



Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Recognition of degraded characters using dynamic Bayesian networks

Laurence Likforman-Sulem*, Marc Sigelle

TELECOM Paris Tech/TSI and CNRS LTCI UMR 5141, 46 rue Barrault F-75634 Paris Cedex 13, France

ARTICLE INFO

Article history:

Received 1 March 2007

Received in revised form 15 January 2008

Accepted 15 March 2008

Keywords:

Markovian models

Hidden Markov models

Dynamic Bayesian networks

Historical documents

Broken character recognition

ABSTRACT

In this paper, we investigate the application of dynamic Bayesian networks (DBNs) to the recognition of degraded characters. DBNs are an extension of one-dimensional hidden Markov models (HMMs) which can handle several observation and state sequences. In our study, characters are represented by the coupling of two HMM architectures into a single DBN model. The interacting HMMs are a *vertical* HMM and a *horizontal* HMM whose observable outputs are the image columns and image rows, respectively. Various couplings are proposed where interactions are achieved through the causal influence between state variables. We compare non-coupled and coupled models on two tasks: the recognition of artificially degraded handwritten digits and the recognition of real degraded old printed characters. Our models show that coupled architectures perform more accurately on degraded characters than basic HMMs, the linear combination of independent HMM scores, as well as discriminative methods such as support vector machines (SVMs).

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Since the seminal work of Rabiner [1], stochastic approaches such as hidden Markov models (HMMs) have been widely applied to speech recognition, handwriting [2,3] and degraded text recognition [4,5]. This is largely due to their ability to cope with incomplete information and non-linear distortions. These models can handle variable length observation sequences and offer joint segmentation and recognition which are useful to avoid segmenting cursive words into characters [6]. However, HMMs may also be used as classifiers for single characters [7,8] or characters segmented from words by an "explicit" segmentation method [9]: the scores output for each character and each class are combined at the word level. Another property of HMMs is that they belong to the class of generative models. Generative models better cope with degradation since they rely on scores output for each character and each class while discriminative models, like neural networks and support vector machines (SVMs), are powerful to discriminate classes through frontiers. In case of degradation, characters are expected to be still correctly classified by generative models even if lower scores are given.

Noisy and degraded text recognition is still a challenging task for a classifier [10]. In the field of historical document analysis, old printed documents have a high occurrence of degraded characters, especially broken characters due to ink fading. When dealing with

broken characters, several options are generally considered: restoring and enhancing characters [11–13] or recovering characters through sub-graphs within a global word graph optimization scheme [14]. Another solution is to combine classifiers or to combine data. Several methods can be used for combining classifiers [15], one of them consists of multiplying or summing the output scores of each classifier. In the works of [16,17], two HMMs are combined to recognize words. A first HMM, modeling pixel columns, proposes word hypotheses and the corresponding word segmentation into characters. The hypothesized characters or sub segments are then given to a second HMM modeling pixel rows. This second HMM normalizes and classifies single characters. The results of both HMMs are combined by a weighted voting approach or by multiplying scores. Our approach differs with restoration methods as it aims at enhancing the classification of characters without restoration. This is motivated by the fact that preprocessing may introduce distortions to character images. In our previous work [18], we compared data and decision fusion and showed that data fusion yields better accuracy than decision fusion for HMM-based printed character recognition. The present dynamic Bayesian network (DBN) approach is a data fusion scheme which couples two data streams, image columns and image rows into a single DBN classifier. It differs from the approach presented in [16,17] where two classifiers are coupled (one classifier per stream) in a decision fusion scheme, and from a data fusion scheme consisting of a multi-stream HMM which would require large and full covariance matrices in order to take into account dependencies between the streams [18].

Our study consists of building DBN models which include in a single classifier two sequences of observations: the pixel rows and

* Corresponding author. Tel.: +33 1 45 81 73 28.

E-mail address: laurence.likforman@telecom-paristech.fr (L. Likforman-Sulem).

the pixel columns. It can be seen as coupling two HMMs into a single DBN classifier, as opposed to combining the scores of two basic HMM classifiers in a decision fusion scheme. The two HMM architectures, each including an observation stream associated with state variables, are linked in a graphics-based representation. Two different streams are jointly observed and the model parameters (state transition matrices) reflect the spatial correlations between these observations.

We apply the DBN models to broken character recognition. As generative models, DBNs are adapted to degraded character recognition. These models also provide a certain robustness to degradation due to their ability to cope with missing information. They have the ability to exploit spatial correlations between observations. Thus a corrupted observation in the image can be compensated by an uncorrupted one. We compare several DBN architectures among themselves, with other fusion models like the combination of independent HMMs, and with a SVM classifier.

The paper is organized as follows. In Section 2, we briefly introduce Bayesian networks (BN) and DBNs. In Section 3, we present several independent or coupled models. In Section 4, we apply these models to the problem of broken character recognition (artificial and real). We conduct several experiments to show the advantages of DBNs by comparing their performance with the combination of HMM scores and with a SVM classifier. Conclusions are drawn in Section 5.

2. Dynamic Bayesian networks

A (static) BN associated with a set of random variables $\mathbf{X} = (X_1, X_2, \dots, X_N)$ is a pair: $B = (G, \theta)$ where G is the structure of the BN i.e., a directed acyclic graph (DAG) whose nodes correspond to the variables $X_i \in \mathbf{X}$ and whose edges represent their conditional dependencies, and θ represents the set of parameters encoding the conditional probabilities of each node variable given its parents. The distributions are represented either by a conditional probability table (CPT) when a node and its parents represent discrete variables, or by a conditional probability distribution (CPD) when a node represents a continuous variable. Each CPD usually follows a Gaussian probability density function (pdf). A key property of BNs is that the joint probability distribution factors as

$$P(X_1, X_2, \dots, X_N) = \prod_{i=1}^N P(X_i | Pa(X_i))$$

where $Pa(X_i)$ denotes the parents of X_i . This property is central in the development of fast inference algorithms. Static BNs have been applied to on-line character recognition and signature authentication for modelling dependencies between stroke positions or signature components [19–21].

DBNs are an extension of static BNs to temporal processes occurring at discrete times $t \geq 1$. In the following, we consider DBN models which have two observation streams. We will use indices $i = 1, 2$ to denote the two streams. The variables X_i and Y_i denote the respective

hidden state and observation attributes in stream i . X_t^i and Y_t^i are the random variables (nodes) for X_i and Y_i at time t .

We assume that the process modelled by DBNs is first-order Markovian and stationary. In practice, this means that the parents of any variable X_t^i or Y_t^i belong to the time-slice t or $t - 1$ only, and that model parameters are independent of t . Parameters are thus tied and a DBN can be represented by the first two time slices as in Fig. 1. For each observation sequence, the network is repeated as many times as necessary. Fig. 1 shows an example of unrolled DBN for an observation sequence of length $T = 3$: the initial network is repeated T times. Parameters for this model are given by CPTs and CPDs: the three CPTs are the initial state distribution encoding $P(X_1^1)$, the conditional state distribution $P(X_t^2 | X_t^1)$, the state transition distribution $P(X_t^2 | X_{t-1}^2)$ and the two CPDs are the Gaussian pdfs $P(Y_t^i | X_t^i)$, $i = 1, 2$.

DBNs provide general-purpose training and decoding algorithms based on the expectation-maximization (EM) algorithm and on inference mechanisms [22]. Model training consists of estimating model parameters, CPTs and CPDs. Inference algorithms are performed on the network to compute the best state sequences or the likelihoods of observation sequences.

An HMM is a particular case of DBN where there is only one observation stream and one state sequence. The dynamic character of DBNs makes it suitable for applications such as speech and character recognition. In [23,24], DBNs are used to model the interactions between speech observations at different frequency bands in a way that is robust with respect to noise.

3. Independent and coupled architectures

In this study, we couple data streams into single DBN classifiers. This coupling is performed through various DBN architectures (graphical representations) which combine two basic HMMs: the *vertical* HMM whose outputs are the columns of pixels and the *horizontal* HMM whose outputs are the image rows. In our models, the interactions are usually (but not only) performed through states, leading to efficient models in terms of model complexity (see Section 3.3). Brand et al. [25] have proposed coupled architectures "coupled HMMs" for modeling human interactions: in their models, a state of one HMM is linked to all other HMM states of the adjacent time-slice. This yields symmetric architectures while our coupled architectures are highly non-symmetric.

In our framework, all character classes share the same DBN architecture. Admissible architectures do not include continuous variables with discrete children (for exact inference purposes [23]) and have also a small number of parameters (in order to get a tractable inference algorithm). One approach consists of learning network architecture from data [26]. This approach is tractable for static BNs when dealing with a few observed variables but becomes rapidly too complex in the presence of hidden state variables. Automatic architecture learning is beyond the scope of this paper and our strategy consists of heuristically looking for various admissible architectures and selecting those which provide the best recognition performance.

