian with non-diagonal covariance matrix would be straightforward, even with mixtures.

IPs are introduced in the model. Overall, a static interval-valued summary of the information contained in the temporal sequence has been obtained: the sequence, which is a trajectory in the feature space is described by a hyperbox in the same space (Fig. 1). In the next section, a standard approach to the classification of static data is extended to the case of interval data like the ones produced by this method.

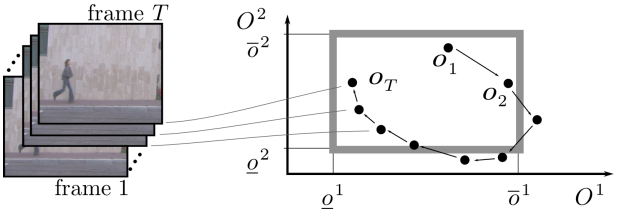

Figure 1: From trajectories to hyperboxes in the feature space. The example refers to footage data from which two features are extracted at the frame level.

4 K-nearest neighbors for interval data

4.1 Distances between hyperboxes

Consider the F-dimensional real space $\mathbb{R}^F$. Let us make it a metric space by considering, for instance, the *Manhattan* distance which, given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^F$, defines their distance δ as

$$\delta(\mathbf{x}, \mathbf{y}) := \sum_{f=1}^{F} |x_f - y_f|. \tag{10}$$

Given two points $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbb{R}^F$ such that, for each $f = 1, \ldots, F$, $\underline{x}_f \leq \overline{x}_f$, the *hyperbox* associated with these two points is denoted by $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ and defined as

$$[\underline{\mathbf{x}}, \overline{\mathbf{x}}] := \left\{ \mathbf{x} \in \mathbb{R}^F \,\middle|\, \underline{x}_f \leq x_f \leq \overline{x}_f \right\}. \tag{11}$$

The problem of extending a distance defined over points to hyperboxes can be solved by considering the general ideas proposed in [1].

Given two hyperboxes, their distance can be characterized by means of a real interval whose bounds are, respectively, the minimum and the maximum distance (according to the distance defined for points) between every possible pair of elements in the two hyperboxes. Accordingly, the lower distance between two boxes is:

$$\underline{\delta}([\underline{\mathbf{x}}, \overline{\mathbf{x}}], [\underline{\mathbf{y}}, \overline{\mathbf{y}}]) := \min_{\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \mathbf{y} \in [\underline{\mathbf{y}}, \overline{\mathbf{y}}]} \delta(\mathbf{x}, \mathbf{y}), \tag{12}$$

and similarly, with the maximum instead of the minimum for the upper distance $\overline{\delta}([\underline{\mathbf{x}}, \overline{\mathbf{x}}], [\underline{\mathbf{y}}, \overline{\mathbf{y}}])$. With the Manhattan distance in Eq. (10), the evaluation of the lower (and similarly for the upper) distance as in Eq. (12) takes a particularly simple form:

$$\underline{\delta}([\underline{\mathbf{x}}, \overline{\mathbf{x}}], [\underline{\mathbf{y}}, \overline{\mathbf{y}}]) = \sum_{f=1}^{F} \min_{\substack{\underline{x}_f \leq x_f \leq \overline{x}_f, \\ \underline{y}_f \leq y_f \leq \overline{y}_f}} |x_f - y_f|. \tag{13}$$

The optimization in the F-dimensional space is in fact reduced to F, independent, optimizations on the one-dimensional real space. Each task can be reduced to linear program whose optimum is in a combination of the extremes, unless intervals overlap. In other words:

$$\min_{\substack{\underline{x}_f \leq x_f \leq \overline{x}_f \\ \underline{y}_f \leq y_f \leq \overline{y}_f}} |x_f - y_f| = \min \left\{ \begin{array}{l} |\underline{x}_f - \underline{y}_f|, |\overline{x}_f - \underline{y}_f|, \\ |\underline{x}_f - \overline{y}_f|, |\overline{x}_f - \overline{y}_f| \end{array} \right\}, \tag{14}$$

unless $\overline{x}_f \geq \underline{y}_f$ or $\overline{y}_f \geq \underline{x}_f$, a case where the lower distance is clearly zero. A dual relation holds for the upper distance case with no special discussion in case of overlapping.

