Modelling sequences using pairwise relational features

Rudy Sicard, Thierry Artières

▶ **To cite this version:**

Rudy Sicard, Thierry Artières. Modelling sequences using pairwise relational features. Pattern Recognition, 2009, 42 (9), pp.1922-1931. 10.1016/j.patcog.2008.11.023 . hal-01172412

HAL Id: hal-01172412
https://hal.science/hal-01172412v1

Submitted on 5 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Modelling sequences using pairwise relational features

Rudy Sicard*, Thierry Artières

LIP6, Université Paris 6, 104 Avenue du Président Kennedy, 75016 Paris, France

We propose a new framework for the modelling of sequences that generalizes popular models such as hidden Markov models. Our approach relies on the use of relational features that describe relationships between observations in a sequence. The use of such relational features allows implementing a variety of models from traditional Markovian models to richer models that exhibit robustness to various kinds of deformation in the input signal. We derive inference and training algorithms for our framework and provide experimental results on on-line handwriting data. We show how the models we propose may be useful for a variety of traditional tasks such as sequence classification but also for applications more related to diagnosis such as partial matching of sequences.

1. Introduction

A number of models have been proposed for sequence processing, recognition and segmentation. In order to make learning tractable these models generally rely on a number of simplifying assumptions, this is the case of one of the most popular models for sequence processing, namely hidden Markov models (HMM). A number of extensions of HMM have been proposed to take into account dependencies between observations in a sequence. One may cite among others regressive HMMs [16] and trajectory models [14]. These systems allow taking into account local temporal dependencies between observations.

Recently, conditional models have been proposed to overcome some of the major drawbacks of HMM and more generally of generative models, conditional random fields (CRF) is one of these models [12] which has been recently used in the handwriting field [19]. The nature of these conditional models avoids making any restrictive assumption about the input data distribution; no simplifying assumption is required. Although these models have been shown to outperform more traditional generative models like HMM in a few information retrieval tasks such as part-of-speech tagging, they are not so well adapted to real-valued sequence recognition tasks such as on-line handwriting recognition. Also, conditional models are learned in a discriminative way that fits well a classification task but may not be adapted for other tasks of interest in sequence processing and in on-line handwriting in particular, such as diagnosis tasks. Diagnosis covers a wide range of applications; it aims at giving an accurate evaluation about the quality of a sequence with respect to a model.

The goal of this study is to develop efficient models of sequential data for various tasks such as sequence recognition and segmentation but also for more general sequence analysis tasks such as partial recognition, rejection, diagnosis. The main idea of this work is to take into account relationships between all the observations in the input sequence. Rather than assuming independence between observations, we consider pairwise relational features between all pairs of observations and assume these are independent. A pairwise relational feature is any feature defined over a pair of observations. We focused on pairwise features and did not consider in this work triple-wise or higher order relational features since it increases drastically algorithmic complexity. Hence without ambiguity we will use relational features for pairwise relational features. Our modelling leads to a generative model whose distribution on input sequences is rather close to random fields. The choice of relational features and of the choice of a probability distribution to model these may lead to various models, traditional models such as HMM are special cases of this modelling scheme.

Relational features have also been used in the image processing field, e.g. for image segmentation. For instance, Markov random fields are a popular technology for integrating relationship among neighbouring pixels in order to smooth pixel labelling [10].

Although we describe experiments on on-line handwritten signals this work is not dedicated to such signals and could be applied to a variety of sequences and signals as well as to fixed dimensional data. In the context of handwriting relational features may

* Corresponding author.
E-mail addresses: Rudy.Sicard@lip6.fr (R. Sicard), Thierry.Artieres@lip6.fr (T. Artières).

correspond to position and spatial features. Using spatial information has proved to be useful to improve recognition accuracy [13]. It is often used with simplifying constraints [6] or as a post processing step [13] except in the case of Asian character recognition [5,23] and simple drawings analysis [7] where ad hoc methods have been investigated. These latter methods are ad hoc in that they are dedicated to handwriting. They often rely on spatial relational features exploiting the two-dimensional characteristic of handwriting. Usual features characterize the relative spatial positions (*on the left*, *on the right*, *below*, *touching*, etc.) of the different parts in a drawing. The application of our modelling framework to handwriting will allow us to show how spatial information may increase the system's robustness to noisy signals, extra strokes or temporal ordering variations.

We present the motivations and the principle of our approach in Section 2. Then we discuss in Section 3 how our formalism may be used to implement a variety of models for sequence processing tasks. Next we detail inference and training algorithms in Section 4. Finally in Section 5 we report experimental results gained on on-line handwriting data. We first investigate the behaviour of our modelling framework and the robustness of the modelling with respect to various deformations in the input signal. Then we show how our models may be used for recognition of complete or partial sequences. Note that a shorter description of this work has been published in [18]. Compared to this previous version, the present work first describes in more detail learning and inference algorithms and second includes additional experimental results.

2. Relational modelling for sequences: motivations and principle

Usual models for sequence processing include generative models such as HMMs and discriminative models such as CRF models. Discriminative models usually lead to increased accuracy in particular for classification tasks, while generative models may be preferred when dealing with diagnosis or modelling tasks, or with segmentation tasks.

Usually generative models are used to implement a joint probability distribution $p(x,y)$ over first a sequence of observations x, and second, its segmentation, y, which is a sequence of labels (i.e. states). This joint distribution is most often factorized in two terms, the likelihood of x, given the segmentation y, and the probability of y:

$$p(x,y) = p(x|y) \times p(y) \tag{1}$$

We consider in this study a rather classical modelling of the segmentation probability $p(y)$. Besides, we propose a new framework for using relational features in the definition of the likelihood term. We first present the use of relational features in the definition of $p(x|y)$, then we discuss the definition of $p(y)$.

2.1. Likelihood of a sequence using relational features

The probability $p(x|y)$ of a sequence of T observations $x = x_1 \dots x_T$ conditionally to a state sequence (i.e. a segmentation) $y = y_1 \dots y_T$ may be written as

$$p(x|y) = \prod_{t=1}^{T} p(x_t|x_1^{t-1}, y) \tag{2}$$

where x_1^{t-1} stands for $x_1 \dots x_{t-1}$. Distributions such as $p(x_i|x_1^{i-1}, y)$ being difficult to estimate, one traditionally introduces independence assumptions to simplify inference and learning. For instance, in HMM, one assumes conditional independence so that $p(x_t|x_1^{t-1}, y) = p(x_t|y_t)$. Such assumptions lead to efficient algorithms but fail at taking into account complex and long range dependencies. A number of attempts have been made for proposing richer models by considering specific temporal local dependencies. A family of such models consists, for instance, of segmental and trajectory models where one state emits globally a sequence of observations rather than emitting a sequence of successive independent observations [14].

We investigate here another alternative which consists of using as much relational features (i.e. features that describe relationships between observations) as possible for approximating $p(x_t|x_1^{t-1}, y)$. We are interested in approximations that may be expressed as a product of potential functions of the following form:

$$p(x_t|x_1^{t-1}, y) \approx \frac{1}{Z(y, x_1^{t-1})} f(x_t, y_t) \prod_{j=1}^{t-1} g(x_t, x_j, y_t, y_j) \tag{3}$$

where f and g may be any arbitrary potential functions and $Z(y, x_1^{t-1})$ is a normalization factor that ensures the above quantity is a probability.