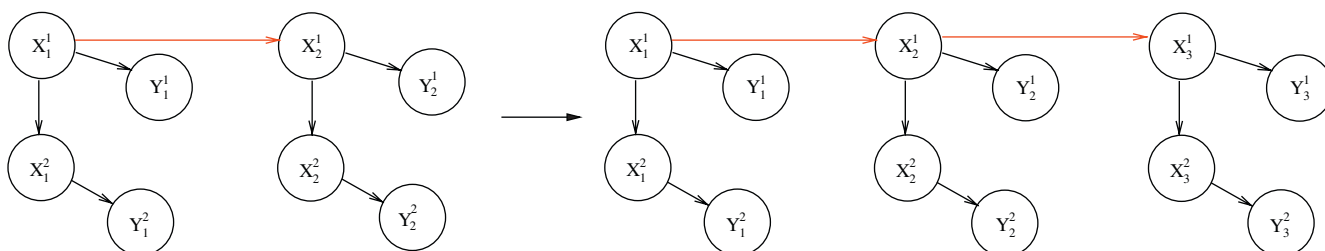


Fig. 1. Because of parameter tying, a DBN can be represented by only two time slices (left). To fit the two observation sequences $\{Y^1\}$ and $\{Y^2\}$ of length $T = 3$, the DBN is unrolled and represented on 3 time slices (right).

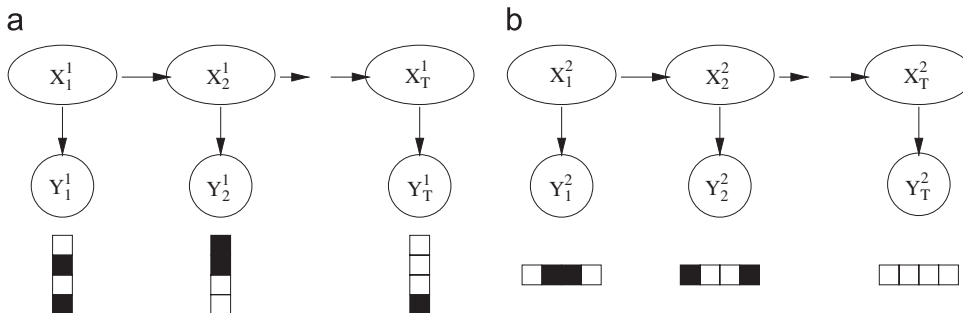


Fig. 2. Independent HMMs represented as DBNs: (a) vertical-HMM and (b) horizontal-HMM.

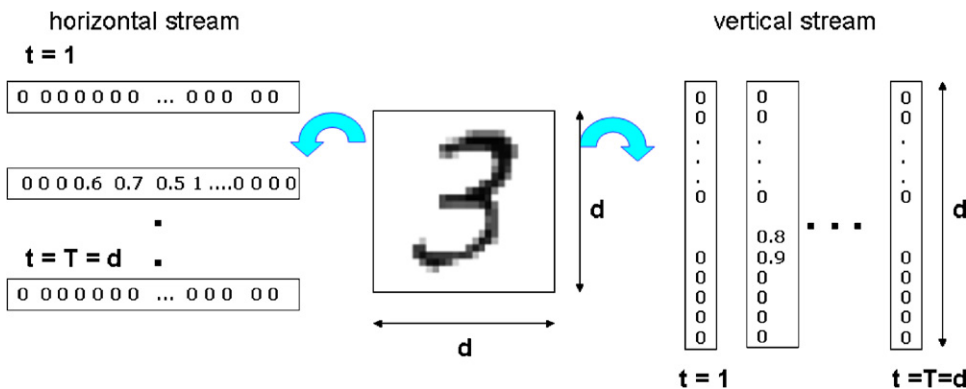


Fig. 3. Horizontal and vertical observation sequences obtained by scanning digit 3 from top to bottom and from left to right, respectively. Digit images are normalized to size $d \times d$. Length of observation sequences is $T = d$, length of observation vectors is also d .

3.1. Independent architectures

We construct two basic HMMs using the DBN formalism. The vertical (resp. horizontal) HMM is constructed using the vertical (resp. horizontal) writing stream, as depicted in Fig. 2a and b. Observations for the vertical (resp. horizontal) HMM consist of columns (resp. rows) of pixels (normalized values) obtained from scanning the character image from left to right (resp. top to bottom) as shown in Fig. 3. Characters are normalized to a square of size $d \times d$ pixels (see Section 4). Thus the length T of each observation sequence, either horizontal or vertical, is $T = d$ and the length of observation vectors is also d .

The parameters of these basic HMMs are CPTs A , CPDs B and the initial distribution Π . CPTs A are associated to nodes X_t^i , $t \geq 2$, CPDs B to observed nodes Y_t^i ¹ and the initial state distribution Π is associated to node X_1^i . They are written for each stream i , vertical ($i = 1$) or horizontal ($i = 2$), and for $t \geq 2$ as

$$\begin{cases} A_{j,k}^i = P(X_t^i = k | X_{t-1}^i = j), & \forall k, j \in [1, Q] \\ B_k^i(y_t^i) = P(Y_t^i = y_t^i | X_t^i = k) \\ \quad = \mathcal{N}(y_t^i; \mu_k^i, \Sigma_k^i), \\ \Pi_1^i(k) = P(X_1^i = k), & \forall k \in [1, Q]. \end{cases} \quad (1)$$

Q is the *global* number of hidden states. CPTs A are state transition matrices of size $Q \times Q$. As in classical HMMs, we constrain A to allow only left-right state transitions for parameter reduction purposes: the value of X_t^i is either equal to the value of X_{t-1}^i or equal to that of $X_{t-1}^i + 1$. Each observation variable Y_t^i follows a *single* Gaussian

probability density function (pdf) $\mathcal{N}(y_t^i, \Sigma_k^i)$: μ_k^i is the mean vector of length d of the Gaussian pdf associated to the current state k , Σ_k^i is a full covariance matrix of size $d \times d$ (see Section 3.3).

A first limitation of HMMs is the observation independence assumption conditionally to hidden states. However, we can bypass it by building auto-regressive (AR) architectures where observations are explicitly dynamically linked in time. An auto-regressive HMM is determined by its type and the order p of the regression. There are two types of auto-regressive HMMs: linear predictive models [1] and switching Markov auto-regressive models [27]. The AR models proposed here are switching Markov models and the model order is one. An observed node Y_t^i depends on both the current state X_t^i and the previous observed node Y_{t-1}^i . The two resulting *vertical-AR* and *horizontal-AR* single stream architectures remain however independent (Fig. 4a and b). The only parameters which differ from basic HMMs are the CPDs B . The mean μ_k^i of the Gaussian probability density function associated to the current state k is shifted by $W_k^i y_{t-1}^i$ according to the previous observation y_{t-1}^i and the regression matrix W_k^i . Each regression matrix W_k^i is of size $d \times d$, with d being the length of observation vectors. Regression matrices for each stream and each state are estimated during training. Each observation variable Y_t^i follows a Gaussian probability density function $\mathcal{N}(y_t^i; \mu_k^i + W_k^i y_{t-1}^i, \Sigma_k^i)$. CPDs B are written for each stream i and for $t \geq 2$ as

$$\begin{cases} B_k^i(y_t^i, y_{t-1}^i) = P(Y_t^i = y_t^i | X_t^i = k, Y_{t-1}^i = y_{t-1}^i), \forall k \in [1, Q] \\ \quad = \mathcal{N}(y_t^i; \mu_k^i + W_k^i y_{t-1}^i, \Sigma_k^i). \end{cases} \quad (2)$$

The matrix A and the initial distribution Π remain the same as for basic HMMs. The matrix A is still constrained to be left-right. Note that basic HMMs are a particular case of AR-HMMs with $W_k^i = 0$.

¹ Because of the stationarity assumption, all nodes X_t^i share the same CPT A and all nodes Y_t^i share the same matrix B .