Replacing the Manhattan with the Euclidean distance makes little difference if we consider only the sum of the squared differences of the coordinates without the square root.[4] In this case the lower distance is the sum, for $f = 1, \ldots, F$ of the following terms:

$$\min_{\substack{\underline{x}_f \leq x_f \leq \overline{x}_f, \\ \underline{y}_f \leq y_f \leq \overline{y}_f}} (x_f - y_f)^2. \tag{15}$$

This is the minimum of a convex function, which is attained on the border of its (rectangular) domain. It is straightforward to check that the minimum should lie on one of the four extreme points of the domain. Thus, the minimum in Eq. (15) is the minimum of the squares of the four quantities in Eq. (14). Again, the only exception is when the two intervals overlap (the global minimum is in $x_f = y_f$), and the lower distance becomes zero. Similar considerations hold for the upper distance.

4.2 Hyperboxes classification

The above defined interval-valued distance for hyperboxes is the key to extend the *k-nearest neighbors* (k-NN) algorithm to the case of interval-valued data. First, let us review the algorithm for pointwise data.

Let C denote a *class* variable taking its values in a finite set $\mathcal{C}$. Given a collection of supervised data $\{c^d, \mathbf{x}^d\}_{d=1}^{D}$ classification is intended as the problem of assigning a class label $\tilde{c} \in \mathcal{C}$ to a new instance $\tilde{\mathbf{x}}$ on the basis of the data. The k-NN algorithm for $k = 1$

[4] The square root is a monotone function, which has no effect on the ranking-based classification method we define here.

assigns to $\tilde{\mathbf{x}}$ the label associated with the instance nearest to $\tilde{\mathbf{x}}$, i.e., the solution is $\tilde{c} := c^{d^*}$ with

$$d^* = \text{argmin}_{d=1,\ldots,D}\, \delta(\mathbf{x}, \mathbf{x}^d). \tag{16}$$

For $k > 1$, the k nearest instances need to be considered instead: a voting procedure among the relative classes decides the label of the test instance.

To extend this approach to interval data just replace the sharp distance among points used in Eq. (16) with the interval-valued distance for hyperboxes proposed in Section 4.1. Yet, to compare intervals instead of points a decision criterion is required.

To see that, consider for instance three hyperboxes and the two intervals describing the distance between the first hyperbox and, respectively, the second and the third. If the two intervals do not overlap, we can trivially identify which is the hyperbox nearer to the first one. Yet, in case of overlapping, this decision might be controversial. The most cautious approach is *interval dominance*, which simply suspends any decision in this case.

When applied to classification, interval dominance produces therefore a *credal* classifier, which might return more than a class in output. If the set of optimal classes according to this criterion is defined as $\mathcal{C}^*$, we have that $c \in \mathcal{C}^*$ if and only if there exists a datum $(c^i, \mathbf{x}^i)$ such that $c = c^i$ and

$$\overline{\delta}([\underline{\mathbf{x}}^i, \overline{\mathbf{x}}^i], [\underline{\mathbf{x}}, \overline{\mathbf{x}}]) < \underline{\delta}([\underline{\mathbf{x}}^d, \overline{\mathbf{x}}^d], [\underline{\mathbf{x}}, \overline{\mathbf{x}}]) \tag{17}$$

for each $d = 1, \ldots, D$ such that $c^d \neq c^i$. Classes in the above defined set are said to be *undominated* because they correspond to instances in the dataset whose interval-valued distance from the test instance is not clearly bigger that the interval distance associated to any other instance. A demonstrative example is in Fig. 2. Note also that the case $k > 1$ simply requires the iteration of the evaluation in Eq. (17).

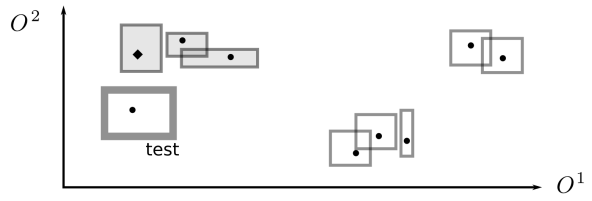

Figure 2: Rectangular data processed by the 1-NN classifier. Gray background denotes data whose interval distance from the test instance is undominated. Points inside the rectangles describe consistent precise data and the diamond is the nearest instance.