Function f encodes local dependencies between an observation x_t and the corresponding state variable y_t while function g encodes dependencies between pairs of observations and the corresponding states. Such a pairwise modelling appears as an efficient alternative for estimating complex probabilistic distributions over a set of variables. It is an interesting trade-off between expressive power and tractability. It allows taking into account dependencies between the predicted variable x_t and multiple observed variables $x_1 \dots x_{t-1}$ through the dependencies of x_t with each one of these observed variables.

Hence, the form in Eq. (3) is quite general and exhibits more expressive power than traditional models (e.g. HMM). Using Eqs. (2) and (3) the probability of a sequence may be rewritten as

$$p(x|y) \approx \frac{1}{Z(y)} \prod_{t=1}^{T} \left[f(x_t, y_t) \prod_{j=1}^{t-1} g(x_t, x_j, y_t, y_j) \right] \tag{4}$$

The model in Eq. (4) exhibits some similarity with well-known pairwise Markov random fields that have been popularized in the image segmentation and recognition processing field [10]. Pairwise Markov random fields exploit g functions of the form $g(y_t, y_j)$ between pairs of labels only and observations are handled through f functions only.

For instance in image processing tasks x_t is a local feature describing a pixel (e.g. grey level) and g functions are used to introduce smoothing constraints on labels of neighbouring pixels (y_t and y_j).

The main difficulty in Eq. (4) lies in the normalization factor $Z(y)$ that may lead to complex and even intractable algorithm for inference. This term may, however, be computed in particular cases, for instance if all potential functions are Gaussian functions. In this work we consider locally normalized potential functions so that the model in Eq. (4) may be rewritten as a generative model as follows:

$$p(x|y) \equiv p(s,r|y) = \prod_{t=1}^{T} \left[p(s_t|y_t) \times \prod_{j=1}^{t-1} p(r_{t,j}|y_t, y_j) \right] \tag{5}$$

where s and r are two sets of features that are derived from x. The representation (s,r) of an input sequence x may be viewed as a dual representation of it. It is composed of:

- $s = (s_1, \dots, s_T)$ is a sequence of local feature vectors, where s_t stands for a vector of local features. Although there may be many other choices we systematically use in our implementation $s_t = x_t$.
- $r = \{r_{t,j}\}_{1 \leqslant t,j \leqslant T}$ consists of a matrix of relational feature vectors encoding pairwise relationships between all pairs of observations x_t and x_j. $r_{t,j}$ stands for a vector of relational features that characterizes the relationship between x_t and x_j, for instance $r_{t,j} = x_t - x_j$.

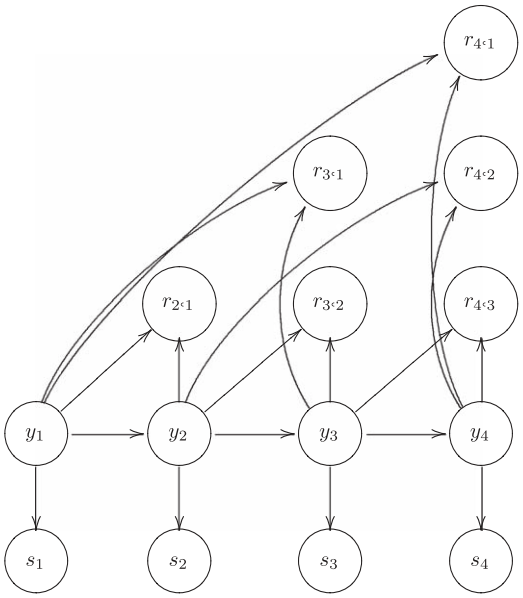

Fig. 1. Graphical representation of a dynamic pairwise relational model as in Eq. (6). Nodes y correspond to labels (states), nodes s to local feature vectors and nodes r to relational feature vectors. The model is represented unfolded for an input sequence of length 4.

As we will see, various models may be implemented depending on the definition of local feature vectors and of relational feature vectors. To make things more concrete, a simple possibility may consist of choosing $s_t = x_t$ and $r_{t,j} = x_t - x_j$, we examine other choices in Section 3.

2.2. Joint probability

We consider in this study only simple forms of the segmentation probability $p(y)$ (see Section 3.2). We considered two cases only. Either we assume that $p(y)$ is constant (i.e. segmentation are equiprobable), or we assume y obeys a first order Markov process (as in HMMs). In this latter case $p(y)$ rewrites $\prod_{t=1}^{T} p(y_t|y_{t-1})$ and the joint probability in Eq. (5) may be expressed as

$$p(x,y) = p(s,r,y) = \prod_{t=1}^{T} \left[p(s_t|y_t) \times p(y_t|y_{t-1}) \times \prod_{j=1}^{t-1} p(r_{t,j}|y_t,y_j) \right] \qquad (6)$$

with the convention that $p(y_1|y_0) = p(y_1)$. Fig. 1 illustrates the model in Eq. (6) as a dynamic Bayesian network that has been unfolded for an input sequence of size 4.

Other choices are possible for $p(y)$. Although we will derive the inference and learning algorithms in Section 5 in the particular case of a constant probability function (i.e. all events are equiprobable) $p(y)$, our approach and derivations stand whenever the prior $p(y)$ may be factorized into a product of terms involving a pair of state variables (y_t, y_j), i.e. $p(y)$ may be written as a product of functions $g(y_t, y_j)$, which is the case in the two particular cases we consider (uniform and Markovian).

3. Implementing various models

Various models may be implemented depending on prior choices concerning the form of the segmentation probability $p(y)$, the definition of local feature vectors s_t and of relational feature vectors $r_{t,j}$, and the family of parametric probability distributions for these feature vectors (e.g. $p(r_{t,j}|y_t,y_j)$).

First note that models may be built that use one kind of features only by using a uniform distribution over the features one wants to omit (local or relational). For instance one may build standard HMM by completely ignoring relational modelling which may be done by assuming that $p(r_{t,j}|y_t,y_j)$ is uniform whatever y_t and y_j.

In the following we first discuss the choices concerning fundamental components of the models (definition of relational feature vectors, etc). Then we detail how some classical models as well as new ones, may be built within our framework.

3.1. Prior and structural choices

3.1.1. Relational modelling

Two main choices determine the nature of the likelihood in Eq. (5), the choice of relational features and the related probability distribution, and the set of relational feature vectors actually exploited in the model. We first discuss how limited relational modelling may be defined through the use of a subset of relational feature vectors only, and then we detail the definition of relational feature vectors.

3.1.1.1. Range of relational modelling. The formulation in Eq. (5) makes use of all relational feature vectors between all pairs of observations, it is a general case. But one may choose to exploit a subset only of all these relational feature vectors. This subset may be defined according to a predetermined range of dependencies between observations or it may depend on the segmentation.

First one may decide to define a model that exploits a subset of relational feature vectors only, which correspond to pairs of observations that are close to each other. For instance a relational model of range k takes into account relational features $r_{t,j}$ between observations o_t and o_j whose indexes differ from less than k, i.e. $|t-j| <= k$. This may be done by assuming that $p(r_{t,j}|y_t,y_j)$ is constant if $|t-j| > k$.