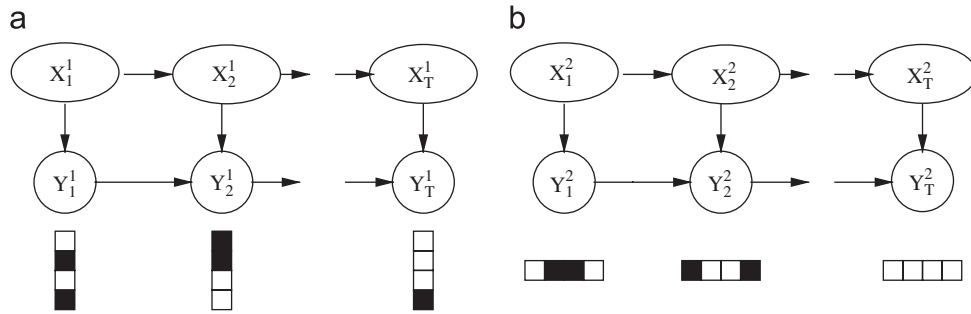


Fig. 4. Independent auto-regressive AR-HMMs represented as DBNs: (a) AR-vertical and (b) AR-horizontal.

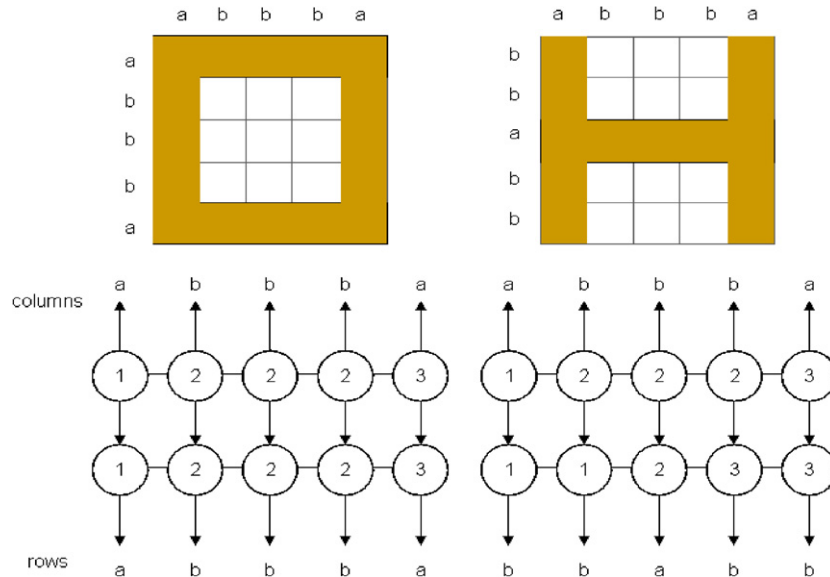


Fig. 5. Observation and state sequences for simple O and H shapes on a 5 × 5 grid. Joint configurations of long bars of pixels (observations a) and short bars of pixels (observations b) occur for different state configurations.

3.2. Coupled architectures

Starting from the previous single stream and independent HMMs, we now construct several coupled architectures. They are obtained by adding directed edges between the two streams within the same time-slice. Edges are directed from the vertical stream to the horizontal one in order to enhance the influence of the vertical stream. Experiments of Section 5 show that the vertical HMM is more reliable than the horizontal one since vertical strokes are predominant for the shapes considered [28,29]. The coupling proposed here requires that both observation sequences have the same length since streams are synchronized at each time slice: each image column is associated with one row. The observation length is $T = d$ as the character image is previously normalized to a square of size $d \times d$ with $d = 28$ pixels.

At each time, coupled models are in two states, the state corresponding to the column observation (the *vertical state*) and the state corresponding to the row observation (the *horizontal state*). A transition to the vertical state X_t^1 depends only on the value of the preceding state X_{t-1}^1 like classical left–right HMMs. But a transition to the horizontal state X_t^2 depends on both the value of the preceding state X_{t-1}^2 and the value of the current vertical state X_t^1 . This dependence between the horizontal and the vertical states expresses the dependence of the observations, i.e. between row t and column t . Although row t and column t share only one pixel in common, the whole row and the whole column of pixels may be correlated. The

more they are correlated the higher the probability of observing one column configuration captured by the vertical state, in conjunction with one row configuration captured by the horizontal state. As an example, consider simple shapes on a 5 × 5 grid belonging to two classes: H and O shapes, as shown in Fig. 5. We set the number of states to three and we consider two discrete observation symbols a and b : $Y_t^i = a$ when the number of pixels in column (or row) t is > 3 , else $Y_t^i = b$. For H shapes the long central bar (row observation a) is correlated with short bars (column observation b) in the central area of the image. For O shapes, long bars (row observations a) are correlated with long bars (column observations a) at the top and bottom of the image. The state/observation sequences shown for both models in Fig. 5 express these correlations. For O shapes, when $(X_t^1 = 1, X_t^2 = 1)$ or $(X_t^1 = 3, X_t^2 = 3)$, the probability of observing long bars (a) in both row and column is high. For H shapes, when $(X_t^1 = 2, X_t^2 = 2)$ the long horizontal bar (a) is observed jointly with a short bar (b).

- To obtain the first coupled architecture, called the *state-coupled model* (ST_CPL), we add directed edges between the hidden state nodes of the vertical and horizontal HMMs as shown in Fig. 6a. The parameters of ST_CPL are the CPDs b^i and CPTs A and U . The conditional probability table A capturing the HMM left–right structure of the vertical sequence $\{X^1\}$ can be written:

$$A_{j,k} = P(X_t^1 = k \mid X_{t-1}^1 = j) \quad \forall k, j \in [1, Q],$$

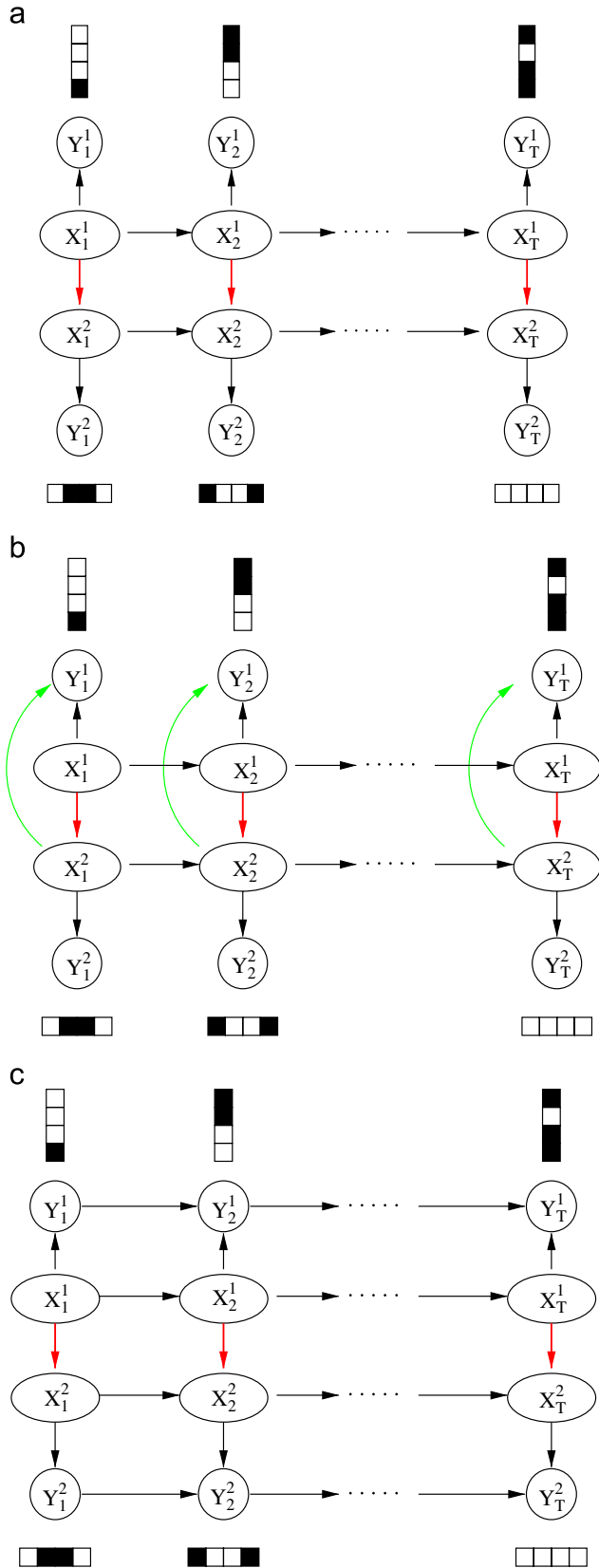


Fig. 6. Coupled architectures represented as DBNS: (a) state-coupled: ST_CPL, (b) general-coupled: GNL_CPL, and (c) auto-regressive coupled: AR_CPL.

