Title : will be set by the publisher
Editors : will be set by the publisher
EAS Publications Series, Vol. ?, 2016

MODELLING STRUCTURED DATA WITH PROBABILISTIC GRAPHICAL MODELS

Florence Forbes[1]

Abstract. Most clustering and classification methods are based on the assumption that the objects to be clustered are independent. However, in more and more modern applications, data are structured in a way that makes this assumption not realistic and potentially misleading. A typical example that can be viewed as a clustering task is image segmentation where the objects are the pixels on a regular grid and depend on neighbouring pixels on this grid. Also, when data are geographically located, it is of interest to cluster data with an underlying dependence structure accounting for some spatial localisation. These spatial interactions can be naturally encoded via a graph not necessarily regular as a grid. Data sets can then be modelled via Markov random fields and mixture models (e.g. the so-called MRF and Hidden MRF). More generally, probabilistic graphical models are tools that can be used to represent and manipulate data in a structured way while modeling uncertainty. This chapter introduces the basic concepts. The two main classes of probabilistic graphical models are considered: Bayesian networks and Markov networks. The key concept of conditional independence and its link to Markov properties is presented. The main problems that can be solved with such tools are described. Some illustrations are given associated with some practical work.

1 Introduction

Graphical models are used in various domains including machine learning and artificial intelligence, computational biology, statistical signal and image processing, communication and information theory, and statistical physics to name a few. Probabilistic graphical models refer to a set of tools based on correspondences between graph theory and probability theory and that aim at solving mainly two types of important but difficult problems, namely 1) the computation of likelihoods, marginal distributions and modes of distributions in non trivial settings

[1] INRIA Grenoble Rhône-Alpes, Mistis team, 655 avenue de l'Europe, 38335 Montbonnot, France.

DOI: (will be inserted later)

and 2) the estimation of model parameters and model structures from noisy data. In this task, the role of the graphs is to provide a graphical representation of a probability distribution of interest and this with the objective of providing an easier way to deal with this distribution. The graphical representation can help in visualizing the structure of a model and can provide better insights into the model properties. It allows for instance an immediate visualization of conditional independences by inspection of the graph. More generally, complex computations required to perform inference and learning in sophisticated models can be expressed in terms of graphical manipulations. In addition, the framework is quite general in that a number of standard statistical models such as Kalman filters, hidden Markov models, Potts models, can be described as graphical models. The combination of graph theory and probability theory is not providing new models per se but the diagrammatic representation can help in designing and motivating new probabilistic models and also in designing graph based algorithms for their estimation.

In this chapter, we will review the main concepts of probabilistic graphical models. For a more detailed treatment, the interested readers are referred to better and more complete monographs on the subject Koller and Friedman 2009, Murphy 2012. A number of good tutorials are also available on the web, *e.g.*
- `http://www.di.ens.fr/~fbach/courses/fall2014/`,
- `http://cs.nyu.edu/~dsontag/courses/inference15/slides/lecture1.pdf`
- `http://www.cedar.buffalo.edu/~srihari/CSE574/`
as well as recent Moocs: `https://class.coursera.org/pgm/lecture/preview`.

In section 2, we recall the main useful probability notation and concepts. In the sequel, two main classes of probabilistic graphical models are introduced. Section 3 presents the class of directed graphs also referred to as Bayesian networks in which the links have directional meaning. The key concept of conditional independence and its link with Markov properties is presented in section 4. Then the second class of undirected graphical models which contains the famous Markov random fields is presented in section 5. Mixed directed and undirected graphs (*e.g.* chain graphs) will not be covered here. Section 6 presents the main problems that can be solved with such tools considering inference and learning issues. Some illustrations are given with practical work proposed in section 7.

2 Elements of probability theory

Probability theory plays a central part in modern pattern recognition. It can be expressed in terms of two simple equations corresponding to the sum rule and the product rule below. All of the probabilistic inference and learning manipulations amount to repeated application of these two equations.

