Use of PGM for Form recognition

Emilie Philippot
Loria - Nancy University
Nancy, France
emilie.philippot@loria.fr

Abdel Belaïd
Loria - Nancy University
Nancy, France
abdel.belaid@loria.fr

Yoalnde Belaïd
Loria - Nancy University
Nancy, France
yolande.belaid@loria.fr

***Abstract*—This paper addresses the use of PGM (Probabilistic Graphical Model) for form model identification from just few items filled up by an electronic pen. Only the electronic ink is sent to the system without any indication on the form model. Two applications are made in this study: one is related to keynote form classification from its filled fields, while the second application concerns a design modelling problem for the on-line configuration of shower areas. In the former, only indications on the filled fields are sent to the system, while in the latter, the designer send strokes corresponding to the elements designed on the form model. In this application a unique form is proposed to the user to fill up the configuration of his shower area. The PGM is exploited advantageously in both cases translating precisely the relationships between corresponding elements in conditional probabilities, from individual elements up to the complete model constitution.**

***Keywords*-On-line Form, Probabilistic Graphical Model, Keynote modelling, Shower design**

I. Introduction

PGMs are the meeting between graph theory and probability. There are three types of PGM based on their structure: 1) the Directed Acyclic Graph (DAG) with oriented arcs, 2) the Markov Random Field (MRF) with undirected arcs and 3) the chains of graphs that are composed at the same time of directed and undirected arcs.

A. Bayesian Network Definition

A Bayesian network(BN)is a DAG defined by:

- a structure represented by a graph $G = (V, E)$, where V is the set of nodes and E the set of arcs.
- a finite probability space (Ω, Z, p), where Ω represents a non empty finite set, Z the events on Ω and p the probability distribution associated to the graph.
- a set of random variables for each node of G, defined on (ω, Z, p) such as:

$$p(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} p(V_i|C(V_i)) \qquad (1)$$

where $p(V_i)$ is the probability distribution defined for an ordered set of V_i random variables, $C(V_i)$ is the set of the direct fathers of V_i and $p(V_i|C(V_i)$ is the conditional probability between successive nodes in the graph.

X is a BN with respect to G if it satisfies the local Markov property: each variable is conditionally independent of its non-descendants given its parent variables. To develop a BN, we often first create a causal DAG G. We then ascertain the conditional probability distributions of each variable given its parents in G. In many cases, in particular in the case where the variables are discrete, if we define the joint distribution of X to be the product of these conditional distributions, then X is a BN with respect to G [1].

B. Graph structure creation

Among the main possible creation algorithms, three are commonly used: MWST, PC and Naive. We will remind their functioning principle in the following:

MWST: It is part of the family of algorithms based on a score. The goal is to find the tree that goes through all nodes in the network by maximizing a score defined for all possible arcs. The starting point of the algorithm is a set of n trees composed of a single node (as many trees as variables). Then the trees are merged according to the arc weights. The advantage of this algorithm is that all variables are connected and therefore comes into account during the recognition step. The score is calculated using the formula:

$$W_{CL}(X_A, X_B) = \sum_{a,b} P(X_A = a, X_B = b) log \frac{P(X_A = a, X_B = b)}{P(X_A = a)P(X_B = b)} \qquad (2)$$

PC: It is a search algorithm of conditional independence. The starting point is a graph completely connected. Then, for each pair of random variables connected by an arc, we test the existing of a conditional independence using the χ^2 and if so, it removes the corresponding arc. Then, we test the conditional independence for a set of 3, 4 variables and so on until all the conditional independences are removed.

Naive: Its structure does not require learning. It is simply a tree where all variables are directly connected to the result node. There is no interaction between variables.

C. Probability learning

In order to fully specify the BN and thus fully represent the joint probability distribution, it is necessary to specify for each node X the probability distribution for X conditional upon X parents. Often, these conditional distributions include parameters which are unknown and

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.93

must be estimated from data, sometimes using the maximum likelihood approach:

$$\hat{p}(X_i = x_k | pa(X_i) = x_j) = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \quad (3)$$

where $N_{i,j,k}$ is the number of event in the database where the random variable X_i is in the state x_k and its parents are in the configuration x_j.

D. Inference

The inference is to spread the known information to the rest of the BN to change the probabilities of random variables that have not been observed. Initially, the structure of the BN is transformed in a tree using the junction tree algorithm. Then the "message passing" is used to spread information in the tree.