where $A_{j,k}$ is a left-right state transition matrix as for classical left-right HMMs. The value k of the current state is either equal to the value j of the preceding state or to $j + 1$. For $t \geq 2$, we write:

$$\begin{cases} U_{j,k,l} = P(X_t^2 = l | X_{t-1}^2 = j, X_t^1 = k) & \forall k, j, l \in [1, Q], \\ b_k^i(y_t^i) = P(Y_t^i = y_t^i | X_t^i = k) & \text{for } i = 1, 2. \end{cases} \quad (3)$$

the CPD b_k^i is a single Gaussian pdf $\mathcal{N}(y_t^i; \mu_k^i, \Sigma_k^i)$ as for basic HMMs. The CPTs U are more complex and of larger size than the CPTs A : The CPTs U are a set of Q matrices (one for each value of X_t^2) of size $Q \times Q$. Left-right transitions are allowed for state transitions within stream 2 while ergodic transitions are allowed for state transitions from stream 1 to stream 2 as shown in Fig. 7(a). All state values increase through time because of the left-right constraint. On the other hand, the value of X_t^2 can be equal to, greater than or less than the value of X_t^1 because of the ergodic property (all state transitions are allowed between X_t^1 and X_t^2). In practise, the values of X_t^2 and X_t^1 follow each other as can be seen in Fig. 7(b) during the decoding of a sample digit, but without predefined order. Sample values from CPT U of the digit four model are shown in Fig. 8. A transition can be made to state $X_t^2 = 8$ from states (X_{t-1}^2, X_t^1) only if X_{t-1}^2 equals 7 or 8 because of the left-right assumption. To transit to state $X_t^2 = 8$, the highest transition probability is 0.9052 from states $(X_{t-1}^2 = 7, X_t^1 = 8)$.

The initial state distributions Π^1 and Π^2 for the vertical and the horizontal streams, respectively, can be written:

$$\begin{cases} \Pi^1(k) = P(X_1^1 = k), & \forall k \in [1, Q], \\ \Pi^2(j, k) = P(X_1^2 = k | X_1^1 = j), & \forall k, j \in [1, Q]. \end{cases} \quad (4)$$

The conditional probability table Π^1 is of length d while CPT Π^2 is of size $d \times d$. Π^2 expresses interdependence of the states of the horizontal stream and those of the vertical stream, at $t = 1$.

- Starting from the previous architecture, the second coupled architecture is obtained by adding an edge from hidden states of the horizontal stream X_t^2 to observation variables of the vertical stream Y_t^1 . This architecture is called the *general coupled model* (GNL_CPL, see Fig. 6b). In the GNL_CPL model, the importance of vertical observations is stressed as they are controlled by the states of both horizontal and vertical streams in the same time slice. The mathematical form of CPTs A , U and Π are identical for ST_CPL and GNL_CPL. The difference lies in the Gaussian CPDs which are written:

$$\begin{cases} b_{j,k}^1(y_t^1) = P(Y_t^1 = y_t^1 | X_t^1 = j, X_t^2 = k) \\ \quad = \mathcal{N}(y_t^1; \mu_{j,k}^1, \Sigma_{j,k}^1), \\ b_k^2(y_t^2) = P(Y_t^2 = y_t^2 | X_t^2 = k) = \mathcal{N}(y_t^2; \mu_k^2, \Sigma_k^2). \end{cases} \quad (5)$$

The form of the distribution of the horizontal observations is the same as in the previous ST_CPL model. The distribution of vertical observations b^1 is a single pdf with mean $\mu_{j,k}^1$ of length d and $d \times d$ covariance matrix $\Sigma_{j,k}^1$. But one needs $Q \times Q$ mean vectors and covariance matrices to account for the cartesian product of states to describe b^1 , and only Q mean vectors and covariance matrices to describe b^2 .

- Last, we construct an auto-regressive coupled architecture (AR_CPL) by coupling the vertical and horizontal AR HMMs as shown in Fig. 6c. Parameters for this model are CPTs A , U and Π which are identical for all coupled models. The CPDs are represented by two Gaussian pdfs, which are defined for each stream i and for $t \geq 2$ by

$$\begin{cases} B_k^i(y_t^i, y_{t-1}^i) = P(Y_t^i = y_t^i | Y_{t-1}^i = y_{t-1}^i, X_t^i = k) & \text{for } i = 1, 2 \\ \quad = \mathcal{N}(y_t^i; \mu_k^i + W_k^i y_{t-1}^i, \Sigma_k^i). \end{cases} \quad (6)$$

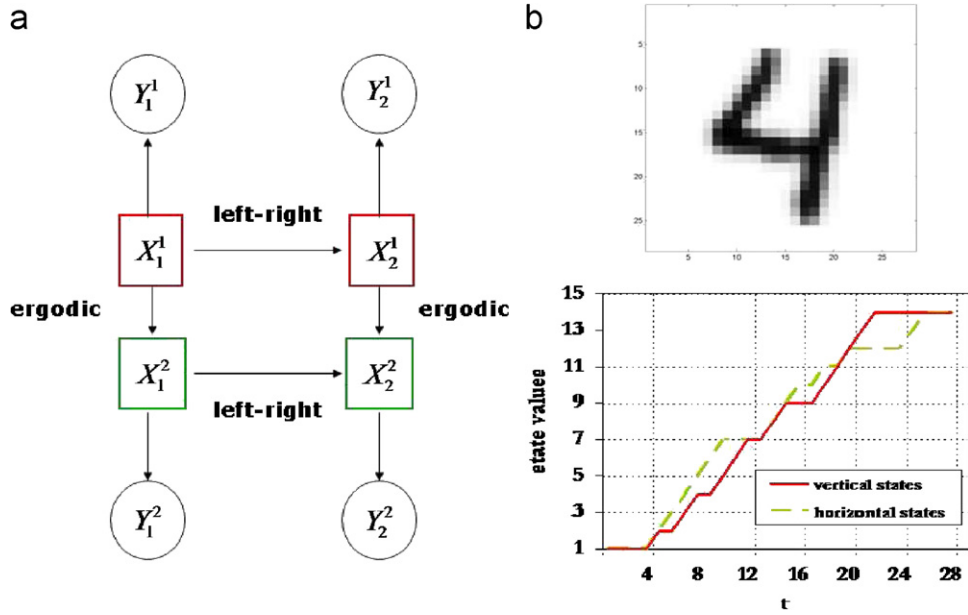


Fig. 7. (a) Types of transitions (ergodic or left-right) allowed between the different state variables. (b) Example of state value sequences for both streams (horizontal and vertical) during decoding of digit 4. Horizontal and vertical state values increase through time (left–right assumption) but at time t the value of the horizontal state can be equal, greater or less than the value of the vertical state.

X_{t-1}^2	X_t^1	$\dots P(X_t^2 = 8 X_{t-1}^2, X_t^1) P(X_t^2 = 9 X_{t-1}^2, X_t^1) \dots$
6	all	0
.	.	.
7	7	0.0343
7	8	0.9052
7	9	0.9050
.	.	.
8	6	0.7570
8	7	0.8820
8	8	0.5460
8	9	0.2021
8	10	0.1772
.	.	.
9	7	0
9	8	0
.	.	.
10	all	0

Fig. 8. Sample values from CPT U which includes state transition probabilities to state X_t^2 from states X_{t-1}^2 and X_t^1 .

As in the case of AR independent models, the mean associated to the current state k is shifted by $W_k^i y_{t-1}^i$ according to the previous observation y_{t-1}^i and the regression matrix W . This model benefits from both the predicting abilities of AR- models and the fusion of observations performed by the coupling through states.

3.3. Complexity

The above architectures, associated to their parameters (CPTs and CPDs) provide character models. In order to limit the number of parameters and to follow the HMM paradigm, we assume that the matrix A is left-right for all models. We also assume that for each CPT U (which can couple up to three states together), state transitions within the same stream are left–right whereas state transitions between different streams can be ergodic (see Section 3.2). Covariance

Table 1
Space and time complexity of independent and coupled models as functions of Q (number of states) and d (length of observation vectors)

Model	Cov + Mean	A	U	W	Decoding
Single HMM	$Qd^2 + Qd$	$Q - 1$			$O(Qd)$
Single AR-HMM	$Qd^2 + Qd$	$Q - 1$		Qd^2	$O(Qd)$
ST_CPL	$2(Qd^2 + Qd)$	$Q - 1$	$2Q^2$		$O(Q^2d)$
GNL_CPL	$(Q^2 + Q)(d^2 + d)$	$Q - 1$	$2Q^2$		$O(Q^2d)$
AR_CPL	$2(Qd^2 + Qd)$	$Q - 1$	$2Q^2$	$2Qd^2$	$O(Q^2d)$

matrices for Gaussian pdfs may be full matrices. The space complexity for all models is given in Table 1 as a function of the common number of states Q and of the length d of observation vectors. Because of character size normalization, the length of the observation sequence is $T = d$. Since the number of states Q is inferior to the length of the observation sequences T , $Q < T$ as in classical HMMs, the coupled model with lowest complexity is ST_CPL: its complexity is of order $O(Qd^2)$, similar to that of the AR_CPL model. Though the AR-coupled model has the most dependence between observations, the GNL_CPL model is the one with highest space complexity: its complexity is of order $O(Q^2d^2)$, since the dimension of the space of conditioning states for the observations $\{Y_t^1\}$ is $Q \times Q$ in this case. The computational time complexity is dominated by inference. Indeed, inference complexity depends both on the size of the cliques in the junction tree and on their number [23,30]. Since all coupled models share the same number of cliques, only clique sizes may be different. Time complexity is of order $O(TQ^{p+1})$ where p is the maximum number of parents for hidden state variables in the original graph. This complexity is reduced by a factor Q in our models because the state space is reduced in the cliques which include the hidden state variables: there is always one hidden variable related to its parent in a left–right transition. Inference time complexity is shown in Table 1 for the decoding (likelihood of an observation set) of a single character. Our time estimation does not include the computation of the observation pdfs for all states in each time-slice.