Let us first recall some notation. Let $X_1, X_2, \ldots, X_n$ be random variables with distribution:

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = p_X(x_1, \ldots, x_n) = p(x) \tag{2.1}$$

where x stands for $(x_1, \ldots, x_n)$. p is also called the probability density function (pdf) of X. Given $A \subset \{1, \ldots, n\}$, we denote the marginal distribution of x_A by:

$$p(x_A) = \sum_{x \in A^c} p(x_A, x_{A^c}). \tag{2.2}$$

Note that the above equation is written for discrete variables but the extension to continuous variables is straightforward using integrals instead of sums. With this notation we can write the conditional distribution as:

$$p(x_A|x_{A^c}) = \frac{p(x_A, x_{A^c})}{p(x_{A^c})} \tag{2.3}$$

The sum rule links the marginal probability distribution function (pdf) of some variable X to the joint probability distribution of (Y, X):

$$\text{Sum rule :} \quad p(x) = \sum_y p(x, y) \ ,$$

while the product rule expressed the joint pdf as a product of conditional and marginal pdfs:

$$\text{Product rule :} \quad p(x, y) = p(x|y)p(y) \ .$$

From these two equations, we can derive Bayes' theorem:

$$\text{Bayes' theorem :} \quad p(y|x) = \frac{p(x|y)p(y)}{p(x)} \ ,$$

where we can also write $p(x) = \sum_y p(x|y)p(y)$.

We also recall the so-called chain rule which can be derived from a repeated application of the product rule :

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1) \ldots p(x_n|x_1, \ldots, x_{n-1}) \ . \tag{2.4}$$

3 Directed graphs and Bayesian networks

Bayesian networks also called belief networks or causal networks are based on directed graphs. The nodes of the graph are the random variables and the edges correspond intuitively to direct influence of a node on another. The graph can be seen as a compact representation of a probability distribution. Let us show how this can be done by considering a first simple example with 3 variables. The chain rule leads to:

$$p(x, y, z) = p(x)p(y|x)p(z|x, y) \tag{3.1}$$

in which we observe 3 factors that involve separately only parts of the variables. The underlying directed graph semantics is to associate each variable to a node and to draw an arrow from X to Y whenever X is in a conditioning term for Y. This leads to the graph in Figure 1.

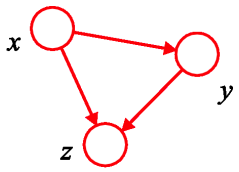


Figure 1. A general 3 variable DAG representing the pdf in (3.1).

In the general case, as an arbitrary joint distribution can always be decomposed using the chain rule (2.4) above, the corresponding graphical representation would correspond to a fully connected graph with n nodes and where each node is linked by an edge to all lower-numbered nodes. It appears that useful information about the specificity of a joint distribution is not so much in the edges but rather relies in the absence of links between variables. This idea leads to the concept of Directed Acyclic Graphs (DAG) also called Bayesian networks. The general factorization property can then be stated as follows. Let $X_1, \ldots, X_n$ be n random variables with distribution $p(x) = p_X(x_1, \ldots, x_n)$.

Definition 3.1 *Let $G = (V, E)$ be a DAG with $V = \{1, \ldots, n\}$. We say that $p(x)$ factorizes in G, denoted $p(x) \in \mathcal{L}(G)$ iff $p(x)$ is of the form:*

$$\forall x, p(x) = \prod_{i=1}^{n} p(x_i|pa_i) \tag{3.2}$$

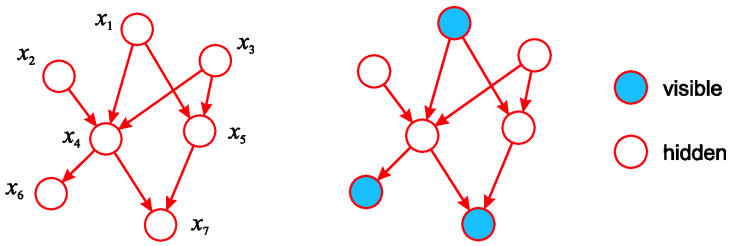
where pa_i stands for the set of parents of the vertex i in G.