Also, one may define the subset of relational features to consider as a function of the segmentation y. For instance one may decide to exploit intra-state dependencies only by assuming that $p(r_{t,j}|y_t,y_j)$ is constant if $y_t \neq y_j$.

3.1.1.2. Relational features. Delta relational feature vectors: A first intuitive choice for $r_{t,j}$ consists of using *delta relational feature vectors* based on the difference between observations vectors, e.g. $r_{t,j} = x_t - x_j$. Delta features have been shown to be very efficient and have been popularized in the field of speech recognition, but their uses were limited to delta features between successive observations. These delta features were implicitly used in standard Gaussian HMMs by transforming, in the preprocessing step, observation feature vectors x_t (e.g. cepstral feature vectors) in what we call here local feature vectors s_t of the form $s_t = \binom{x_t}{x_t - x_{t-1}}$. Our framework allows generalizing this delta modelling by exploiting delta feature vectors between non-successive observations that may have been emitted in different states. Of course, one may use non-linear delta relational features instead of linear ones, e.g. quadratic delta relational features such as $r_{t,j} = \binom{x_t - x_j}{(x_t - x_j)^2}$ where $(x_t - x_j)^2$ stands for the vector of the squares of the components of $(x_t - x_j)$.

In our experiments we used Gaussian distribution for *delta relational feature vectors*, hence $p(r_{t,j}|y_t,y_j) \equiv \mathcal{N}_{y_t,y_j}(r_{t,j})$ where $\mathcal{N}_{y_t,y_j}$ stands for a Gaussian distribution associated with the pair of labels (y_t, y_j), whose parameters are a mean relational feature vector $\mu^r_{y_t,y_j}$ and a covariance matrix $\Sigma^r_{y_t,y_j}$.

Concatenated relational feature vectors: A more general choice consists of choosing *concatenated relational feature vectors* $r_{t,j}$ to be defined as the concatenation of observations vectors x_t and x_j, i.e. $r_{t,j} = \binom{x_t}{x_j}$. Using such feature vectors allows building richer models and include *delta relational feature vectors* as a special case. This comes from the fact that one can extract x_t and x_j from $\binom{x_t}{x_j}$,

a byproduct being that one can compute $x_t - x_j$. Let note $\mathbb{1}_p$ the identity $p \times p$ matrix and $\mathbb{0}_p$ the null $p \times p$ matrix, with p being the dimension of the feature vectors x_t, and let define

$$P_1 = (\mathbb{1}_p \mathbb{0}_p) = \begin{pmatrix} 1 & & 0 & 0 & & 0 \\ & \ddots & & & \ddots & \\ 0 & & 1 & 0 & & 0 \end{pmatrix}$$

and

$$P_2 = (\mathbb{0}_p \mathbb{1}_p) = \begin{pmatrix} 0 & & 0 & 1 & & 0 \\ & \ddots & & & \ddots & \\ 0 & & 0 & 0 & & 1 \end{pmatrix}$$

Then, $x_t = P_1 \times r_{t,j}$ and $x_j = P_2 \times r_{t,j}$ and one can implement delta relational feature vectors as a special case by defining

$$p(r_{t,j}|y_t, y_j) \equiv \mathcal{N}_{y_t,y_j}(P_1 \times r_{t,j} - P_2 \times r_{t,j}) \tag{7}$$

Using such relational feature vectors allows building models, for instance predictive models [11]. As above one may use richer (e.g. non-linear) concatenated relational features including powers of input observations, for instance one may use concatenated quadratic relational features such as

$$r_{t,j} = \begin{pmatrix} x_t \\ (x_t)^2 \\ x_j \\ (x_j)^2 \end{pmatrix}$$

3.1.2. Segmentation probability

As we said previously, we investigated only two choices for the segmentation probability in our experiments, which we present now.

A first possibility is to consider that $p(y)$ is uniform, which leads to what we call *relational models*. Such models take into account the temporal ordering of the observations through relational modelling only. Consequently in such models, the segmentation of an observation sequence is driven by relational features between observations rather than by their temporal order. This may be very interesting in some pattern matching problems and in diagnosis problems as we will see in the experimental section (see Section 5).

Second, we considered a standard choice in Markovian modelling, where y is assumed to obey a first order Markov process, and is then modelled with a Markov chain. In particular, this allows implementing standard models such as HMMs and predictive models.

3.2. Examples of models

We detail now how to build various models within our framework.

3.2.1. Hidden Markov models (HMMs)

In order to build a HMM model we choose local features as $s_t = x_t$ and we assume Gaussian (or Gaussian mixtures) distributions, $p(s_t|y_t) = \mathcal{N}_{y_t}(s_t)$. The model uses no relational modelling at all and the segmentation probability is assumed to be given by a Markov chain. Hence, one gets the usual HMM joint probability:

$$p(x, y) \equiv p(s, y) = \prod_t p(s_t|y_t) \times a_{y_{t-1},y_t} \tag{8}$$

In our terminology, this model is a *local* model since it does not exploit relational modelling and it is a *sequential* model since it exploits the temporal ordering of observations, through the use of transition probabilities.

3.2.2. Pure relational models (PRM)

One may also build pure relational models (PRM) that do not make use of local information at all (e.g. by using an uniform distribution on local feature vectors s) and of temporal order of observations (by using uniform $p(y)$). Consider a model that exploits extensive relational modelling (i.e. relational feature vectors for all pairs of observations) with delta relational feature vectors. If we choose to model these relational feature vectors with Gaussian distributions, i.e. $p(r_{t,j}|y_t, y_j) \equiv \mathcal{N}_{y_t,y_j}(r_{t,j})$, then

$$p(x, y) \propto \prod_{t=1}^{T} \prod_{j=1}^{t-1} \mathcal{N}_{y_t,y_j}(r_{t,j}) \tag{9}$$

Such a model exploits all the available relational information to segment an input sequence without taking into account the temporal order of observations at all. We call it a PRM. As we will see this modelling is well suited to diagnosis tasks.

3.2.3. Hybrid relational models (HRM)

Based on the above ideas one may define a variety of models depending on the choice of local and relational features, and on the choice of corresponding distributions. The above subsections have only shown some of the models that fit in our framework. Of course it is possible to merge previous ideas, we give an example now. For instance one may simultaneously use local features in a HMM-like fashion as in Section 3.2.1, and delta relational features as in Section 3.1.1.2. At the end, the likelihood of an input sequence is defined as

$$p(x|y) = \prod_{t=1}^{T} \left[\mathcal{N}^l_{y_t}(s_t) \prod_{j=1}^{t-1} \mathcal{N}^r_{y_t,y_j}(r_{t,j}) \right] \tag{10}$$

where $\mathcal{N}^l_{y_t}$ stands for the Gaussian distribution over local features and $\mathcal{N}^r_{y_t,y_j}$ stands for the Gaussian distribution over relational features.

