

Bernoulli mixture models for binary images*

Alfons Juan
DSIC, Universitat Politècnica de València, 46071 València, Spain
ajuan@dsic.upv.es

Enrique Vidal
evidal@dsic.upv.es

Abstract

Mixture modelling is a hot area in pattern recognition. Although most research in this area has focused on mixtures for continuous data, there are many pattern recognition tasks for which binary or discrete mixtures are better suited. This paper focuses on the use of Bernoulli mixtures for binary data and, in particular, for binary images. Results are reported on a task of handwritten Indian digits.

Key words: Mixture Models, EM Algorithm, Multivariate Bernoulli Distribution, Binary Data, Indian Digits

1. Introduction

Mixture modelling is a popular approach for density estimation in both supervised and unsupervised pattern classification [8]. On the one hand, mixtures are flexible enough for finding an appropriate tradeoff between model complexity and the amount of training data available. Usually, model complexity is controlled by varying the number of mixture components while keeping the same (often simple) parametric form for all components. On the other hand, maximum likelihood estimation of mixture parameters can be reliably accomplished by the well-known *Expectation-Maximisation (EM)* algorithm.

Although most research in mixture modelling has focused on mixtures for continuous data, there are many pattern recognition tasks for which binary or discrete mixture models are better suited. This paper focuses on the use of (*multivariate*) *Bernoulli mixtures* for binary data and, in particular, for *binary images*. EM-based maximum likelihood estimation of Bernoulli mixtures is known even before the general statement of the EM algorithm in 1977 [3]. In fact, the basic formulae appear in a proposed problem of the classic 1973 book by Duda and Hart [4, pp. 256 and 257], who attribute to Wolfe their derivation in 1970 [4, p. 249]. In spite of being known for more than three decades, Bernoulli

mixtures as such have seldom been assessed in practice. In [7], for instance, a more complex yet closely-related model is successfully tested on a conventional OCR task, but no comparative results are provided for the simpler, pure Bernoulli mixture model. It seems that, at most, this pure model has been only applied to non-conventional applications such as unsupervised modelling of *electropalato-graphic data* [2].

During the past few years, we have found that Bernoulli mixtures are really effective in certain supervised text classification tasks [6, 9]. These tasks, however, can be also considered somewhat non-conventional. Here, our aim is to show that the pure model is very promising also for conventional pattern recognition applications involving binary data and, in particular, binary images. We first briefly review the model and the basic theory on its EM-based maximum likelihood estimation. Then, experimental results are reported on a task of handwritten Indian digits recognition.

2. Bernoulli mixtures

A finite mixture model is a probability (density) function of the form:

$$p(\mathbf{x}) = \sum_{i=1}^I p(i) p(\mathbf{x} | i) \quad (1)$$

where I is the *number of mixture components* and, for each component i , $p(i)$ is its *prior or coefficient* and $p(\mathbf{x} | i)$ is its *component-conditional probability (density) function*. It can be seen as a generative model that first selects the i th component with probability $p(i)$ and then generates \mathbf{x} in accordance with $p(\mathbf{x} | i)$.

A Bernoulli mixture model is a particular case of (1) in which each component i has a D -dimensional Bernoulli probability function governed by its own vector of parameters or *prototype* $\mathbf{p}_i = (p_{i1}, \dots, p_{iD})^t \in [0, 1]^D$,

$$p(\mathbf{x} | i) = \prod_{d=1}^D p_{id}^{x_d} (1 - p_{id})^{1-x_d} \quad (2)$$

Note that this equation is just the product of independent, unidimensional Bernoulli probability functions. Therefore,

* Work supported by the Spanish "Ministerio de Ciencia y Tecnología" under grant DPI2001-0880-CO2-02.

for a fixed i , it can not capture any kind of dependencies or correlations between individual bits.

Since we aim at representing pixels of a binary image as bits governed by Bernoulli distributions, it becomes clear that a single multivariate Bernoulli component will usually be inadequate to cope with the kind of complex pixel dependencies that often underly in binary images of interest. This drawback is overcome when several Bernoulli components are adequately mixed. Consider, for instance, the classical XOR problem. The goal is to have high probability for vectors $(0, 0)^t$ or $(1, 1)^t$ and low probability for the other two bit combinations, $(0, 1)^t$ and $(1, 0)^t$. Although this can not be done using a single-component model, it can be easily done with a two-component mixture model (see Table 1).