4. Datasets and training

We apply the DBN architectures to broken character recognition. We first consider artificially broken handwritten digits. Breaks are created within characters according to a degradation model. We then consider real degraded characters. These characters are extracted from an historical printed book [31] and are naturally broken due to ink fading. This section describes the two datasets and the training process.

4.1. Artificially degraded handwritten digits

We start from the MNIST database of handwritten digits [32] which provide separate training and test sets. A training set of 5000 digits is used to train DBN models (see Section 4.3) and the test set includes 10,000 samples. Degradations are obtained by creating breaks within digit strokes. The degradation model we propose shares some similarities with the process related to the 'sensitivity' parameter of Baird's image defect model [33]. Random pixel values are added to original ones within a 5×5 window. Window position is randomly chosen, following a uniform distribution in the 28×28 character image. If the resulting window is centered on a background pixel, the nearest writing pixel is searched and the window is moved toward this pixel. The values added to each pixel within the window are distributed according to a Gaussian pdf, with mean μ and standard deviation σ . The number of windows applied to each character is w . In the following experiments, we set $\sigma = 0.015$, $\mu = 0$ and $w = 0, 1$ or 2 . The value $\mu = 0$ corresponds to changing the pixels within the window to background pixels as normalized pixel values vary from 0 to 1. The value $w = 0$ corresponds to the original handwritten digits. When $w = 1$, one break is created and when $w = 2$, two breaks are created within digit strokes. Fig. 9 shows samples of original and degraded characters.

4.2. Real degraded old printed characters

The set of old printed characters is extracted from the British Library's collection of digitized Renaissance festival books [31]. This collection describes the ceremonies that took place in Europe between the 15th and 17th centuries. The book we have selected is written in French and describes the reception in 1636 of the Duke of Parma in Fontainebleau by King Louis XIII. It was printed in 1656 in Paris and written in Roman type. The set of lowercase characters then included long s which were used instead of usual 's' if occurring at the beginning or in the middle of a word. Characters 'j' and 'k' were not in use and v was often printed instead of u. There were also many ligatures such as (long s + t, or two long s) as can be seen in the sample document in Fig. 10.

Characters from seven pages were extracted and manually labeled. It should be noted that ligature characters such as 'fi' (f + i), 'long s - t' (long s + t), etc. were considered as single characters and were assigned to additional classes. The first five pages were standard pages, well contrasted, whereas the other two were degraded, including many broken characters due to ink fading. This led to two sets of characters: a standard set including 2796 characters from the standard pages and a degraded set including 1216 characters from the degraded pages. Characters were then binarized, normalized to size 20×20 and placed in 28×28 images.² It should be noted that character normalization and image size follow the MNIST paradigm [32] so that white borders are added around character images. In word recognition methods such as in [34], white

² We intend to make this database publicly available (with permission of the British Library).

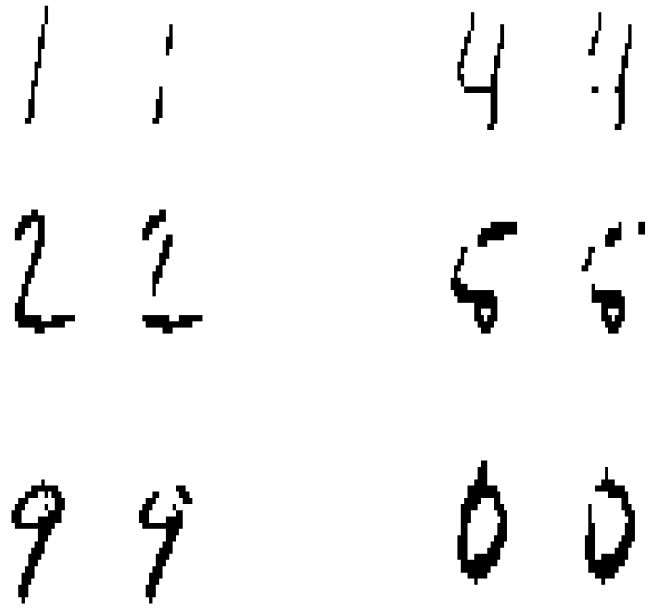


Fig. 9. Pairs of original (left) and degraded (right) characters. Two breaks are created within each digit ($w = 2$).

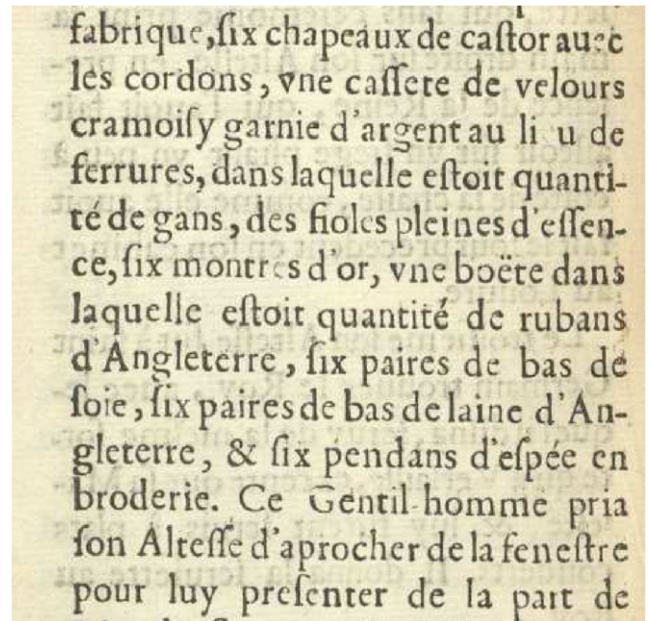


Fig. 10. Sample document from the Renaissance Festival Books collection.

borders are added to training characters in order to deal with intra-word spaces. In other cases, white borders can be removed as well as the states representing each border: models with lower complexity are then obtained by reducing the number of states from Q to $Q - 2$.

Because some classes had very few samples, we selected the classes which had enough samples (around 50) within the first three pages dedicated to training. This led to 16 classes: a, b, c, d, e, i, l, m, n, o, p, r, s, long s, t and u. Sample characters from standard and degraded sets are shown in Fig. 11.

class	a	b	c	d	e	i	l	m
standard	a	b	c	d	e	i	l	m
degraded	a	b	c	d	e	i	l	m
class	n	o	p	r	s	long s	t	u
standard	n	o	p	r	s	f	t	u
degraded	n	o	p	r	s	f	t	u

Fig. 11. Sample old printed characters from standard and degraded pages.