4.3 Summary and related work

By merging the discussions in Sections 3 and 4 we have a classifier, to be called iHMM-kNN, for temporal data based on imprecise HMMs. In summary, for each sequence we: (i) learn an imprecise HMM (Section 3.1); (ii) compute its stationary credal set (Appendix A); (iii) solve the LP tasks required to compute the hyperbox associated with the sequence (Section 3.2). These supervised hyperboxes are finally used to learn a credal classifier (Section 4).

Another credal classifier for temporal data based on imprecise HMMs, called here iHMM-Lik, has been proposed in [3]. Each imprecise HMM learned from a supervised sequence is used to "explain" the test instance, i.e., the lower and upper bounds of the probability of the sequence are evaluated. These (probability) intervals are compared and the optimal classes according to interval dominance returned.

Regarding traditional (i.e., not based on IP) classifiers, *dynamic time warping* (DTW) is a popular state-of-the-art approach. Yet, its performance degrades in the multi-feature (i.e., $F > 1$) case [14]. Both these methods will be compared with our classifier in the next section.

Other approaches to the specific problem of classifying interval data have been also proposed. E.g., remaining in the field of IP, the approach proposed in [15] can be used to define a SVM for interval data. Yet, time complexity increases exponentially with the number of features, thus preventing an application of the method to data with high feature dimensionality. This is not the case for iHMM-kNN, whose complexity is analyzed below.

4.4 Complexity analysis

Our approach to the learning of imprecise HMMs has the same time complexity of the precise case, namely $O(M^2TF)$. The computation of the stationary credal set is $O(T)$, while to evaluate the hyperboxes a LP task should be solved for each feature, i.e., roughly, $O(M^3F)$. Also the distance between two hyperboxes can be computed efficiently: the number of operations required is roughly four times the number of operations required to compute the distance between two points, both for Manhattan and Euclidean metrics. To classify a single instance as in Eq. (17), lower and upper distances should be evaluated for all the sequences, i.e., $O(DF)$. Overall, the complexity is linear in the number of features and in the length of the sequence and polynomial in the number of hidden states. Similar results can be found also for space.

4.5 Metrics for credal classifiers

Credal classifiers might return multiple classes in output. Evaluating their performance requires therefore specific metrics, which are reviewed here. First, a characterization of the level of indeterminacy is achieved by: the *determinacy* (*det*), i.e., percentage of instances classified with a single label; the *average output size* (*out*), i.e., average number of classes on instances for which multiple labels are returned. For accuracy we distinguish between: *single-accuracy* (*sing-acc*), i.e., accuracy over instances classified as a single label; *set-accuracy* (*set-acc*), i.e., the accuracy over the instances classified with multiple labels[5].

A utility-based measure has been recently proposed in [20] to compare credal and precise classifiers with a single indicator. In our view, this is the most principled approach to compare the 0-1 loss of a traditional classifier with a utility score defined for credal classifiers. The starting point is the *discounted accuracy*, which rewards a prediction containing q classes with $1/q$ if it contains the true class, and with 0 otherwise. This indicator can be already compared to the accuracy achieved by a determinate classifier.

Yet, risk aversion demands higher utilities for indeterminate-but-correct outputs when compared with wrong-but-determinate ones (see [20] for details). Discounted accuracy is therefore modified by a (monotone) transformation u_w with $w \in [.65, .80]$. A conservative approach consists in evaluating the whole interval $[u_{.65}, u_{.80}]$ for each credal classifier and compare it with the (single-valued) accuracy of traditional classifiers. Interval dominance can be used indeed to rank performances.

The precise counterpart of a credal classifier is a classifier always returning a single class included in the output of the credal classifier. E.g., a counterpart of iHMM-kNN is obtained by setting $s = 0$ in the IDM. If a precise counterpart is defined, it is also possible to evaluate: the *precise single accuracy* (*p-sing-acc*), i.e., the accuracy of the precise classifier when the credal returns a single label; the *precise set-accuracy* (*p-set-acc*), i.e., the accuracy of the precise classifier when the credal returns multiple labels.