We can then observe that a missing link in G implies conditional independence between the corresponding variables. The graph can be used to impose or to account for constraints on the random vector *i.e.* on its distribution p.

Example 3.1 *Some DAGs*

- *Trivial Graphs : Assume $E = \emptyset$,* i.e. *there is no edges. Then we have $p(x) = \prod_{i=1}^{n} p(x_i)$, implying the random variables $X_1, \ldots, X_n$ are independent. Hence variables are independent if they factorize in the empty graph.*
- *Complete Graphs : As already mentioned, the chain rule leads to $p(x) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1})$ which is always true, corresponds to a complete graph with $n(n-1)/2$ edges as we need acyclicity for it to be a DAG. Every random process factorizes in the complete graph.*
- *7 node graph example : As an illustration, let us consider a 7-dimensional vector $(x_1, \ldots, x_7)$. If p admits the following decomposition*

$$p(x_1 \ldots x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5) \tag{3.3}$$



(a) (b)

Figure 2. (a): 7 node DAG with independence constraints corresponding to factorization (3.3). (b) : same model with only part of the variables observed.

it can be translated into the graph in Figure 2 (a).

Example 3.2 *Hidden variables: Variables may be hidden (latent) or visible (observed). In Figure 2 (b), some of the nodes correspond to missing variables which may have a specific interpretation or may be introduced to permit a richer class of distributions.*

- *Mixture of Gaussians: A typical hidden variable model is that of mixtures of Gaussians. A mixture of Gaussians is a linear combination of K Gaussians whose pdf is denoted by $\mathcal{N}(y;\mu_k,\sigma_k^2)$. The mixture pdf is $p(y) = \sum_{k=1}^K \pi_k \mathcal{N}(y;\mu_k,\sigma_k^2)$ with $\sum_{k=1}^K \pi_k = 1$ and $\pi_k \in [0,1]$. We can recover this model by adopting a latent variable viewpoint. Let X be a discrete latent variable that takes values in $\{1,\dots,K\}$ and that describes which component of the mixture generated data point y. Let us consider the model defined by:*

  Conditional distribution of the observed variable: $p(y|X=k) = \mathcal{N}(y;\mu_k,\sigma_k^2)$

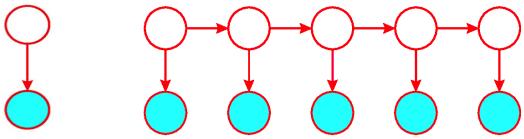
  Prior distribution of the latent variable: $p(X=k) = \pi_k$.

  Marginalizing over the latent variable X, we recover:

$$p(y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y;\mu_k,\sigma_k^2).$$

  In terms of graphical model this corresponds to the simple graph of Figure 3 (a).

- *Hidden Markov Chain : Another example is that of state space models also referred to as Hidden Markov chains or Kalman Filters whose graphical representation is given in Figure 3 (b). In such a setting, frequently the goal is to solve the problem of computing $p(x_i|y_1,\dots y_n)$ where the y_i's are the observed variables and x_i one of the hidden ones.*



(a) (b)

Figure 3. (a): Mixture model. (b): Hidden Markov Chain.

At last, let us say a word about an important but subtile concept of causality. Indeed directed graphs can naturally express causal relationships. Often child variables are observed and the goal is to infer the posterior distribution of parent variables as illustrated in the example of Figure 4 where the result of a blood test is hoped to inform on the presence of a disease. However, note that often statistical analysis leads to the determination of correlation which is a symmetric notion while causality is a directional notion and his therefore much more difficult to infer in a reliable manner. In this chapter we will not address further this issue. Last, it is important to note that not every relationships can be expressed in terms of graphical modes. As a counter-example take three random variables that are pairwise independent, but not fully independent.