\mathbf{x}	$p(\mathbf{x} 1)$	$p(\mathbf{x} 2)$	$p(\mathbf{x})$
$(0, 0)^t$	$0^0 1^1 0^0 1^1 = 1$	$1^0 0^1 1^0 0^1 = 0$	0.5
$(0, 1)^t$	$0^0 1^1 0^1 1^0 = 0$	$1^0 0^1 1^1 0^0 = 0$	0
$(1, 0)^t$	$0^1 1^0 0^0 1^1 = 0$	$1^1 0^0 1^0 0^1 = 0$	0
$(1, 1)^t$	$0^1 1^0 0^1 1^0 = 0$	$1^1 0^0 1^1 0^0 = 1$	0.5

Table 1. Probability distribution for the XOR problem given by a Bernoulli mixture of two equiprobable components with prototypes $\mathbf{p}_1 = (0, 0)^t$ and $\mathbf{p}_2 = (1, 1)^t$.

As with other types of mixtures, Bernoulli mixtures can be used as class-conditional models in supervised classification tasks. Let C denote the number of supervised classes. Assume that, for each supervised class c , we know its prior $p(c)$ and its class-conditional probability function $p(\mathbf{x} | c)$, which is a mixture of I_c Bernoulli components,

$$p(\mathbf{x} | c) = \sum_{i=1}^{I_c} p(i | c) p(\mathbf{x} | c, i) \quad (3)$$

Then, the optimal Bayes decision rule is to assign each pattern vector \mathbf{x} to a class $c^*(\mathbf{x})$ giving maximum a posteriori probability or, equivalently,

$$c^*(\mathbf{x}) = \arg \max_c \log p(c) + \log p(\mathbf{x} | c) \quad (4)$$

$$= \arg \max_c \log p(c) + \log \sum_{i=1}^{I_c} p(i | c) p(\mathbf{x} | c, i) \quad (5)$$

Figure 1 (left) shows a simple example of binary image classification task. Two classes of boats are considered which only differ in the relative position of their two masts of different lengths. While boats of class 1 have the tall mast in the stern area, those of class 2 have it towards the bow.

Since boats can appear in both ahead or astern directions, when the boat silhouette images are seen as raw collections of binary pixels, only subtle pixel correlations can help distinguish both classes. For class 1, if black pixels are found in the image zone A (Figure 1, right), zone B should also

be black and zones C and D should be white, while if black pixels are found in zone D, then zone C should be black and both zones A and B should be white. Similar, but opposite, pixel correlations hold for images of class 2.

Clearly, these subtle dependencies can by no means be captured by a single multivariate Bernoulli model for each class: as shown in Figure 2, both classes become completely confused. Nevertheless, as with the XOR example previously discussed, the required dependencies can be properly modelled by a two-component mixture per class which, in this simple case, is enough for perfect classification.

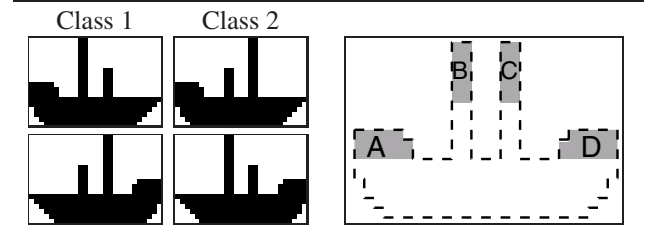


Figure 1. Left: Samples of boats from two classes. Right: Image zones of interest.

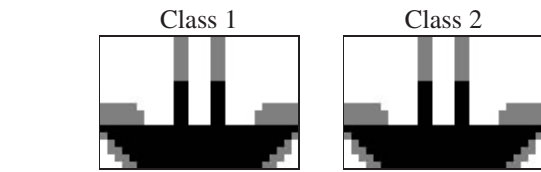


Figure 2. Prototypes of the single-component Bernoulli models for the two boat classes. The probability of each pixel to be 1 is represented as a grey value with "white=0" and "black=1".

3. Maximum likelihood estimation

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be a set of samples available to learn a Bernoulli mixture model. This is a statistical parameter estimation problem since the mixture is a probability function of known functional form, and all that is unknown is a parameter vector including the priors and component prototypes:

$$\Theta = (p(1), \dots, p(I), \mathbf{p}_1, \dots, \mathbf{p}_I)^t. \quad (6)$$

Here we are excluding the number of components from the estimation problem, as it is a crucial parameter for controlling model complexity and receives special attention in section 4.

Following the maximum likelihood principle, the best parameter values maximise the log-likelihood function

$$\mathcal{L}(\Theta | X) = \sum_{n=1}^N \log \left(\sum_{i=1}^I p(i) p(\mathbf{x}_n | i) \right). \quad (7)$$

In order to find these optimal values, it is useful to think of each sample \mathbf{x}_n as an *incomplete* component-labelled sample, which can be completed by an indicator vector $\mathbf{z}_n = (z_{n1}, \dots, z_{nI})^t$ with 1 in the position corresponding to the component generating \mathbf{x}_n and zeros elsewhere. In doing so, a complete version of the log-likelihood function (7) can be stated as

$$\mathcal{L}_C(\Theta|X, Z) = \sum_{n=1}^N \sum_{i=1}^I z_{ni} (\log p(i) + \log p(\mathbf{x}_n|i)), \quad (8)$$

where $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ is the so-called *missing* data.