Moralization and Triangulation: A BN is usually transformed into a (decomposable) Markov network for inference. During this transformation, two graphical operations are performed on the DAG of a BN, namely, moralization and triangulation

The moralized counterpart of a directed acyclic graph is formed by connecting nodes that have a common child, and then making all edges in the graph undirected. This is done by looking for cliques. The variables that appear in several cliques are called separators. They will be used during the information propagation in the junction tree.

For two cliques C_1 and C_2 with a common separator S_{12}: $S_{12} = C_1 \bigcap C_2$.

The figure 1 shows an example of moralization where red arcs are added.

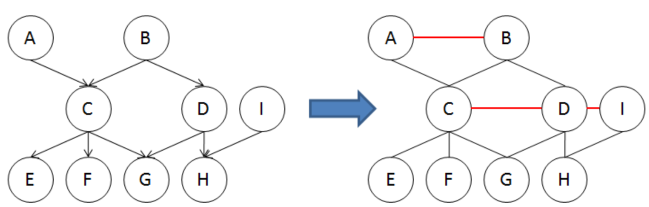

Figure 1. Moralization example

Junction tree construction: The construction follows two steps: structure search and probability calculation

The junction tree structure is built up from a list of cliques with respect of the property of the current intersection defined by: $\forall i, \exists j < i, C_i \cup \bigcup_{l<i}[C_l] \subset C_j$

The table I gives a list of the cliques obtained with the current intersection property.

1	2	3	4	5	6
e,c	f,c	b,c,d	c,d,g	a,b,c	d,i,h

Table I
LIST OF CLIQUES WITH THE CURRENT INTERSECTION PROPERTY

The figure 2 shows the junction tree obtained from the list of cliques of the table I.

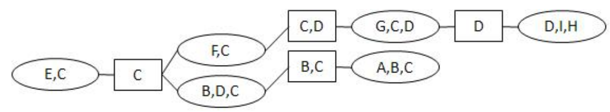

Figure 2. Junction tree example

The second step of the junction tree construction is related to the probability calculation. This step is divided into three parts: initialisation, collection and distribution.

For the initialisation, we use the following formulas:

$$\begin{array}{l} \forall c_i \in C, \text{in the order of the current property:} \\ \Psi^0_{c_i} = \prod_{X \in C_i, X \notin C_j, j<i} p(X | \prod X) \\ \forall s_i \in S, \Psi^0_{s_i} = 1 \end{array} \quad (4)$$

The collection is iterative. Let a clique C_i be for which $\Psi^1_{C_k}$ is performed for all adjacent cliques C_k except for a clique C_j. We performed the potential of S_{ij} and S_j with the formula:

$$\Psi^1_{S_{ij}}(s) = \sum_{C_i \backslash S_{ij}} \Psi^1_{C_i}(c) \quad (5)$$

$$\Psi^1_{C_j} = \Psi^0_{C_j} . \frac{\Psi^1_{S_{ij}}}{\Psi^0_{S_{ij}}} \quad (6)$$

This step is repeated until there is a clique C_i.

For the distribution, we start from the last potential performed in the previous step in order to distribute it to its neighbours that will distribute at their turn to their neighbours, etc., using the following formulas:

$$\Psi^2_{S_{ij}}(s) = \sum_{C_i \backslash S_{ij}} \Psi^2_{C_i}(c) \quad (7)$$

$$\Psi^2_{C_j} = \Psi^1_{C_j} . \frac{\Psi^2_{S_{ij}}}{\Psi^1_{S_{ij}}} \quad (8)$$

1) Propagation: The propagation uses the graph probabilities to perform the initial potential by factoring the cliques and separators.

$$P(V) = \frac{\prod_{c \in C} \Psi_c(V)}{\prod_{s \in S} \Psi_s(V)} \quad (9)$$

where C represents the set of cliques of the junction tree and S the separators.

II. The proposed approach

Our approach deals with form classification. Forms are filled up using an electronic pen and only the electronic ink (i.e. strokes) are sent to the recognition system. The objective of the study is to be able to find the original model of the form by just considering these strokes: their positions and relationships. We use the conditional dependencies between filled form fields as a basis for the BN. As an example, in a form containing the boxes: Ms., Mr. and Miss, we observed that they are never checked at the same time. Moreover, in an another case, the presence of a customer identification number avoids his coordinates filling, which implies the absence of corresponding fields in the form.