5 Experiments

5.1 Benchmark datasets

To validate the performance of the iHMM-kNN algorithm we use two of the most important computer vision benchmarks: the Weizmann [8] and KTH [11] datasets for *action recognition*. For this problem, the class is the action depicted in the sequence (Fig. 3).

Figure 3: Frames extracted from the KTH dataset.

These data are footage material which requires a *features extraction* procedure at the frame level. Our approach is based on histograms of oriented optical flows [4], a simple technique which describes the flows distribution in the whole frame as an histogram with 32 bins representing directions (Fig. 4).

For a through validation also the AUSLAN dataset [9] based on gestures in the Australian sign language and the JAPVOW dataset [10] with speech about Japanese vowels are considered. Table 1 reports relevant information about these benchmark datasets.

Dataset	$\|\mathcal{C}\|$	F	D	T
KTH_1	6	32	150	51
KTH_2	6	32	150	51
KTH_3	6	32	149	51
KTH_4	6	32	150	51
KTH	6	32	599	51
Weizmann	9	32	72	105-378
AUSLAN	95	22	1865/600	45-136
JAPVOW	9	12	370/270	7-29

Table 1: Datasets used for benchmarking. The columns denotes, respectively, name, number of classes, number of features, size (test/training datasets sizes if no cross validation has been done) and the number of frames of each sequence (or their range if this number is not fixed). As usually done, the KTH dataset is also split in four subgroups.

To avoid features with small ranges being penalized by the k-NN with respect to others spanning larger domains a feature normalization step has been performed. This is a just a linear transformation in the feature space which makes the empirical mean of the sample equal to zero and the variance equal to one.

[5] In this case, classification is considered correct if the set of labels includes the true class.

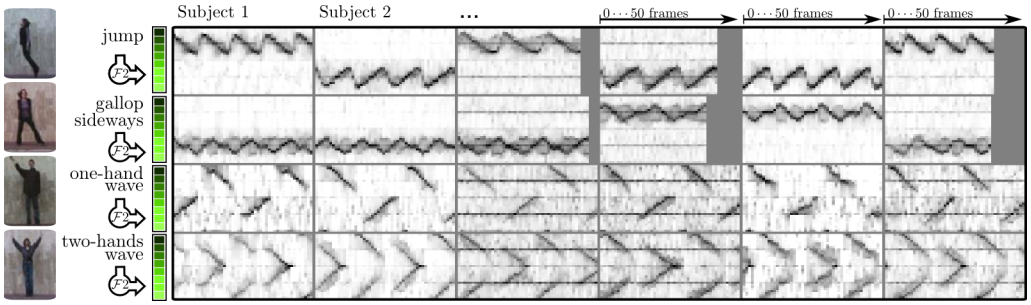

Figure 4: Low-level feature extraction. Rows correspond to different actions (i.e., class labels), columns to subjects. In each cell, feature values are shown as gray levels, with the different feature variables on the y axis, and frames on the x axis. Characteristic time-varying patterns are visible for each action.

5.2 Results

Our iHMM-kNN algorithm is empirically tested against the iHMM-Lik algorithm and the DTW on the seven datasets described in the previous section. Five runs of ten-fold cross validation are considered for KTH and Weizmann. A single run with fixed test and training set is considered instead for AUSLAN and JAPVOW. We implemented in Matlab both iHMM-kNN and iHMM-Lik.[6] Regarding DTW, the Mathworks implementation for Matlab has been used.

Our classification algorithm has only two parameters to be specified: the integer value of k in the k-NN and the real parameter s of the IDM as in Eq. (6).[7] We choose $k = 1$ because higher values could make the classifier too indeterminate. As reported in the second column of Table 2, small values are used also for s. The remaining columns of that table report the determinacies and average output size of both our algorithm and iHMM-Lik (with the same value of s). As a comment, with the selected values of s, either the determinacy is high or the average output size is consistently lower than the number of class labels. For AUSLAN, in particular, despite the very high number of classes the classifier is mostly determinate and, if not, much fewer than the original 95 classes are returned. When compared to iHMM-Lik, iHMM-kNN is less determinate and its average output size smaller. This can be explained by the high dimensionality of the feature space.