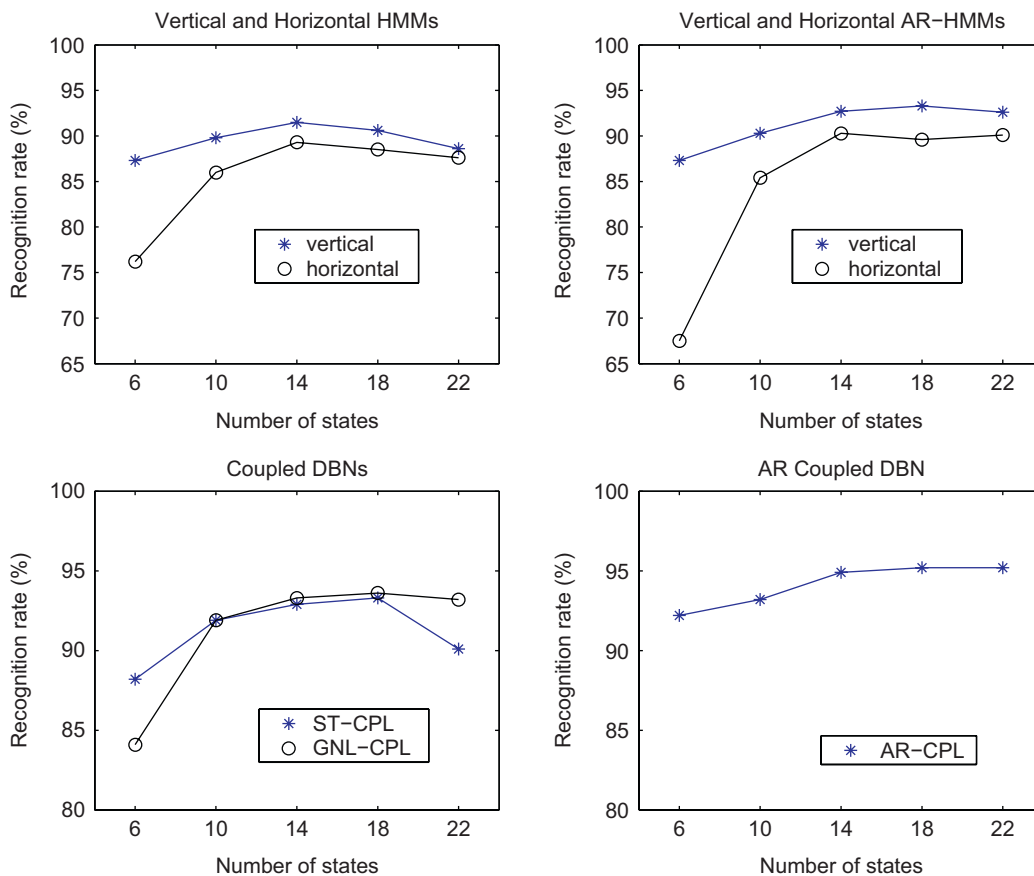


Fig. 12. Performance of independent and coupled models according to the number of states.

4.3. Training and recognition

Observation sequences are obtained by scanning character images from left to right and top to bottom. Characters are first preprocessed by a 3×3 Gaussian mask with standard deviation 0.5. The resulting pixel values are then normalized in $[0., 1.]$. Two observation sequences of length $T = 28$ are obtained from the respective vertical and horizontal streams. All character models share a single DBN architecture but their parameters differ for each class.

Parameters are learnt using the EM algorithm and inference. For independent HMMs, observation parameters are initialized by assigning observations to states linearly. For AR models (independent and coupled), observation parameters are initialized randomly. For all other models, observation parameters are initialized to a common value for all states and each stream i.e. the empirical mean and covariance matrix obtained from the sample data.

The common number of states for hidden variables is denoted by Q . We study the effect of varying Q on the digit recognition task

with a training set of 4000 digits and a test set of 1000 digits. The value of Q ranges from 6-state models to 22-state models. Results in Fig. 12 show that recognition performance increases with Q until reaching a maximum for $Q = 14$ or 18. There is no improvement for values of $Q > 18$ and a slight improvement is obtained for coupled models by increasing Q from $Q = 14$ to 18. The price for this improvement is higher space and time complexity: in the following, we set $Q = 14$ which offers the best compromise between complexity and performance.

Fig. 12 also shows that the best performances are always reached by the AR_CPL model whatever the number of states Q . This shows the superiority of the AR_CPL model over all independent and other coupled models (see also Section 5.1).

For training digit models, we used a subset S of 5000 samples (500 per class) from the MNIST training database. Then, we conducted cross validation experiments in the following way: the subset S was split into $F = 5$ sets of 1000 characters. Each DBN architecture was trained on $F - 1$ sets and tested on the remaining set, F times. Within each model, the parameters yielding the best cross-validation recognition performance were selected for testing. Testing was performed on the 10 000 digits of the MNIST test set.

For training old printed character models, 50 characters per class were selected from the first three standard pages. Then character models were tested on two test sets: a standard test set (test-s) from the two remaining standard pages and a degraded test set (test-d) from the two degraded pages. The standard and degraded test sets include 1009 and 1079 characters, respectively, for the 16 classes considered.

During recognition, each character was assigned to the class with the highest log-likelihood value. We use the *BayesNet* toolbox [35] which provides general MatLab source code for training and inference in static and dynamic Bayesian networks.

5. Experimental results

We evaluate the various DBN architectures, independent and coupled, for the problem of the recognition of degraded characters. Two off-line recognition tasks are considered. We first evaluate DBN architectures for the recognition of artificially broken handwritten digits: their robustness to degradation is evaluated against different degradation parameters. We also test these architectures for the recognition of real degraded printed characters.

5.1. Handwritten digits

5.1.1. Comparison between independent and coupled architectures

Independent and coupled architectures are first tested on the set of artificially broken digits. We consider three levels of degradation: no additional degradation using the original MNIST test set ($w = 0$), one break created within digits ($w = 1$) and two breaks created ($w = 2$). Recognition accuracies for each model are given in Table 2.

For each degradation level, vertical models perform better than horizontal ones within each type of independent models (HMM/AR). This means that columns of character images are more discriminating than rows for handwritten digits. This is also observed for old printed characters: there is a predominance of vertical strokes for these forms of letters and digits [28]. Comparing between independent models, the auto-regressive vertical model performs better than the basic vertical-HMM. This is due to the fact that the basic HMM assumes conditional independence of observation variables with respect to hidden states, whereas the AR model assumes explicit dynamic dependence between observations. For horizontal independent models, performances of the basic horizontal-HMM and the horizontal-AR model are comparable. The prediction of image rows is less efficient than the prediction of image columns.

Table 2

Recognition rates (%) for handwritten digits under different levels of degradation ($w = 0$: no additional degradation, $w = 1$: one break, $w = 2$: two breaks)

Model	$w = 0$	$w = 1$	$w = 2$
Vertical-HMM	90.2	86.9	83.8
Horizontal-HMM	87.4	82.8	75.3
Vertical-AR	93.2	89.8	85.3
Horizontal-AR	87.7	81.6	75.6
ST_CPL	92.4	90.8	87.4
GNL_CPL	93.4	90	86.2
AR_CPL	94.9	93.4	90.9
Combination of HMM scores	93.1	90.6	87
Combination of AR-HMM scores	94.7	91.9	89
SVM	96.1	91.1	85.4

Our results show that coupled models perform significantly better than basic HMMs. Although the horizontal stream is less reliable, its coupling with the vertical stream improves any corresponding single stream representation.

The general coupled model (GNL_CPL) differs from the other coupled models because it uses state–observation relations in addition to state–state relations to express the interdependence of streams. This model requires more observation parameters, but provides little recognition improvement compared with ST_CPL. Achievement of coupling between streams using state–state relations as in ST_CPL and AR_CPL, rather than state–observation relations as in GNL_CPL, leads to more efficient models in terms of complexity and performance. Last, the AR coupled (AR_CPL) model emphasizes the importance of observations through dynamic linking in time.

Coupled architectures behave better than any independent HMM (basic and AR) as the level of degradation increases ($w = 1$ and 2). Moreover the auto-regressive coupled architecture performs best. This is because missing observations (such as in broken characters) may be predicted through auto-regressive models. Coupled architectures may also include at least one uncorrupted stream, horizontal or vertical, within each time slice and thus better cope with missing information.

5.1.2. Comparison with the combination of HMM scores

We also compared coupled models with the weighted combination of HMM scores. The combined score for a pattern given a class model results from the weighted sum of the log-likelihoods (scores) provided by each HMM, vertical and horizontal. The weights α_i , $i = 1, 2$ must satisfy the constraints: $\alpha_i \geq 0$ and $\alpha_2 = 1 - \alpha_1$. Thus, only the value $\alpha = \alpha_1$ dedicated to the vertical HMM needs to be optimized. We search for the optimal α on a validation set of 1000 digits. Fig. 13 shows recognition rates versus α for the combination of basic HMMs. For digits, the maximum is reached for $\alpha = 0.5$. We have observed that log-likelihoods provided by the vertical HMM are in average higher than those provided by the horizontal HMM. Consequently, $\alpha = 0.5$ gives more weight to the vertical HMM.

Results for the test set using the optimal value of α are given in Table 2. The AR-coupled model outperforms the combination of HMM scores whatever the level of degradation. When $w = 0$ (no degradation), the combination of HMM scores performs better than the state-coupled model and worse than GNL_CPL and AR_CPL models. When the level of degradation increases ($w = 1$, $w = 2$), both the ST_CPL and the AR_CPL models perform better than the combination of HMM scores. The performance of individual HMMs as well as the linear combination of them, deteriorates more rapidly as degradation increases than when they are combined in a coupled DBN model.