It may happen that one wants to put more weight on local or relational features, hence an alternate model consists of computing

$$p(x|y) = \left[\prod_{t=1}^{T} \mathcal{N}^l_{y_t}(s_t) \right]^{w_l} \times \left[\prod_{t=1}^{T} \prod_{j=1}^{t-1} \mathcal{N}^r_{y_t,y_j}(r_{t,j}) \right]^{w_r} \tag{11}$$

where w_l and w_r are real values that allow weighting the relative contributions of local and relational features to the final likelihood.

Such a modelling may be combined with various assumptions on the segmentation probability, which may be assumed uniform, or modelled with a Markov chain.

4. Algorithms

To improve the clarity of the presentation of ideas we consider all along this section that the segmentation probability $p(y)$ is uniform. Note, however, that, as we said previously, all the following derivations stand for any factorized form of $p(y)$. We first present the inference algorithm, then we discuss the training algorithm in the supervised case and in the unsupervised case.

4.1. Inference and segmentation

Given an input observation sequence x, the segmentation step consists of finding the best label sequence $\hat{y}$, i.e. the one that maximizes $P(y|x)$. This is performed though inference in a Bayesian network corresponding to the Bayesian model expressed in Eq. (6) (cf. Fig. 1). It is an inference problem which is NP-hard in our case because of the existence of, eventually many, loops in the Bayesian network [21].

There are a number of algorithms for performing inference in Bayesian networks, such as belief revision (BR) and belief propagation (BP) to name most popular ones [15]. BR aims at finding the maximum a posteriori solution (MAP) for $\hat{y}$. It is an exact algorithm for loop-less networks but its behaviour for more complex (i.e. loopy) networks is less appealing [21], e.g. its convergence is not warranted. Besides, although BP does not explicitly provide an approximation for $\hat{y}$, and despite it is an exact algorithm for loop-less networks only, BP is known to exhibit interesting properties for loopy networks [22], and to be more robust than BR for such networks. This is why we chose to use BP for inference.

BP aims at calculating marginal distributions for every random variable in the network, i.e. y_t since other variables s and r are known (computed from the input sequence x). The product of the marginal distributions may be used as an approximation of the probability $P(y|x)$. Noting $q_t(\cdot)$ the marginal distribution for random variable y_t, one may use

$$P(y|x) \approx q(y) = \prod q_t(y_t) \tag{12}$$

with $q_t(y_t = l) = \sum_{\boldsymbol{y}:y_t=l} q(\boldsymbol{y})$.

Actually, it may be shown that this approximation of $P(y|x)$ is the best factorized approximation (i.e. in a product form like $\prod h_t(y_t)$), according to the Kullback–Leibler divergence criterion [4]:

$$q(y) = \underset{h(y) \text{ such that } h(y)=\prod h_t(y_t)}{\operatorname{argmax}} \mathrm{KL}(P(y|x) \| h(y)) \tag{13}$$

where $\mathrm{KL}(P(y|x)\|h(y))$ denotes the Kullback–Leibler divergence between the true distribution and its approximation $h(y)$.

Note that if the features and the probability distributions are chosen as in Section 3.2.1 in order to implement a HMM, BP produces the same result as the forward–backward algorithm, since it is well-known that this latter dynamic programming algorithm is an instance of BP [22].

Also, note that in the general case (infinite range of relational modelling) the complexity of this BP algorithm is proportional to T^2L^2 where T stands for the sequence's length and L stands for the number of states.

4.2. Learning algorithm

Let α_l be the set of parameters of the probability distribution of local feature vectors in state l, and let $\beta_{l,m}$ be the set of parameters of the probability distribution of relational feature vectors given a pair of states, l and m. We will abusively use the following notation:

$$p(s_t|y_t = l) = p(s_t|\alpha_l) \tag{14}$$

$$p(r_{t,j}|y_t = l, y_j = m) = p(r_{t,j}|\beta_{l,m}) \tag{15}$$

Also, we will note α the set $\{\alpha_l / l = 1 \dots L\}$ (with L the number of states in the model), β the set $\{\beta_{l,m} / l, m = 1 \dots L\}$, and $\theta = (\alpha, \beta)$.

We first examine in Section 4.2.1 the supervised case where the training data set is fully labelled, i.e. one has at his disposal the true sequence of states corresponding to any sequence of observations. Then we generalize to the unsupervised case and make use of the inference algorithm described in previous section to derive an EM learning algorithm.

4.2.1. Supervised case

Assume that the training set includes a complete labelling of observation sequences. The learning set includes a set of pairs, each consists of an observation sequence x and its corresponding label sequence y, where x and y have the same length. Using superscript to index the number of a training sample, we note $X = \{x^1, \dots, x^N\}$ the set of N training observation sequences and $Y = \{y^1, \dots, y^N\}$ the set of corresponding label sequences, and T^k the length of the k th training sequence x^k (and y^k). Also we note $d^k = (s^k, r^k)$ the dual representation of x^k (as discussed in Section 2.1) and $D = \{d^1, \dots, d^N\}$. Then

$$p(\theta|D, Y) = \frac{p(D, Y|\theta)p(\theta)}{p(D, Y)} \tag{16}$$

with

$$p(D, Y|\theta) = p(D|Y, \theta) \times p(Y|\theta) \tag{17}$$

and

$$p(D|Y, \theta) = \prod_{k=1}^{N} p(s^k, r^k|y^k, \alpha, \beta) \tag{18}$$

Besides, using Eq. (5)

$$p(D|Y, \theta) = \prod_{k=1}^{N} \prod_{t=1}^{T^k} \left[p(s_t^k|\alpha_{y_t^k}) \times \prod_{j \neq t} p(r_{t,j}^k|\beta_{y_t^k, y_j^k}) \right] \tag{19}$$

where s_t^k is the t th term in sequence s^k. Let further assume that one uses a prior on model parameters and that this prior may be factorized as

$$p(\theta) = \prod_{l=1}^{L} \left[p(\alpha_l) \prod_{m=1}^{L} p(\beta_{l,m}) \right] \tag{20}$$

Then, using Eqs. (19) and (20)

$$\begin{aligned} p(\theta|D, Y) \propto & \prod_{l=1}^{L} \left[p(\alpha_l) \prod_{k=1}^{N} \prod_{t=1}^{T^k} p(s_t^k|\alpha_l)^{\delta(y_t^k, l)} \right] \\ & \times \prod_{l=1}^{L} \prod_{m=1}^{L} \left[p(\beta_{l,m}) \prod_{k=1}^{N} \prod_{t=1}^{T^k} \prod_{j \neq t} p(r_{t,j}^k|\beta_{l,m})^{\delta(y_t^k, l)\delta(y_j^k, m)} \right] \end{aligned} \tag{21}$$

where we use the delta notation $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise.