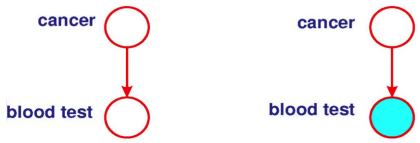


Figure 4. Inferring causal structure from data.

4 Conditional independence and Markov properties

Conditional independence is a key concept in practical applications as we can rarely work with a general joint distribution. The conditional independence between X and Y given a third variable Z is denoted by $X \perp\!\!\!\perp Y \mid Z$ and is characterized by two equivalent formulations:

$$\begin{aligned} X \perp\!\!\!\perp Y \mid Z \quad &\Leftrightarrow \quad p\left(x, y|z\right) = p\left(x|z\right) p\left(y|z\right) \\ &\Leftrightarrow \quad p\left(x|y, z\right) = p\left(x|z\right) = \frac{p\left(x, y|z\right)}{p\left(y|z\right)} \ . \end{aligned}$$

To comment on the difference between dependence and conditional dependence, consider the Traffic jams and snowmen example of Figure 5. In case of heavy snowfalls, traffic jams and snowmen may occur simultaneously and we have no

trouble understanding the possible causal relations between snow and traffic jams and between snow and snowmen. However, ignoring this common cause, one may conclude that traffic jams and snowmen are correlated. But conditionally on snow falls, the size of the traffic jams and the number of snowmen are independent. In other words, the whole link between snowmen and traffic jams is included in the occurrence of snow falls. The concept of conditional independence is more suited than dependence to capture "direct" dependencies between variables because it potentially remove common effects that are by no means causal.

Figure 5. Traffic jams and snowmen are correlated.

Then a natural question is whether we can determine the conditional independence properties of a distribution directly from its graph. The answer is "yes" via the notion of "d-separation" that stands for directed separation. This extended notion of separation is necessary to account for one subtlety due to the presence of so called head-to-head nodes and the *explaining away* effect as detailed in the next section.

4.1 Reading conditional independence

Conditional independences are readable from a directed graph by inspecting edges as illustrated in the following 3-node examples. Besides the empty graph, leading to independence, and the complete graph that gives no further information than the chain rule, 3 different configurations of 3 nodes are possible.

- Tail-to-head node: it corresponds to the DAG showed in Figure 6. In this configuration we show that we have:

$$p(z|y,x) = \frac{p(x,y,z)}{p(x,y)} = \frac{p(x,y,z)}{\sum_{z'} p(z',x,y)} = \frac{p(x)p(y|x)p(z|y)}{\sum_{z'} p(x)p(y|x)p(z'|y)} = p(z|y)\ , \tag{4.1}$$

which means that X and Z are independent conditionally to Y: $X \perp\!\!\!\perp Z \mid Y$. In the graph, we observe that Y separates X from Z in the sense that the path from X to Z is blocked by Y.

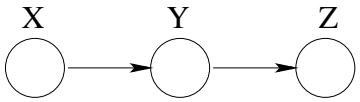


Figure 6. Tail-to-head node: an observed Y separates X from Z.

- Tail-to-tail node: it corresponds to the DAG given in Figure 7. We show that:

$$p(x, y|z)\frac{p(x, y, z)}{p(z)} = \frac{p(z)p(y|z)p(x|z)}{p(z)} = p(x|z)p(y|z), \tag{4.2}$$

which means that $X \perp\!\!\!\perp Y \mid Z$ and an observed Z separates X from Y in the sense that the path from X to Y is blocked by Z.

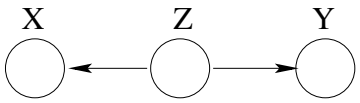


Figure 7. Tail-to-tail node: an observed Z separates X from Y.

In these two first examples, conditional independence is easy to check on the graph as it corresponds to removing the conditioning node and observe whether or not the remaining nodes are connected. However this simple visual rule does not hold in the third case below.