5.1.3. Comparison with the combination of auto-regressive HMM scores

We can also compare the AR_CPL model with the weighted combination of auto-regressive HMMs. As previously, the combined score for a pattern given a class model results from the weighted sum of

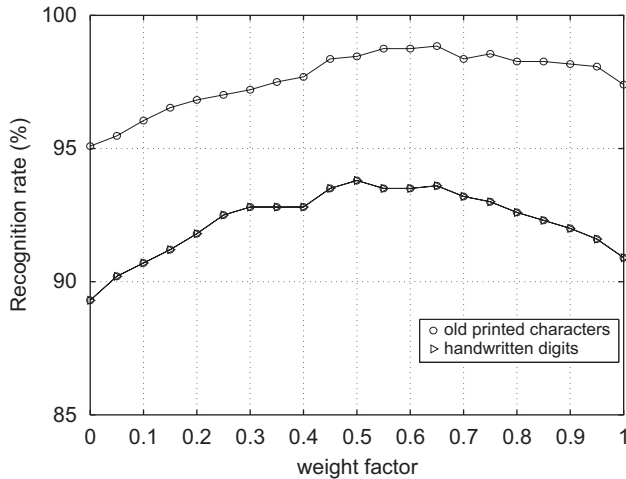


Fig. 13. Combination of HMM scores: recognition rate versus weight factor α .

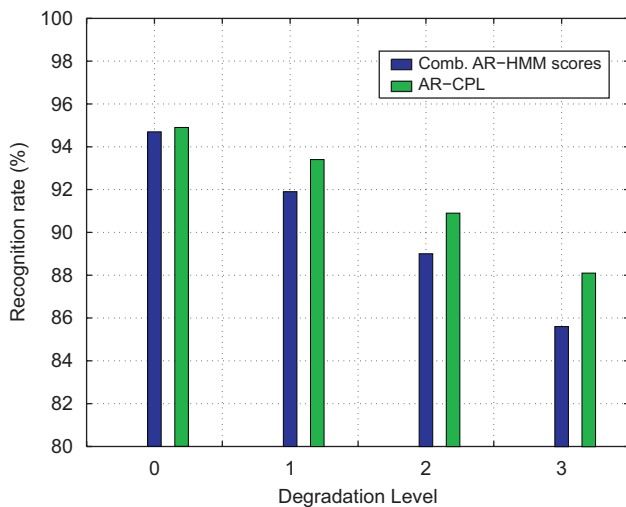


Fig. 14. Comparison of the AR_CPL model with the combination of AR-HMM scores according to degradation.

the scores provided by each AR-HMM, vertical and horizontal. The optimal weight α , searched on the validation set, is equal to $\alpha = 0.45$ for the AR-HMM combination. Recognition accuracies are given in Table 2. The AR-coupled model outperforms the combination of AR-HMM scores whatever the level of degradation. The improvement brought by the AR_CPL model is enhanced for degraded characters ($w > 0$). This is also observed in Fig. 14 where the performances of the AR_CPL model and the combination of the AR-HMMs scores are compared for w ranging from $w = 0$ to 3. An additional level of degradation is provided here: when $w = 3$, three breaks are created within characters. Results in Fig. 14 show that the improvement brought by the AR_CPL model increases as the level of degradation w increases.

5.2. Old printed characters

5.2.1. Model comparison

Recognition accuracies for old printed characters are given in Table 3. Similarly to handwritten digits, the horizontal stream is less reliable but its coupling with the vertical stream increases performance. The vertical-AR model shows some advantage over the basic vertical-HMM on the set of degraded characters but performances are comparable on the standard set. The GNL_CPL model deteriorates

Table 3
Recognition rates (%) for standard and degraded old printed characters

Model	Standard (test-s)	Degraded (test-d)
Vertical-HMM	98.3	93.8
Horizontal-HMM	93.7	88.1
Vertical-AR	97.9	94.5
Horizontal-AR	96.2	91.2
ST_CPL	98.7	95.5
GNL_CPL	98.6	94
AR_CPL	98.8	96
Comb. of HMM scores	98.4	95.4
Comb. of AR-HMM scores	98.7	95.5
SVM	98.4	94.9

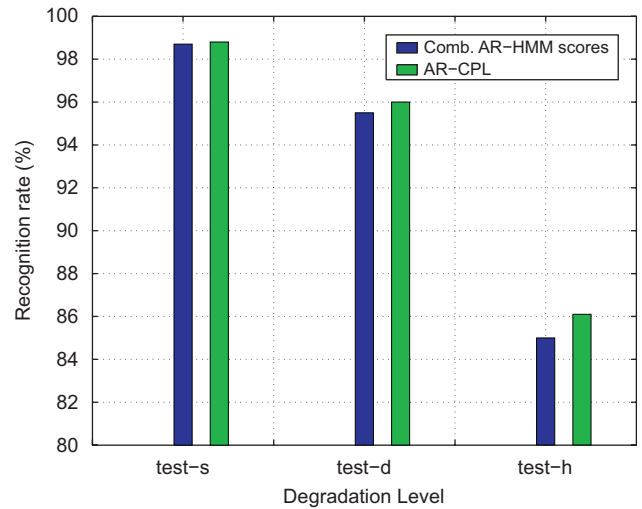


Fig. 15. Comparison of the AR_CPL model with the combination of AR-HMM scores for old printed characters and several degradation levels.

more rapidly on the degraded set because defects in the vertical observation disturb both horizontal and vertical state sequences.

The auto-regressive coupled model (AR_CPL) always performs better than independent models, the state-coupled model and the combination of HMM scores for which the optimal value of $\alpha = 0.65$ was found on a validation set of 1038 characters (Fig. 13). As before, the ST_CPL and AR_CPL coupled architectures better cope with degraded characters (test-d) than independent HMMs (basic and AR) and the combination of HMM scores.

5.2.2. Comparison with the combination of auto-regressive HMM scores

For the combination of auto-regressive HMMs (AR-HMMs), the optimal weight α searched on the validation set is equal to $\alpha = 0.4$. Recognition accuracies are given in Table 3. The AR_CPL model performs better than the combination of AR-HMMs whatever the level of degradation. However, several classifiers have high performance on the set of non-degraded characters (test-s): coupled DBN classifiers and the combination of AR-HMM scores perform accurately on such characters. The improvement brought by the AR_CPL model is enhanced for degraded characters. To highlight this property, an additional set of highly degraded characters is provided by lowering the binarization threshold of degraded characters by 20%. This leads to the test - h set (highly degraded) including highly fading characters. Recognition accuracies are compared in Fig. 15 for all test sets. The improvement brought by the AR_CPL model over the combination of AR-HMM scores increases as degradation increases as seen previously for handwritten digits (see Section 5.1.3).

When characters are broken due to natural fading process or low binarization threshold, the AR_CPL model performs better than the other models. This coupled architecture is thus particularly

convenient for recognizing old printed characters since broken characters are often found in old printed books.

5.3. Comparison with SVM classifier

However, higher accuracies can be achieved on the MNIST digit database with discriminative classifiers such as SVMs as reported in [36,37]. We compare below the influence of defects such as broken characters on DBN and SVM classifiers respectively. The SVM classifier is implemented with the LIBSVM toolbox [38] with a RBF kernel and parameters $C = 26$, $\gamma = 2^5$ as suggested in [37]. SVM recognition accuracies are given in Tables 2 and 3 for handwritten digits and old printed characters respectively, under different levels of degradation.

For handwritten digits and without any additional degradation ($w = 0$), the SVM classifier outperforms other classifiers. When the level of degradation increases ($w = 1$), the AR-coupled model outperforms the SVM classifier. But all coupled models, outperform the SVM classifier in case of high degradation ($w = 2$).

For old printed characters (see Table 3), the SVM classifier obtains slightly lower performances than coupled architectures on the standard data set. On the degraded set (test- d) which includes many broken characters, SVM performance decreases significantly more than that of coupled architectures ST_CPL and AR_CPL. Also, performances of these coupled architectures remain higher than that of the SVM on the degraded test set. This shows that state-coupled and auto-regressive coupled architectures are more robust to degradation than the SVM classifier in case of highly broken characters.

6. Conclusion

We have presented a new approach for off-line character recognition, based on DBN. The modeling consists of coupling two HMMs in various DBN architectures. The observations for these HMMs are the image rows and the image columns, respectively. Interactions between rows and columns are modeled through state transitions or state/observation transitions. This results in finer representations of character images and in improvement of the basic HMM framework.

We first investigated independent HMM and AR models. We showed that vertical models perform better than horizontal ones since columns of character images are more discriminating than rows. Secondly, we coupled these independent models into single models providing better performance than for the non-coupled models, as well as for the combination of the scores of the independent HMMs. We also demonstrated that the coupling through states such as in ST_CPL is more efficient than the coupling from state to observation as in GNL_CPL. The AR-coupled architecture which dynamically links observations in time gives the best recognition results.