Based on Eq. (21) the parameters of all potential functions may be learned independently in order to maximize $p(\theta|D, Y)$ so that any learning method relying on the parameter's posterior probability can be used. In our implementation we used the MAP criterion with standard scale invariant priors for Gaussian distribution parameters [2]. Hence optimal parameter values are chosen according to

$$\hat{\alpha}_l = \underset{\alpha_l}{\operatorname{argmax}}\, p(\alpha_l) \prod_{k,t} p(s_t^k|\alpha_l)^{\delta(y_t^k, l)} \tag{22}$$

$$\hat{\beta}_{l,m} = \underset{\beta_{l,m}}{\operatorname{argmax}}\, p(\beta_{l,m}) \prod_{k,t,j} p(r_{t,j}^k|\beta_{l,m})^{\delta(y_t^k, l)\delta(y_j^k, m)} \tag{23}$$

4.2.2. Unsupervised case

Most often the label sequences of training sequences are unknown, the only labelling information for a sequence is a single class label. For instance in handwriting recognition, the label information corresponding to an observation sequence consists of the name of the corresponding character. In such a case, computing the parameter posterior probability requires summing over all possible segmentations Y:

$$p(\theta|D) = \sum_{Y} p(\theta, Y|D) \tag{24}$$

Initialization : $\tau = 1$; $\hat{\theta}^0$ (manually or randomly set)
Repeat :
E step: Compute marginals $q_Y^\tau(Y) = p(Y|\hat{\theta}^{\tau-1}, D)$
M step: Find $\hat{\theta}^\tau = \text{argmax}_\theta E_Y[\log p(\theta|Y, D)]$
where Y follows distribution $q_Y^\tau(Y)$
$\tau = \tau + 1$
Return $\hat{\theta}^\tau$

Fig. 2. EM algorithm for a relational model.

This formulation comes with a problem; the summation over Y is intractable and makes model parameters dependent on each other. A popular solution to overcome this problem is to rely on an EM-like algorithm, alike in HMM training. Following the derivation in [20,3], we derive an EM algorithm that maximizes the expected logarithm of the parameters posterior probability. Below, we provide some details on how the algorithm can be instantiated for our model. As usual, an iteration of the EM algorithm consists of an estimation step, and then of a maximization step. The algorithm is illustrated in Fig. 2.

In the τth iteration, the E-step consists of computing a distribution over hidden variables (Y), given observed variables (D) and model's current set of parameters (as computed in previous iteration) $\hat{\theta}^{\tau-1}$. Next the M-step consists of choosing as the new set of model's parameters, $\hat{\theta}^\tau$, the parameters that maximize the expected log posterior probability $P(\theta/D, Y)$, where the expectation is taken over Y with the approximated distribution $p(Y|\hat{\theta}^{\tau-1}, D)$.

As usual in EM like algorithms, what is actually needed in the M-step are the marginal distributions and not the complete distribution $p(Y|\hat{\theta}^{\tau-1}, D)$. This is an important point since this means we can use our inference algorithm (detailed in previous section in the E-step) whose result is precisely the marginal distributions.

We show now that, like in the supervised case, the only required quantities are marginals $p(Y|\hat{\theta}^{\tau-1}, D)$ and that the parameters corresponding to each state can be estimated independently. First note that maximizing $E_Y[\log p(\theta|Y, D)]$ is equivalent to maximizing its exponential $Q_\Theta^\tau(\theta) = \exp(E_Y[\log p(\theta|Y, D)])$. Furthermore

$$\begin{aligned} Q_\Theta^\tau(\theta) &= \exp \sum_Y p(Y|\hat{\theta}^{\tau-1}, D) \times \log p(\theta|Y, D) \\ &= \prod_Y p(\theta|Y, D)^{p(Y|\hat{\theta}^{\tau-1}, D)} \end{aligned} \tag{25}$$

Since $p(\theta|Y, D) = p(D|Y, \theta) \times p(Y|\theta) \times p(\theta)/p(D, Y)$, and assuming for simplicity that $p(Y|\theta)$ is uniform, we get

$$\begin{aligned} Q_\Theta^\tau(\theta) &= \prod_Y \left(\frac{p(D|Y, \theta) \times p(Y|\theta) \times p(\theta)}{p(D, Y)} \right)^{p(Y|\hat{\theta}^{\tau-1}, D)} \\ &= Z(D) \times p(\theta) \times \prod_Y p(D|Y, \theta)^{p(Y|\hat{\theta}^{\tau-1}, D)} \end{aligned} \tag{26}$$

where we used that $\sum_Y p(Y|\hat{\theta}^{\tau-1}, D) = 1$ and that

$$\prod_Y p(\theta)^{p(Y|\hat{\theta}^{\tau-1}, D)} = p(\theta)^{\sum_Y p(Y|\hat{\theta}^{\tau-1}, D)} = p(\theta)$$

Then, based on $\hat{\theta}^{\tau-1}$, one can compute marginals

$$p(y_t^k = l|\hat{\theta}^{\tau-1}, D) = \sum_{y^k : y_t^k = l} p(y^k|\hat{\theta}^{\tau-1}, D)$$

and $p(y_t^k{=}l, y_j^k{=}m|\hat{\theta}^{\tau-1}, D)$. Finally, $Q_\Theta^\tau(\theta)$ may be put in the following form:

$$\begin{aligned} Q_\Theta^\tau(\theta) = \prod_{l=1}^{L} \left[p(\alpha_l) \prod_{k=1}^{N} \prod_{t=1}^{T^k} p(s_t^k|\alpha_l)^{q^{\tau-1}(y_t^k=l)} \right] \\ \times \prod_{l=1}^{L} \prod_{m=1}^{L} p(\beta_{l,m}) \prod_{k=1}^{N} \prod_{t=1}^{T^k} \prod_{j<t} p(r_{t,j}^k|\beta_{l,m})^{q^{\tau-1}(y_t^k=l, y_j^k=m)} \end{aligned} \tag{27}$$

where $q^{\tau-1}(y_t^k = l)$ stands for $p(y_t^k = l|\hat{\theta}^{\tau-1}, D)$ and $q^\tau(y_t^k = q_l, y_j^k = q_m)$ stands for $p(y_t^k = l, y_j^k = m|\hat{\theta}^{\tau-1}, D)$. It follows that the optimal parameters α and β (with respect to a MAP criterion) are given by

$$\hat{\alpha}_l = \underset{\alpha_l}{\text{argmax}}\, p(\alpha_l) \prod_{k,t} p(s_t^k|\alpha_l)^{q^\tau(y_t^k=l)} \tag{28}$$

$$\hat{\beta}_{l,m} = \underset{\beta_{l,m}}{\text{argmax}}\, p(\beta_{l,m}) \prod_{k,t,j} p(r_{t,j}^k|\beta_{l,m})^{q^\tau(y_t^k=l, y_j^k=m)} \tag{29}$$

Note that these terms are computed from marginal distributions that are approximated using the inference algorithm described in Section 4.1.

5. Experiments

As discussed in Section 3, various models may be implemented from our framework that may fit with different applications, we give a few examples below. Except if indicated, all the following experiments have been performed on on-line handwritten signals of the international benchmark Unipen database [8]. An on-line handwriting signal is a temporal signal representing the successive positions of the pen (sampled at 100–200 Hz), gathered with an electronic tablet.

All the following experiments have been performed on digit recognition in a writer-dependent mode (samples in training set and the test set are written by the same writer). We only kept signals from the 12 writers who wrote at least 32 samples per digit in order first to get acceptable training set sizes (up to 20 samples per digit) and test set sizes (at least 10 samples per digit), and second to fairly compare to previous works [1,13]. In all our experiments (except for results in Fig. 7) we actually performed four experiments per writer by randomly choosing 22 training samples per digit and testing on the remaining samples (10 minimum and about 20 in average), and we report averaged results over these four experiments and over all the writers.