Tables 3 and 4 report information about accuracy. Results in Table 3 about single and set accuracy clearly report a higher performance of iHMM-kNN when compared to iHMM-Lik.

As noted in Section 4.5, the interval $[u_{.65}, u_{.80}]$ pro-

Dataset	s	iHMM-kNN		iHMM-Lik	
		det	*out*	*det*	*out*
KTH_1	.5	.311	2.85	.700	2.28
KTH_2	.5	.055	3.96	.565	2.13
KTH_3	.5	.135	2.91	.820	2.00
KTH_4	.5	.040	3.31	.600	2.42
KTH	.5	.111	3.51	.601	2.28
Weizmann	.5	.053	4.00	.766	2.00
AUSLAN	.01	.749	6.77	.935	2.37
JAPVOW	.01	.968	2.00	.965	2.15

Table 2: Determinacies and average output sizes for the benchmark datasets.

Dataset	iHMM-kNN		iHMM-Lik	
	sing-acc	*set-acc*	*sing-acc*	*set-acc*
KTH_1	.989	.990	.301	.017
KTH_2	.534	.981	.180	.384
KTH_3	.901	.972	.070	.083
KTH_4	.680	1.000	.269	.524
KTH	.883	.986	.299	.448
Weizmann	1.000	1.000	.275	.143
AUSLAN	.782	.675	.021	.062
JAPVOW	.958	.917	.283	.462

Table 3: Single and set accuracies on the benchmark.

[6] Both these tools are available as a free software at `http://ipg.idsia.ch/software`.

[7] Remember that the method described in [12] is used to fix the number M of states of the hidden variables. In our experiments this number ranges between 2 and 30.

vides a better summary of the credal classifiers performance by also allowing for a comparison with a traditional classifier like DTW. The results are in Table 4. Also this descriptor shows that iHMM-kNN clearly outperforms iHMM-Lik. This basically means that our interval-valued descriptor provides a better summary of a sequence rather than the interval-valued likelihood. Impressively, iHMM-kNN also competes with the DTW, showing both the quality of our approach and the (known) degradation of the DTW performance in the multiple-features case.

Dataset	iHMM-kNN		iHMM-Lik		DTW
	$u_{.65}$	$u_{.80}$	$u_{.65}$	$u_{.80}$	*acc*
KTH_1	**.659**	**.752**	.211	.212	.613
KTH_2	**.409**	**.517**	.201	.225	.369
KTH_3	**.550**	**.662**	.073	.076	.529
KTH_4	**.474**	**.597**	.281	.310	**.480**
KTH	**.495**	**.604**	.283	.309	**.525**
Weizmann	**.463**	**.575**	.236	.242	**.540**
AUSLAN	.680	.702	.021	.022	**.838**
JAPVOW	**.946**	**.951**	.283	.285	.697

Table 4: Accuracies for the benchmark datasets. Best performances are boldfaced.

Moreover, we already noted that iHMM-kNN has a precise counterpart obtained by setting $s = 0$ in the IDM constraints as in Eq. (6) and corresponding to the precise approach described in Section 2. This allows to check whether the classifier discriminates between "easy" instances (on which a single class is returned) and "difficult" ones. Results in Table 5 show that the precise single accuracy is larger than the precise set accuracy. KTH_4 is the only exception which can be explained by its low determinacy.

Dataset	*p-sing-acc*	*p-set-acc*	*acc*
KTH_1	.989	.787	.849
KTH_2	.534	.447	.451
KTH_3	.901	.671	.703
KTH_4	.680	.782	.779
KTH	.883	.674	.698
Weizmann	1.000	.842	.853
AUSLAN	.782	.351	.674
JAPVOW	.958	.333	.938

Table 5: Precise single and set accuracy of iHMM-kNN. The same classifier with $s = 0$ is used as a precise counterpart and its accuracy is in the last column. The values of *p-sing-acc* in this table coincide therefore with the *sing-acc* in Table 3.