We applied this approach to the recognition of handwritten digits and old printed characters. We demonstrated the robustness of this approach in the presence of artificial and real world degradations. Our experiments show that coupled architectures cope better with highly broken characters than both basic HMMs and discriminative methods like SVMs. This is because coupled architectures are able to predict missing information and may provide at least one uncorrupted stream within time slices.

The proposed coupled DBN architectures are thus particularly efficient for the recognition of broken characters. We expect further improvements from an accurate initialization of the parameters.

Acknowledgements

The authors wish to thank the reviewers for their constructive comments. They are also grateful to Chafic Mokbel from Balamand University and Franck Lebourgeois from INSA Lyon for fruitful discussions.

References

- [1] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE* 77 (2) (1989) 257–286.
- [2] R. Plamondon, S. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, *IEEE PAMI* 22 (1) (2000) 63–84.
- [3] C. Bahlmann, H. Burkhardt, The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping, *IEEE PAMI* 26 (3) (2004) 299–310.
- [4] M. Schenkel, M. Jabri, Low resolution degraded document recognition using neural networks and hidden Markov models, *Pattern Recogn. Lett.* 19 (1998) 365–371.
- [5] A. Senior, A. Robinson, An off-line cursive handwriting recognition system, *IEEE PAMI* 20 (3) (1998) 309–321.
- [6] A. Vinciarelli, S. Bengio, H. Bunke, Offline handwriting recognition of unconstrained handwritten texts using HMMs and statistical language models, *IEEE PAMI* 26 (6) (2004) 709–720.
- [7] J.-C. Anigbogu, A. Belaid, Recognition of multifont text using Markov models, In: *Proceedings of the Seventh Scandinavian Conference on Image Analysis*, Aalborg (Denmark), 1991, pp. 469–476.
- [8] H.S. Park, S. Lee, Off-line recognition of large-set handwritten characters with multiple hidden Markov models, *Pattern Recogn.* 31 (1998) 1849–1864.
- [9] N. Arica, F.T. Yarman-Vural, Optical character recognition for cursive handwriting, *IEEE PAMI* 24 (6) (2002) 801–813.
- [10] H. Baird, The state of the art of document image degradation modeling, in: *Proceedings of the Fourth Workshop on Document Analysis Systems, DAS, Rio de Janeiro, 2000*, pp. 1–16.
- [11] A. Whichello, H. Yan, Linking broken character borders with variable sized masks to improve recognition, *Pattern Recogn.* 29 (8) (1996) 1429–1435.
- [12] B. Allier, N. Bali, H. Emptoz, Automatic accurate broken character restoration for patrimonial documents, *IJDAR* 8 (4) (2006) 246–261.
- [13] A. Antonacopoulos, D. Karatzas, Document image analysis for World War II personal records, In: *First International Workshop on Document Image Analysis for Libraries, DIAL 04, Palo Alto, 2004*, pp. 336–341.
- [14] M. Droettboom, Correcting broken characters in the recognition of historical printed documents, in: *Proceedings of Joint Conference on Digital Libraries, JCDL'03, 2003*.
- [15] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, *IEEE PAMI* 3 (20) (1998) 226–239.
- [16] W. Wang, A. Brakensiek, G. Rigoll, Combining HMM-based two pass classifiers for off-line word recognition, in: *Proceedings of ICPR, Quebec, 2002*, pp. 151–154.
- [17] A.J. Elms, S. Procter, J. Illingworth, The advantage of using an HMM based approach for faxed word recognition, *IJDAR* 1 (1998) 18–36.
- [18] K. Hallouli, L. Likforman-Sulem, M. Sigelle, A comparative study between decision fusion and data fusion in Markovian printed character recognition, in: *Proceedings of ICPR, Quebec, 2002*, pp. 147–150.
- [19] X. Xiao, G. Leedham, Signature verification using a modified Bayesian network, *Pattern Recogn.* 35 (2002) 983–995.
- [20] S. Cho, J. Kim, Bayesian Network modeling of hangul characters for on-line handwriting recognition, In: *Proceedings of ICDAR, 2003*, pp. 297–311.
- [21] R. Sicard, T. Artieres, E. Petit, Modeling on-line handwriting using pairwise relational features, In: *Proceedings of IWFHR, La Baule, 2006*.
- [22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, second ed, Morgan Kaufman, Los Altos, CA, 1988.
- [23] G. Zweig, Bayesian network structures and inference techniques for automatic speech recognition, *Comput. Speech Language* 17 (2003) 173–193.
- [24] K. Daoudi, D. Fohr, C. Antoine, Dynamic Bayesian networks for multi-band automatic speech recognition, *Comput. Speech Language* 17 (2003) 263–285.
- [25] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: *Proceedings of the IEEE Conference CVPR 97, 1997*, pp. 994–999.
- [26] N. Friedman, D. Koller, Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks, *Mach. Learning* 2001, pp. 201–210.
- [27] J.D. Hamilton, Analysis of time series subject to changes in regime, *J. Econometr.* 45 (1990) 39–70.
- [28] C. Sirat, *Handwriting and the writing hand*, in: W.C. Watt (Ed.), *Writing Systems and Cognition: Perspectives from Psychology, Physiology, Linguistics, and Semiotics*, Kluwer Academic Publishers, Dordrecht, 1994, pp. 375–459.
- [29] A. Tonazzini, S. Vezzosi, L. Bedini, Analysis and recognition of highly degraded printed characters, *IJDAR* 6 (2003) 236–247.
- [30] G. Bilmes, Dynamic Bayesian multinets, in: *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, Stanford, CA, 2000, pp. 38–45.
- [31] British Library, *British Library Digitised Festival Books*. Available on the web at: (<http://www.bl.uk/treasures/festivalbooks/homepage.html>).
- [32] Y. LeCun, C. Cortes, The MNIST handwritten digit database, 1998. Available on the web at: (<http://yann.lecun.com/exdb/mnist/>).
- [33] H. Baird, Document image defect models, in: H.S. Baird, H. Bunke, K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer, New York, 1992, pp. 546–556.
- [34] S. Procter, A.J. Elms, J. Illingworth, A method for connected hand-printed numeral recognition using hidden Markov models, in: *IEE European Conference on Handwriting Analysis and Recognition*, Brussels, 1998.

- [35] K. Murphy, BayesNet Toolbox for Matlab, 2003. Available on the web at: (<http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html>).
- [36] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, *Pattern Recogn.* 36 (2003) 2271–2285.
- [37] K.-M. Lin, C.-J. Lin, A study on reduced support vector machines, *IEEE Trans. Neural Networks* 14 (2003) 1449–1559.
- [38] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at: (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>).

About the Author—LAURENCE LIKFORMAN-SULEM is graduated in engineering from ENST-Bretagne (Ecole Nationale Supérieure des Télécommunications) in 1984, and received her PhD from ENST-Paris in 1989. She is Associate Professor at TELECOM Paris Tech (former ENST) in the Department of Signal and Image Processing where she serves as a senior instructor in Pattern Recognition and Document Analysis.

Her research area concerns document analysis dedicated to handwritten and historical documents, document image understanding and character recognition. Laurence Likforman and co-researchers won the first place at the ICDAR'05 Competition on Arabic handwritten word recognition.

Laurence Likforman is a founding member of the francophone GRCE (Groupe de Recherche en Communication Ecrite), association for the development of research activities in the field of document analysis and writing communication. She chaired the program committee of the last CIFED (Conference Internationale Francophone sur l'Ecrit et le Document) held in Fribourg (Switzerland) in 2006.

About the Author—MARC SIGELLE was born in Paris in 1954. He graduated from Ecole Polytechnique Paris in 1975 and from Ecole Nationale Supérieure des Télécommunications Paris in 1977. In 1993 he obtained a PhD from Ecole Nationale Supérieure des Télécommunications. He worked first at Centre National d'Etudes des Télécommunications in Physics and Computer algorithms. Since 1989 he has been working in image and more recently in speech processing at Ecole Nationale Supérieure des Télécommunications.

His main fields of interests are restoration and segmentation of signals and images with Markov Random Fields (MRFs), hyperparameter estimation methods and relationship with Statistical Physics. His work has been first devoted to blood vessel reconstruction in angiographic images and then to the processing of remote sensed satellite and synthetic aperture radar images. His most recent interests deal with a MRF approach to image restoration using level sets for Total Variation and its extensions. He is also devoted to speech and character recognition using MRFs and Bayesian Networks. M. Sigelle is IEEE Senior Member since Fall 2003.