5.1. What does a relational model learn?

Here, to explore the modelling of relational features, we investigate what has been learned by a PRM for the handwritten digit 7. The model has three states and has been learned with three training samples similar to the one in Fig. 3, bottom.

Fig. 3, top, illustrates the distributions over delta relational features. It is a 3×3 matrix whose boxes illustrate the β parameters corresponding to pairs of states (i.e. Gaussian distribution parameters). Hence, the box on the l th row and the m th column illustrates values of parameter $\beta_{l,m} = (\mu_{l,m}, \Sigma_{l,m})$. The mean vector $\mu_{l,m}$ is an average displacement vector between observations in states l and m, it is represented by a straight line starting from the centre of the box. The ellipse that is centred at the end of this average displacement vector represents the dispersion modelled by the covariance matrix $\Sigma_{l,m}$. For example in box $(1, 3)$ one can see that observations corresponding to the third state are on the bottom and slightly on

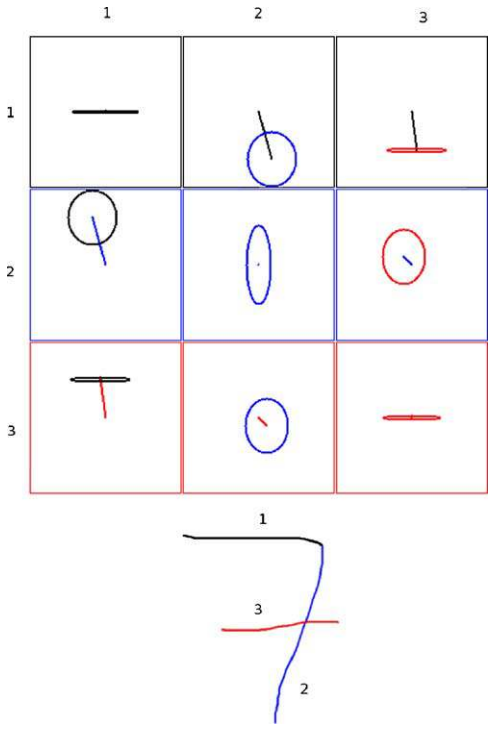

Fig. 3. Illustration of the distributions over relational features (top) in a three states purely relational model of digit 7 that has been learned with three training samples of the digit 7 similar to the one shown (bottom), whose labelling (segment associated to each of the three states) is shown.

the right of observations in the first state, with a high horizontal variability and a low vertical variability. Note that a self-state relation distribution (boxes of the diagonal) represents the deviation of observations in the state, hence a null mean displacement vector, but the variability indicates the global orientation of the set of observations in that state. For instance the second state mainly represents a vertical stroke while states one and three represent horizontal strokes. Hence these distributions allow modelling both the respective position of strokes (part of the drawing) with respect to each other but also the respective positions of observations within a part of the drawing.

5.2. Handwriting quality and robust segmentation

Evaluating the quality of an input handwriting signal may be used for different purposes. First it may be used to design a rejection mechanism in a handwriting recognition engine. One wants to reject parts of an input sequence (e.g. words in a sentence) because of low confidence on the recognition decision or because these parts may correspond to out-of-vocabulary words. Rejection mechanisms are often very simple and consist of comparing likelihoods to thresholds. There are situations where more accurate diagnoses are required. A second application is to evaluate the quality of handwriting in order to detect potential problems in childhood. Hence, there is today some interest in automating handwriting or hand draw diagnosis tools [7,9,17]. For such tasks, it is necessary to have a smart analysis method for detecting poorly written or drawn part. In order to do so one has to detect parts of letters that are badly written or not written at all, to detect additional strokes, etc. Also, one has to identify absolute and relative problems such as when two letters do not have the same height, or when an "o" is not written clockwise (i.e. in a non standard temporal ordering), or when a dot of an "i" is far too high or big, etc. Such information may be gathered from the

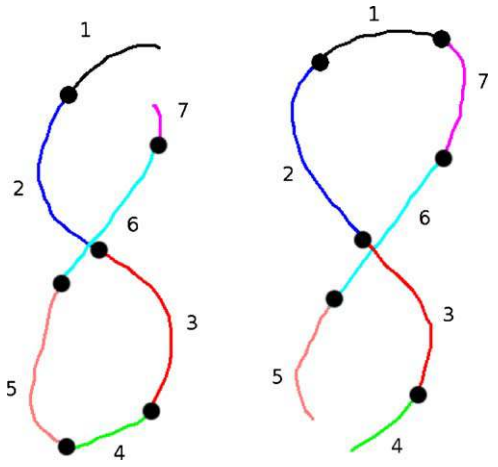

Fig. 4. Example of a test sample (right) that is correctly segmented by a model learned on sample written in reverse order (on the left). The segmentations of both samples into states, as computed by the model, are indicated with states' index.

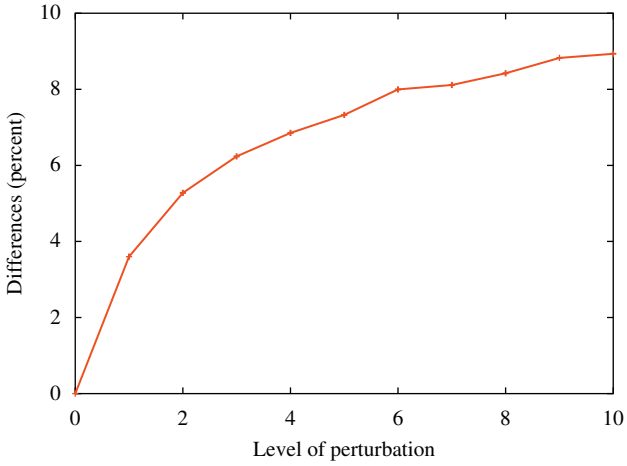

Fig. 5. Percent of points for which segmentation of the original and of the corrupted test sample differs, as a function of the level of perturbation *N*.

segmentation path. Robust segmentation is then required to design automated diagnosis systems able to determine fine grain and accurate information about the quality of a handwriting signal. Unfortunately, standard models (e.g. HMM) are very sensitive to extra strokes and noisy parts in an input signal. For instance, an extra stroke usually perturbs the segmentation of remaining strokes of an input signal.

We present here an illustration (Fig. 4) and some experiments (Fig. 5) that show the robustness of relational modelling. As we show, the use of relational features brings much robustness to the segmentation of correct parts of an input signal. In the case of handwriting, the relational model that we described in Section 2 is interesting in that it allows identifying partial writings of letters as well as unexpected additional strokes.

We observed that relational modelling is robust against drawing order variations and may recognize a letter whatever the temporal order used. Fig. 4 illustrates this with an example. A model of digit "8" with seven states has been learned from a set of training samples that are similar to the sample on the left of the figure. On the right is shown the best segmentation of a test sample that has been drawn in reverse order as computed by this model (with the inference algorithm described in Section 4.1). Segments of observations that are associated to states according to the inference algorithm are

delimited with filled circles and are labelled with the number of the state. It may be seen that although the test sample (order $5,6,7,1,2,3,4$) was not drawn in the same way as training samples were (order $1,2,3,4,5,6,7,8$), the model has been able to correctly segment the test sample. As suggested by this example, a relational model can perform robust segmentation and is rather insensitive to noisy information such as extra additional strokes, variations in temporal order, etc.