As already discussed in Section 3.1, the adopted IDM-EM approach to the learning is the most critical part of the whole methodology. An alternative method, again heuristic and very naive, is therefore tested: LIN-VAC adopts a credal set corresponding to a *linear-vacuous mixture* [17] of the probability mass functions estimated by the EM.[8] The results of a comparison with this method for the Weizmann dataset are in Table 6. To determine the value of ϵ, we choose that leading to a determinacy comparable with that of IDM-EM. The $[u_{.65}, u_{.80}]$ intervals obtained in this way are overlapping, this suggesting the need of new, more sophisticated, models for this learning step.

Method	IDM-EM	LIN-VAC
parameter	$s = .5$	$\epsilon = .03$
det	.053	.054
out	4.00	4.38
$[u_{.65}, u_{.80}]$	[.463, .575]	[.400, .504]

Table 6: An alternative to the IDM-EM learning approach tested on the Weizmann dataset.

Finally, to validate our argument about the descriptor on the right-hand side of Eq. (5) being better than the sample mean, we compare the two descriptors in the precise case over datasets with different time lengths. When coping with short sequences the difference is in favor of our method (+2% on JAPVOW, +5% KTH_2) while the gap disappear with longer sequences (e.g., −.4% on Weizmann). This remark makes our method especially suited for the classification of short sequences.

6 Conclusions and outlooks

A new credal classifier for temporal data has been presented. Imprecise HMMs are learned from each sequence, and described as hyperbox in the feature space. These data are finally classified by a generalization of the k-NN approach. The results are promising: the algorithm outperforms another credal classifier proposed for this task and competes with the state-of-the-art method DTW. As a future work, we want to investigate novel, more reliable, learning techniques like for instance the likelihood-based approach already considered for complete data in [2]. Also more complex topologies should be considered.

[8] Given a mass function $P_0(X)$, its linear-vacuous mixture is a credal set $K(X)$ defined by the constraints $(1-\epsilon)P_0(x) \leq P(x) \leq (1-\epsilon)P_0(x)+\epsilon$. This corresponds to the vacuous credal set for $\epsilon = 1$ and to the original mass function for $\epsilon = 0$.

A Computation of the stationary credal set

Given an imprecise Markov chain as in Section 2, for each $\mathcal{X}' \subseteq \mathcal{X}$, define $Q_{\mathcal{X}'} : \mathcal{X} \to \mathbb{R}$, such that, $\forall x \in \mathcal{X}$:

$$\overline{Q}_{\mathcal{X}'}(x) := \min \left\{ \sum_{x \in \mathcal{X}'} \overline{P}(x'|x), 1 - \sum_{x \in \mathcal{X} \setminus \mathcal{X}'} \underline{P}(x'|x) \right\}. \tag{18}$$

Given this function, $\forall g : \mathcal{X} \to \mathbb{R}$, define $\overline{R}_g : \mathcal{X} \to \mathbb{R}$, such that:

$$\overline{R}_g(x) := \underline{g} + \int_{\underline{g}}^{\overline{g}} \overline{Q}_{\{x' \in \mathcal{X} : g(x') \geq t\}}(x) \mathrm{d}t, \tag{19}$$

for each $x \in \mathcal{X}$, with $\underline{g} := \min_{x \in \mathcal{X}} g(x)$ and $\overline{g} := \max_{x \in \mathcal{X}} g(x)$. Proceed similarly for the unconditional probability of the first hidden variable. In this way the following numbers (instead of functions) are defined:

$$\overline{Q}^0_{\mathcal{X}'} := \min \left\{ \sum_{x \in \mathcal{X}'} \overline{P}(x'), 1 - \sum_{x \in \mathcal{X}'} \underline{P}(x') \right\}. \tag{20}$$

$$\overline{R}^0_g := \underline{g} + \int_{\underline{g}}^{\overline{g}} \overline{Q}^0_{\{x' \in \mathcal{X} : g(x') \geq t\}} \mathrm{d}t. \tag{21}$$

A "lower" version of these functions and numbers can be obtained by simply replacing the lower probabilities with the uppers, maxima with the minima, and vice versa. For each $i = 1, \ldots, n$ let $h_i : \mathcal{X} \to \mathbb{R}$. To characterize the stationary credal set $\tilde{K}(X)$, consider $\overline{P}^*(x') := \max_{P(X) \in \tilde{K}(X)} P(x')$. Given the recursion:

$$h_{j+1}(x) := \overline{R}_{h_j}(x), \tag{22}$$

with initialization $h_1 := I_{x'}$[9], we obtain:

$$\overline{P}^*(x') := \lim_{n \to \infty} \overline{R}^0_{h_n}, \tag{23}$$

and similarly for the upper.