To further justify the use of BN in our case on forms, we highlight the unconstrained property of on-line form filling.

We are going to give more details about the BN construction in form identification context.

A. Areas of interest

The form is divided into three areas of interest, represented each one by a specific BN called Bayesian subnet (BsN). These areas correspond to: the customer identity (header), the order (body), and the order validation (footer).

This division into several networks offers advantages:

- facilitation of the network structure training made possible by the reduction number of the variables,
- reassembling of several parts in the same BN
- simplification of the BN updating just by modifying the necessary BN parts

B. BN variables

The random variables of BsN represent the form fields of the corresponding areas of interest. Each one of them is represented by a node. It may have two values: 1 if the field is filled, 2 if it is empty. The arcs represent well the dependencies between the fields.

Conceptually speaking, let F be, a finite set of n forms $(f_1, f_2, ..., f_n)$ represented by a global BN called GBN. A form f_i is composed on 3 sub-forms (BsN): BsN_{i1}, BsN_{i2} and BsN_{i3}. A sub-form BsN_{ij} is composed on C of m fields $(c_{ij1}, c_{ij2}, ..., c_{ijm})$.

For a field c_{ijk} we know its marginal probability $p(c_{ijk})$.

In the global graph regrouping the BsNs, the random variables represent the probability distributions obtained from the BsNs. The arcs define the relationships between them.

Thanks to the Bayes theorem, we perform the probability $p(BsN_{ij})$. From its c_{ijk} we have then for each f_i the probability of its three sub-forms $p(BsN_{ij}|c_{ijk})$ where $k = [1...m]$.

We then use its probabilities to perform $p(f_i)$ using also the Bayes theorem. We obtain:

$$p(f_i) = p(f_i|sf_{ij}) = p(f_i|sf_{ij}|c_{ijk}) \tag{10}$$

III. Experiments

Two experiments were made on two types of forms, one on keynote form modelling and one on design form identification. We will relate in the following the corresponding BNs and their BsNs construction.

A. Keynote Form

The purpose of this application is the classification of an on-line handwritten forms by filling up few fields [2] (see Figure 3).

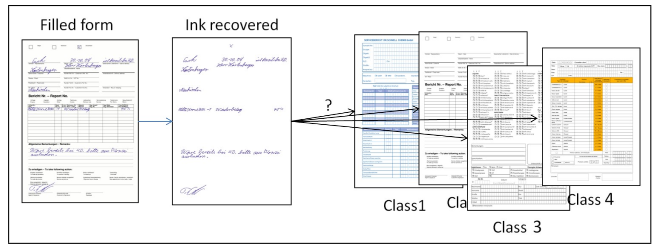

Figure 3. Keynote form problem

If we consider the example of a block address filled in the order form, Figure 4 shows the BsN corresponding to the block obtained using the MWST algorithm, while Table II shows examples of probabilities associated to this network, trained using formula(3).

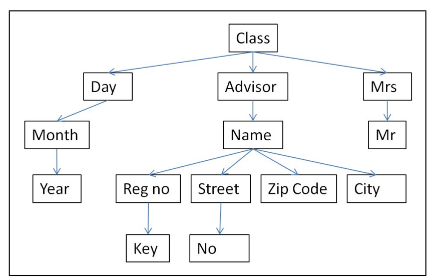

Figure 4. Block address BN obtained by MWST

$\hat{p}(Month\|Year)$			$\hat{p}(Day\|Month)$		
Year \ Month	Filled	Empty	Month \ Day	Filled	Empty
Filled	0.78	0.22	Filled	0.74	0.26
Empty	0.04	0.96	Empty	0.09	0.91

$\hat{p}(Sir\|Madam)$		
Madam \ Sir	Filled	Empty
Filled	0.02	0.98
Empty	0.76	0.24

Table II
Conditional probability distribution

Using the junction tree formulas for the example above, Tables III-A illustrate the values performed.

Once the junction tree is built, we can perform the marginal probability of any of the variables, for example: $P(year = filled) = 0,86$.