We investigated in Fig. 5 the robustness of the segmentation found by a relational model to perturbations in the ordering of the strokes drawn. We performed experiments with our subset of UNIPEN as described in the beginning of Section 5 (12 writers, 22 training samples, at least 10 testing samples, repeated 4 times), where all test samples were artificially corrupted. Given an input signal consisting of a sequence of a few strokes (separated by pen-up moves), the signal is corrupted by combining a number of three elementary steps: permuting two strokes, reversing the drawing order of a stroke, splitting a stroke into two parts in order to obtain two new strokes. These perturbations introduce a high level of noise in the temporal order of the writings.

We made the noise level vary by corrupting handwriting samples with a varying number of perturbation steps. We built a test database of samples that have been corrupted with N perturbations of each one of the three elementary steps. For a level N corruption, N "cuts" are first applied, then N "permutations" steps and finally N "reverse" steps. Fig. 4 shows the difference between the segmentation obtained on the original test sample and on the corrupted one, as a function of N. The curve corresponds to the percentage of points for which the two segmentations differ. One may see that the first level of perturbation ($N = 1$) introduces around 4% error then this rate increases slowly to 9% for $N = 10$. Considering the corruption level these results show that relational models are rather insensitive to temporal perturbation.

5.3. Recognition

Of course, our models may also be used for character recognition by training a model for each class (i.e. digit). Previous section has shown that our models may score with high likelihood an input sequence which is not complete with respect to the model (e.g. all states are not visited), which can be responsible of misrecognition. In order to perform recognition, one has to add a mechanism able to handle this completeness information. This is done by estimating during training probabilities that each state is observed. This allows computing the probability that a particular segmentation fits well the model. At recognition time, the score of a class is computed as the product of the likelihood computed by the model and of the probability of the correctness of the segmentation.

In a first series of experiments, we compare a number of standard hybrid systems using Gaussian distributions and exploiting rough observation features, the position of the pen and the direction of the trajectory (estimated with finite differences). These systems use different relational feature ranges. The first system is a standard HMM with a null range (no relational features) while next systems use increasing values of the range. In all systems, HMMs are five states left-right models. Table 1 shows that relational model significantly outperform HMM with the same topology and reduce errors rates by about 30% comparing to standard HMMs.

Fig. 6 provides another view of these results. As the range of relational modelling increases, recognition accuracy increases up to a maximum value that is reached for a range of about 10 (average sequence length is around 30). On the one hand, it shows that taking into account dependencies between distant observations (on the time axis, up to a third of the length of the sequence) may be efficiently used for recognition. On the other hand, either the model is unable to capture longer range relationships (modelling long-term dependencies is known to be difficult), or such longer range relationships are useless for recognition. A deeper analysis of this topic (e.g. through additional experiments with other characters) would be necessary to conclude about this phenomenon.

In a second series of experiments, we evaluate the capacity of our systems to provide acceptable performance with few training samples. In these experiments we compare to the segmental HMMs (SHMM) of [13,1], which were specifically designed for this kind of situation. We compare in Fig. 7 the accuracy of SHMM and of HRM as a function of the number of training samples per digit. We observe that relational model have better performance than the SHMM sys-

Table 1
Recognition rate for digit recognition of standard HMMs, of hybrid relational models (HRM) with increasing range, and of pure relational models (PRM).

System	# states	Range of relational features	Accuracy
HMM	5	1	97.2
HRM	5	3	97.6
HRM	5	5	98.0
HRM	5	10	98.3
HRM	5	20	98.1
HRM	5	∞	98.2
PRM	5	∞	66.0

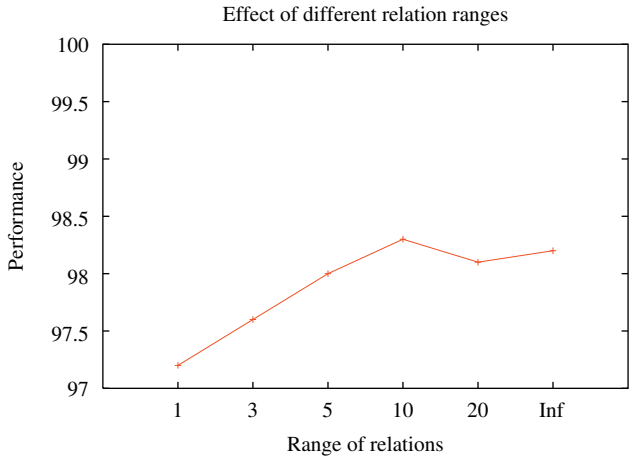

Fig. 6. Recognition accuracy of hybrid relational models using an increasing range for relational features.

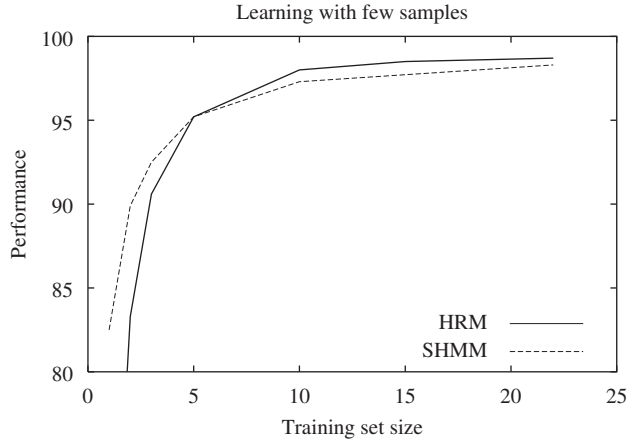

Fig. 7. Performances of the relational model of size 5 and the SHMM system for different training sizes.

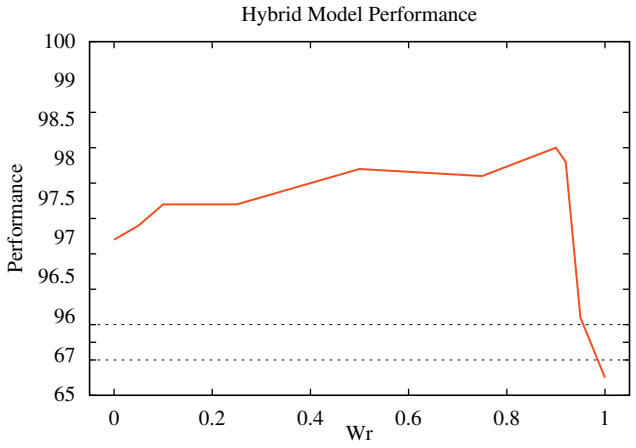

Fig. 8. Accuracy for purely relational models ($w_r = 1$), purely local models ($w_r = 0$) and mixed models ($0 < w_r < 1$), where w_l and w_r are linked by $w_l + w_r = 1$.

Table 2
Accuracy of systems dedicated to on-line handwriting (with a specific preprocessing) on a digit recognition task: segmental HMMs (SHMM) [13], standard HMMs, and hybrid relational models (HRM).