[9] For each $x' \in \mathcal{X}$, $I_{x'}$ is the *indicator function* of x', i.e., a function $\mathcal{X} \to \mathbb{R}$ such that $I_{x'}(x)$ is equal to one if $x = x'$ and zero otherwise.

Acknowledgements

We thank Marco Zaffalon for suggesting us the idea of using expected counts in the imprecise Dirichlet model to learn credal sets from incomplete data.

References

[1] A. Antonucci. An interval-valued dissimilarity measure for belief functions based on credal semantics. In T. Denoeux and Masson M.H., editors, *Belief Functions: Theory and Applications - Proceedings of the 2nd International Conference on Belief Functions*, volume 164 of *Advances in Soft Computing*, pages 37–44. Springer, 2012.

[2] A. Antonucci, M. Cattaneo, and G. Corani. Likelihood-based naive credal classifier. In *ISIPTA '11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications*, pages 21–30. SIPTA, 2011.

[3] A. Antonucci, R. de Rosa, and A. Giusti. Action recognition by imprecise hidden markov models. In *Proceedings of the 2011 International Conference on Image Processing, Computer Vision and Pattern Recognition, IPCV 2011*, pages 474–478. CSREA Press, 2011.

[4] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] G. de Cooman, F. Hermans, A. Antonucci, and M. Zaffalon. Epistemic irrelevance in credal networks: the case of imprecise Markov trees. *International Journal of Approximate Reasoning*, 51(9):1029–1052, 2010.

[6] G. de Cooman, F. Hermans, and E. Quaeghebeur. Sensitivity analysis for finite markov chains in discrete time. In *Uncertainty in Artificial Intelligence: Proceedings of the Twenty-Fourth Conference*, pages 129–136, 2008.

[7] T. Denoeux. Maximum likelihood from evidential data: an extension of the EM algorithm. In C. et al. Borgelt, editor, *Combining soft computing and statistical methods in data analysis (Proceedings of SMPS 2010)*, Advances in Intelligent and Soft Computing, pages 181–188. Springer, 2010.

[8] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[9] J.K. Kies. *Empirical Methods for Evaluating Video-Mediated Collaborative Work*. PhD thesis, Virginia Tech, March 1997.

[10] J. Toyama M. Kudo and M. Shimbo. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20:1103–1111, 1999.

[11] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proc. of International Conference on Pattern Recognition*, 2004.

[12] T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics*, 37(6B):3960–3984, 2009.

[13] P. Smyth. Clustering sequences with hidden Markov models. In *Advances in Neural Information Processing Systems*, pages 648–654. MIT Press, 1997.

[14] G. A. Ten Holt, M. J. T. Reinders, and E.A. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. *Time*, 5249:23–32, 2007.

[15] L.V. Utkin and F.P.A. Coolen. Interval-valued regression and classification models in the framework of machine learning. In *ISIPTA '11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications*, pages 371–380. SIPTA, 2011.

[16] A. Van Camp and G. de Cooman. A new method for learning imprecise hidden markov model. In S. Greco, B. Bouchon-Meunier, G. Coletti, B. Matarazzo, and R.R. Yager, editors, *Communications in Computer and Information Science*, volume 299, pages 460–469. Springer, 2012.

[17] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, 1991.

[18] P. Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 58(1):3–34, 1996.

[19] M. Zaffalon. The naive credal classifier. *J. Stat. Plann. Inference*, 105(1):5–21, 2002.

[20] M. Zaffalon, G. Corani, and D.D. Mauá. Utility-based accuracy measures to empirically evaluate credal classifiers. In *ISIPTA '11: Proceedings of the Seventh International Symposium on Imprecise Probability: Theories and Applications*, pages 401–410. SIPTA, 2011.