The form of the log-likelihood function given in (8) is generally preferred because it makes available the well-known *EM* optimisation algorithm (for finite mixtures) [3]. This algorithm proceeds iteratively in two steps. The *E*(xpectation) step computes the expected value of the missing data given the incomplete data and the current parameters. The *M*(aximisation) step finds the parameter values which maximise (8), on the basis of the missing data estimated in the *E* step. In our case, the *E* step replaces each z_{ni} by the posterior probability of \mathbf{x}_n being actually generated by the i th component,

$$z_{ni} = \frac{p(i) p(\mathbf{x}_n|i)}{\sum_{i'=1}^I p(i') p(\mathbf{x}_n|i')} \quad \begin{pmatrix} n = 1, \dots, N \\ i = 1, \dots, I \end{pmatrix}, \quad (9)$$

while the *M* step finds the maximum likelihood estimates for the priors,

$$p(i) = \frac{1}{N} \sum_{n=1}^N z_{ni} \quad (i = 1, \dots, I), \quad (10)$$

and the component prototypes,

$$\mathbf{p}_i = \frac{1}{\sum_{n=1}^N z_{ni}} \sum_{n=1}^N z_{ni} \mathbf{x}_n \quad (i = 1, \dots, I). \quad (11)$$

To start the EM algorithm, initial values for the parameters are required. To do this, it is recommended to avoid “pathological” points in the parameter space such as those touching parameter boundaries and those in which the same prototype is used for all components [2]. From our experience, we recommend using an equiprobable mixture with each prototype computed as a linear combination of a prototype drawn uniformly (from the open unit hypercube) and a randomly chosen training bit vector (for instance, 75% of the former and 25% of the latter). Provided that a non-pathological starting point is used, each iteration is guaranteed not to decrease the log-likelihood function and the algorithm is guaranteed to converge to a proper stationary point (local maximum). Also, for the sake of robustness, it is important to introduce some sort of model smoothing.

4. Experiments

A Bernoulli (mixture) classifier for binary images requires the images to be represented as binary bit vectors of fixed dimension. Therefore, some kind of normalisation is needed in order to establish a fixed geometry for the images considered. Using such a representation, shapes of arbitrary complexity and variability can be easily handled by letting the underlying class-dependent pixel combination features be “discovered” through appropriate mixtures of multivariate Bernoulli components. Our first experiments under this framework involved recognition tasks where a number of stylised binary shapes, such as those illustrated in the examples of Figure 1, were considered. These experiments showed the great potential of Bernoulli mixtures to model complex shapes. Also, preliminary experiments with Latin (Arabic) digit images from the NIST corpus [5], yielded satisfactory results and clearly showed the interest of this approach. Here we consider another OCR task consisting in the recognition of Indian digits, extracted from *courtesy amounts* of real bank drafts. We have used the 10425 digit samples included in the non-touching part of the *Indian digits database* recently provided by CENPARMI [1].

Original digit samples are given as binary images of different sizes (minimal bounding boxes). To obtain properly normalised images, both in size and position, two simple preprocessing steps were applied. First, each digit image was pasted onto a square background whose centre was aligned with the digit centre of mass. This square background was a white image large enough (64×64) to accommodate most samples though, in some cases, larger background images were required. Second, given a size S , each digit image was subsampled into $S \times S$ pixels, from which its corresponding binary vector of dimension $D = S^2$ was built. Figure 3 shows one preprocessed example of each Indian digit ($S = 30$).

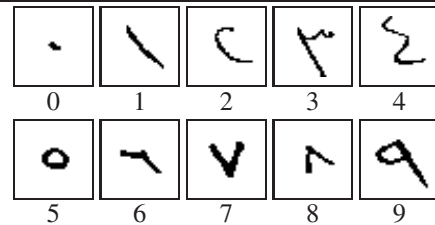


Figure 3. 30×30 examples of each Indian digit.