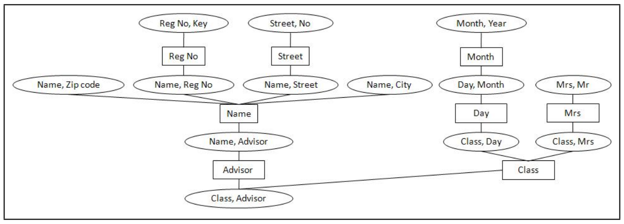

Figure 5. Jonction tree for BsN on figure 4

Initialisation :

$\Psi^0_{C(Month,Year)}$			$\Psi^0_{C(Day,Month)}$		
Year \ Month	Filled	Empty	Month \ Day	Filled	Empty
Filled	0.78	0.22	Filled	0.74	0.26
Empty	0.04	0.96	Empty	0.09	0.91

$\Psi^0_{S(Month)}$	
Filled	Empty
1	1

Collection :

$\Psi^1_{C(Month,Year)}$			$\Psi^1_{C(Day,Month)}$		
Year \ Month	Filled	Empty	Month \ Day	Filled	Empty
Filled	0.61	0.01	Filled	0.64	0.05
Empty	0.03	0.05	Empty	0.01	0.06

$\Psi^1_{S(Month)}$	
Filled	Empty
0.95	0.05

Distribution :

$\Psi^2_{C(Month,Year)}$			$\Psi^2_{C(Day,Month)}$		
Year \ Month	Filled	Empty	Month \ Day	Filled	Empty
Filled	0.61	0.01	Filled	0.67	0.02
Empty	0.03	0.05	Empty	0.08	0.024

$\Psi^2_{S(Month)}$	
Filled	Empty
0.86	0.02

At first, just by studying the chart we can see that the filling of the day depends on the filling of the month, itself depends on the filling of the year. If we look closely at the probabilities, we can find that 91% of the absence of months means that there is no day. This amounts to 96% for year and month. In the case of fields: "Mr." and "Mrs.", the presence of fields: "Mrs." involves in 98 % of the cases, the absence of fields: "Mr.". The absence of the field: "Mrs." implies a filling of the fields: "Mr." in 76% of cases. This highlights the unconstrained side filling.

The database includes four classes, each one consisting of 800 forms. The experiments were performed using the cross validation method. We created four random test bases composed of 600 forms per class for training and 200 for recognition. Each form is divided into three parts: the header, the form body and the footer. For each part a Bayesian network called Bayesian subnet (SRB) is trained. Then the results obtained with the SRB are grouped together in a global Bayesian network that will classify the entire form.

Precision			
Part \ Structure	MWST	PC	Naive
Header	74.59	65.8	91.83
Form Body	88.37	88.37	89.14
Footer	50.03	28.23	52.11
Global	93.63	93.63	95.46
Recall			
Part \ Structure	MWST	PC	Naf
Header	59.53	74.55	90.81
Body	89.61	72.01	85.84
Footer	49.95	50.03	45.93
Global	97.89	90.76	94.97

Table III
EXPERIMENT RESULTS FOR FORM RECOGNITION

B. Design form

Here, the global form represents all the possible configurations for a shower form selection [2]. The user makes his choice by drawing the form parts of interest. Figure 6 shows a selection example. Here also, only the electronic ink is sent to the system. For this application, we take the foundations

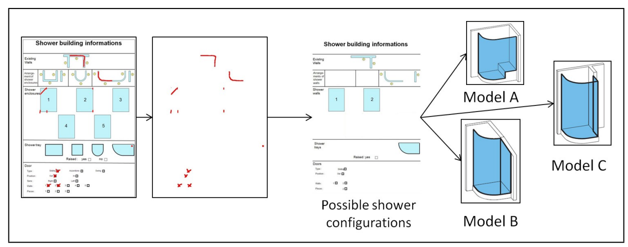

Figure 6. Shower form design

of the approach on the forms. The idea is to define for each space shower model, a form template as a basis for the whole system as well as for learning to recognition, and a general form containing the fields of all models which will be filled by the user.

For recognition, our approach is partly based on the study of the two following dependence observations:

- dependence between the form fields and dependences between components of different parts of the shower area.
- dependence between different parts of the form (i.e. the different parts of the shower area).

For example, the shape of the shower area enclosure will depend on the wall arrangement. Indeed, if the shower area has to be installed in the corner of a bathroom, the model will be composed of up two shower enclosures. Similarly, if the shape of the shower area has an arc, so the shower tray will necessarily have the same arc.