System	# states	Range	Accuracy
SHMM	Automatic	–	98.3
HMM	5	1	98.2
HMM	10	1	98.3
HRM	5	∞	98.5
HRM	10	∞	98.8
HRM	Manual	∞	99.0

tem, for moderate training size. Only for very small training sizes, the SHMM system is better. This was expected since our model is completely statistical and uses no prior knowledge, which is not the case of the SHMM system.

In a third series of experiments we investigated the relative importance of relational features and of local features for recognition. We compare in Fig. 8 the performance of HRM when weighting differently, at training and at recognition time, the local feature likelihoods and the relative features likelihood. We do this by implementing the model of Eq. (11) and by making w_l vary while linking w_r and w_r through $w_l + w_r = 1$.

When $w_r = 0$ only local features are used while when $w_r = 1$ only relational features are used. It is interesting to note that the combination of local and relational features is useful for a large range of w_r values. Depending on the weight considering relational features may help reducing the error-rate to up to 45% over the performance of standard models exploiting local features only.

Finally in a last series of experiments we compare the performances of more complex Markovian model, the SHMM [13,1]. These models are dedicated to the on-line handwriting signal, they use spatial relation features and can be trained with very few training samples. We provide comparative results of HRM that are more dedicated to handwriting signals in that they make use of the same dedicated preprocessing to the handwriting signal as in [13,1] (the preprocessing takes into account pen-up and pen-down moves and use a more detailed coding of trajectory direction angle). We do not detail this preprocessing of the signal here since the main idea is to fairly compare a mature system [13,1] with our new approach. As may be seen in Table 2, HMMs and SHMMs perform similarly while HRM allow reaching higher recognition rates, whatever the number of states in digit models.

6. Conclusion and discussion

We presented a new modelling framework for sequences that overcomes classical drawbacks of HMM concerning independence assumption between successive observations. This framework explicitly models long range dependencies between observations in a sequence by using pairwise relational features characterizing the relation between each pair of observations. This strategy allows building a variety of models that exploit both local and relational features, it includes HMM as a special case. We provided a detailed derivation of inference and learning algorithms for these models. We showed experimentally their intrinsic interest through on-line handwriting experiments for tasks such as partial matching, diagnosis, sequence recognition and segmentation. In particular we showed first that our framework allows building systems that may outperform tuned HMM based systems for classification tasks. Also we showed how introducing relational features may bring much robustness to deformations in the input signal such as extra parts, unusual temporal ordering. A side effect of this is that our models may be used to detect extra parts, deformed parts, and missing parts in a drawing. This is a very new and original feature that may be exploited to build accurate and smart diagnosis systems for e.g. handwriting quality evaluation and disease detection.

There are natural extensions to this work. A first extension is to use segmental models. In SHMMs emission probability functions are defined on segments of observations rather than on isolated observations. These models have been shown to increase modelling accuracy. A segmental extension of relational models would allow taking into account relational features between segments associated to different states, which seems intuitively appealing. Secondly we plan to exploit higher order relational features (e.g. triple-wise relational features) which requires designing more efficient algorithms in order to deal with large sized applications. Finally we aim at applying this approach and its potential extensions to other application fields such as textual data analysis. Our framework could actually fit well with tasks such as information extraction where many items has to be put in correspondence (name, location, date, event), or in post-tagging where grammatical hints between words (concordance rules…) could be successfully exploited.

References

[1] T. Artieres, S. Marukatat, P. Gallinari, Online handwritten shape recognition using segmental hidden Markov models, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 205–217.
[2] J.M. Bernardo, Probability and statistics, encyclopedia of life support systems, Bayesian Statistics. UNESCO, Oxford UK, 2003.
[3] C.M. Bishop, Pattern Recognition and Machine Learning, in: Information Science and Statistics, Springer, Berlin, 2006, pp. 423–455 (Chapter 9).
[4] J.-F. Cardoso, Dependence, correlation and Gaussinanity in independent component analysis, Machine Learning Res. (2003) 1177–1203.
[5] L. Cham, M. Chang, Stroke order and stroke number free on-line Chinese character recognition using attributed relational graph matching, in: International Conference on Pattern Recognition, vol. 3, 1996, pp. 259–263.
[6] S. Cho, J. Kim, Bayesian network modelling of strokes and their relationships for on-line handwriting recognition, in: International Conference of Document Analysis and Recognition, vol. 6, 2001, pp. 86–90.
[7] S. Glenat, L. Heutte, T. Paquet, D. Mellier, Computer-based diagnosis of dyspraxia: the MEDDRAW project, in: Conference of the International Graphonomics Society, vol. 12, 2005, pp. 49–53.
[8] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet, Unipen project of on-line data exchange and recognizer benchmarks, Pattern Recognition, Conference B: Computer Vision and Image Processing, Proceedings of the 12th IAPR International, vol. 2, 1994, pp. 29–33.
[9] L. Hamstrabletz, J. Debie, B.D. Brinker, Concise Evaluation Scale for Children Handwriting, Lisse, Swets and Zeilinger, 1987.
[10] K. Held, et al., Markov random field segmentation of brain MR images, IEEE Trans. Med. Imaging 16 (6) (1997) 878–886.
[11] V. Krishnamurthy, T. Ryden, Consistency and identifiability of autoregressive models with Markov regime, in: Information, Decision and Control, 1999, pp. 251–256.

[12] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: International Conference on Machine Learning, 2001, pp. 282–289.
[13] S. Marukatat, T. Artières, Handling spatial information in on-line handwriting recognition, in: International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 14–19.
[14] M. Ostendorf, V. Digalakis, O. Kimball, From HMMs to segment models: a unified view of stochastic modelling for speech recognition, IEEE Trans. Speech Audio Process. 4 (5) (1996) 360–378.
[15] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kauffmann, Los Altos, CA, 1988.
[16] A. Poritz, Linear predictive hidden Markov models and the speech signal, in: International Conference in Acoustics, Speech and Signal Processing, 1982, pp. 1291–1294.
[17] S. Rosenblum, P. Weiss, S. Parush, Product and process evaluation of handwriting difficulties, Educational Psychology Review (15) (2003) 41–81.
[18] R. Sicard, T. Artières, E. Petit, Modelling on-line handwriting using pairwise relational features, in: International Workshop on Frontiers in Handwriting Recognition, 2006, pp. 267–274.
[19] M. Szummer, Y. Qi, Contextual recognition of hand-drawn diagrams with conditional random fields, in: 9th International Workshop on Frontiers in Handwriting Recognition, 2004, pp. 32–37.
[20] M. Tanner, Tools for Statistical Inference, third ed., Springer, Berlin, 1998 pp. 64–85.
[21] Y. Weiss, W. Freeman, On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs, IEEE Trans. Inform. Theory 47 (2) (2001) 736–744.
[22] Y. Weiss, W.T. Freeman, Correctness of belief propagation in gaussian graphical models of arbitrary topology, Neural Comput. 13 (10) (2001) 2173–2200.
[23] J. Zheng, X. Ding, Y. Wu, Z. Lu, Spatio-temporal unified model for on-line handwritten Chinese character recognition, Document Analysis and Recognition, 1999, pp. 649–652.