The standard experimental procedure for classification error rate estimation in the CENPARMI Indian digits task is a simple partition with 7390 samples for training and 3035 for testing (excluding the extra classes *delimiter* and *comma*). Figure 4 shows, for all $S \in \{14, 20, 30\}$ and $I \in \{1, 2, 5, 10, 15, 20, 25\}$, the average error of the I -component Bernoulli mixture classifier tested on the data subsampled at $S \times S$ pixels. Each average was computed

from 50 runs of the standard experimental procedure, each run entailing a randomly initialised EM-based learning of a Bernoulli mixture per class. For simplicity, we did not try classifiers with class-conditional mixtures of different number of components; i.e. an I -component classifier means that a mixture of $I_c = I$ Bernoulli components was trained for each digit c .

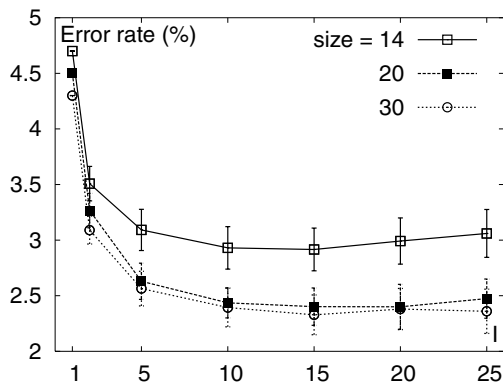


Figure 4. Classification error rate as a function of the number of mixture components in each class (I), for several image sizes. Error bars show standard error.

From the results shown in Figure 4, first note that the curve for $S = 14$ is not as good as those for 20 and 30, which are very similar. Therefore, a subsampling value of 20 can be considered appropriate for this task. Note also that, as expected, the error rate behaviour as a function of I can be described as a smooth concave curve with its minimum at an intermediate value (around $I = 15$). That is, the optimal model complexity is somewhere in between the simplest ($I = 1$) and the most complex ($I \gg$) models.

One of the most salient characteristics of Bernoulli mixture models for binary images is their ease of interpretation. Taking advantage of this characteristic, Figure 5 shows a few steps of the EM-based learning of a 10-component

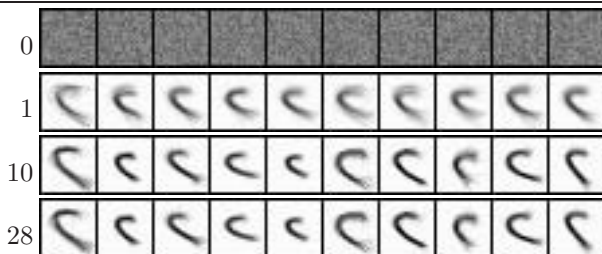


Figure 5. Bernoulli prototypes of digit 2 after iterations 0 (initialisation), 1, 10 and 28 (convergence) of the EM algorithm. The probability of each pixel to be 1 is represented as a grey value with “white=0” and “black=1”.

Bernoulli mixture from digit 2 samples at 20×20 pixels (400-dimensional bit vectors). After the first iteration, all prototypes look more or less the same but, after iteration 10, each prototype has clearly become specialised in a particular writing style.

5. Conclusions

A multivariate Bernoulli mixture model has been proposed for binary image classification. Each image pixel is assumed to be governed by its own scalar Bernoulli distribution for each mixture component. Pixel correlations or dependencies which often underlay complex shapes of interest are captured thanks to the contribution of the different components of the mixture. All the mixture parameters are trained by the standard EM algorithm, leading to very effective classifiers. To assess the capabilities of this model, experiments have been carried out with the recently provided CENPARMI handwritten Indian digits recognition database. Results clearly show the effectiveness of the proposed approach.

References

- [1] Y. Al-Ouali, M. Cheriet, and C. Suen. Databases for recognition of handwritten Arabic cheques. *Pattern Recognition*, 36:111–121, 2003.
- [2] M. A. Carreira-Perpiñán and S. Renals. Practical identifiability of finite mixtures of multivariate Bernoulli distributions. *Neural Computation*, 12(1):141–152, 2000.
- [3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
- [4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.
- [5] M. D. Garris and R. A. Wilkinson. Handwritten segmented characters database. Technical Report Special Database 3, NIST, Feb. 1992.
- [6] J. González, A. Juan, P. Dupont, E. Vidal, and F. Casacuberta. A Bernoulli mixture model for word categorisation. In *Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volume I, pages 165–170, Benicàssim (Spain), May 2001.
- [7] J. Grim, P. Pudil, and P. Somol. Multivariate Structural Bernoulli Mixtures for Recognition of Handwritten Numerals. In *Proc. of the ICPR 2000*, volume 2, pages 585–589, Barcelona (Spain), Sept. 2000.
- [8] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Trans. on PAMI*, 22(1):4–37, 2000.
- [9] A. Juan and E. Vidal. On the use of Bernoulli mixture models for text classification. *Pattern Recognition*, 35(12):2705–2710, Dec. 2002.