To make best use of these dependencies, we decided to separate the shower area into three distinct parts: one corresponding to the shape of the shower area and consisting of walls and shower enclosures (arrangements), a second part concerning the shower tray (receiver) and finally a part for the door definition(door). For each part, a local BN is trained and then all the BNs are gathered in a global network in order to determine the best model suited to the context. This division provides less complex BNs and thus it is more easy to train. Moreover, this solution also sets the same local BN for several shower area models.

Figure 7 shows an example of the BsN with the part to which it corresponds. It was learned using the algorithm MWST. Tables IV show the associated conditional probability obtained with formula (3).

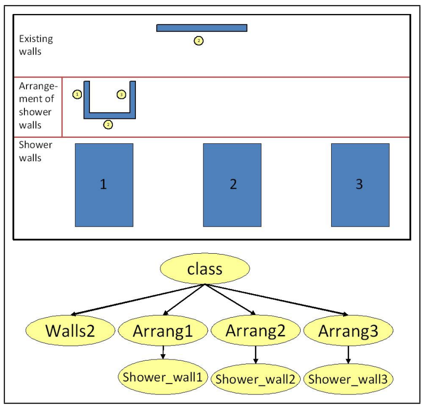

Figure 7. BsN Arrangement

$\hat{p}(Wall2 \mid Class)$		
Wall2 \ Class	True	False
Filled	0.28	0.72
Empty	0.02	0.98

$\hat{p}(Arrang1 \mid Class)$		
Arrang1 \ Class	True	False
Filled	0.38	0.62
Empty	0.11	0.89

$\hat{p}(Shower_wall1 \mid Arrang1)$		
Shower_wall1 \ Arrang1	Filled	Empty
Filled	0.95	0.05
Empty	0.88	0.12

Table IV
CONDITIONAL PROBABILITY DISTRIBUTION FOR BSN

The experiments were performed on a database of 500 forms. The database includes 3 arrangements, two types of receivers and two doors, representing all five models of showers.

	Form			Receiver		Door	
Class	1	2	3	1	2	1	2
Precision	94.1	96.2	96.3	94.7	95	95.9	96.8
Recall	97	96.8	97.7	97.1	96.5	98.3	96

Table V
RECALL AND PRECISION IN % FOR THE THRE PARTS OF THE SHOWER SPACE

Class	1	2	3	4	5
Precision	96.8	92.7	95.4	95.7	96.9
Recall	97.1	95.6	97.9	97.5	95.3

Table VI
RESULTS OF EXPERIMENTS FOR APPLICATION ON SHOWER AREA DESIGN

C. *Probability learning*

In our application, all the variables of the network, except the class, are observed. Indeed, the variable associated with a field always has a value that this field is filled or not. So we have chosen to use the maximum likelihood ([2]) for the learning probability.

IV. CONCLUSION

We have developed and tested two classification systems for unconstrained and on-line forms using two kinds of Bayesian networks. Only the electronic ink was taken into account to discover the original form models. The results are encouraging and pave the way for many opportunities. The modelling done is pretty generic for both application cases which encourages expanding its use easily to several other classes of forms.

The experimentations have been operated with Matlab and BNT (BN Toolbox). The testing computer has a 2.40ghz Processor and 2 giga RAM cadenced at 2.39 ghz. The training time is 2 sec for one net of 6 variables with MWST and 38h 23s 40m with a net with 72 variables. The recognition time of a form is about 32 secondes.

ACKNOWLEDGMENT

This work was conducted under a Cifre grant. We thank Actimage company which has actively contributed to the achievement of this work.

REFERENCES

[1] Neapolitan, R.E.,Learning BNs, Prentice Hall, Upper Saddle River, NJ, 2004.

[2] E. Philippot, Y. Belaïd and A. Belaïd, BNs learning algorithms for on-line form classification. International Conference on Pattern Recognition, Istambul, Turkey, aug. 2019, pp 1981-1984.

[3] E. Philippot, Y. Belaïd and A. Belaïd, Use of Semantic and Physical Constraints in BNs for Form Recognition, International Conference on Document Analysis and Recognition, ICDAR, Bejing, China, pp.946-950, Sept. 2011

[4] D. Doermann and A. Rosenfeld, The processing of form documents. ICDAR, 2:497501, 1993

[5] S. Ramdane, B. Taconet, A. Zahoour and S. Kebairi, Statistical method for an automatic detection of form types, DAS, vol. 1655/1999, pp84-98, 1998