- V-structure or Explaining away: it corresponds to the DAG represented in Figure 8. We can show for this type of graph that:

$$p(x, y) = \sum_z p(x, y, z) = p(x)p(y)\sum_z p(z) = p(x)p(y) \tag{4.3}$$

that is X and Y are independent. But conditionally to Z, we can check that this is not true anymore as $p(x, y|z) \neq p(x|z)p(y|z)$, and we say that an observed Z connects X and Y.

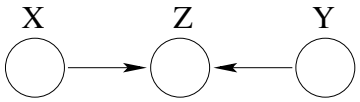


Figure 8. Explaining away: an observed Z connects X to Y.

More generally, we want to answer queries such as, given three subsets A, B and C, is $X_A \perp\!\!\!\perp X_B | X_c$ true? As illustrated above, the notion of separation is not enough in a directed graph and needs to be generalized to that of d-separation as defined in the next section.

4.2 d-separation

Definition 4.1 *Let $a, b \in V$, a chain from a to b is a sequence of nodes, say $(v_1, \ldots, v_n)$ such that $v_1 = a$ and $v_n = b$ and $\forall j, (v_j, v_{j+1}) \in E$ or $(v_j, v_{j+1}) \in E$.*

We can notice that a chain is hence a path in the symmetrized graph, *i.e.* in the graph where if the relation $\rightarrow$ is true then $\leftrightarrow$ is true as well. Assume C is a set that is observed. We want to define a notion of being 'blocked' by this set C.

Definition 4.2 *d-separation*

1. *A chain from a et b is blocked in d if:*
   - *either $d \in C$ and (v_{i-1}, v_i, v_{i+1}) is not a V-structure;*
   - *or $d \notin C$ and (v_{i-1}, v_i, v_{i+1}) is a V-structure and no descendant of d is in C.*
2. *A chain from a to b is blocked if and only if it is blocked at any nodes.*
3. *A and B are said to be d-separated by C if and only if all chains that go from $a \in A$ to $b \in B$ are blocked.*

Note that in other words, the d-separation definition implies that a variable and its non-descendants are conditionally independent given its parents.

Example 4.1 *Markov properties.*

For a Markov chain whose DAG is shown in Figure 9, d-separation gives a direct proof of the Markov property that states that the future is independent on the past given the present.

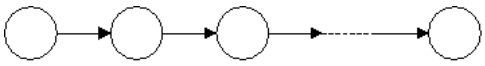


Figure 9. DAG corresponding to a Markov chain

5 Undirected graphs and Markov Random Fields

The second major class of probabilistic graphical models corresponds to undirected graphs. They include Markov random fields also called Markov networks. In this class the graph specifies factorizations of distributions and sets of conditional independence relations which correspond to Markov properties.

5.1 Factorization

Definition 5.1 *Let $G = (V, E)$ be an undirected graph. We denote by $\mathcal{C}$ a set of cliques of G i.e. a set of sets of fully connected vertices. We say that a probability distribution p factorizes in G and denote $p \in \mathcal{L}(G)$ if $p(x)$ is of the form:*

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) \text{ with } \psi_C \geq 0, Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C). \tag{5.1}$$

The functions ψ_C are not probability distributions like in the directed graphical models. They are called potentials. With the normalization by Z of this expression, we see that the function ψ_C are defined up to a multiplicative constant.

Remark 5.1 *The factorization is not unique. We may restrict $\mathcal{C}$ to $\mathcal{C}_{max}$, the set of maximal cliques.*

5.2 Trivial configurations

- **No edges:**

  We consider $G = (V, E)$ with $E = \emptyset$. For $p \in \mathcal{L}(G)$, we get:

  $$p(x) = \prod_{i=1}^{n} \psi_i(x_i) \text{ as } \mathcal{C} = \{\{i\} \in V\} \tag{5.2}$$

  This gives us that $X_1, ..., X_n$ are mutually independent.

- **Complete graphs:**

  We consider $G = (V, E)$ with $\forall i, j \in V, (i, j) \in E$. Then, $\mathcal{C}$ is reduced to a single set V and for $p \in \mathcal{L}(G)$, we get:

  $$p(x) = \frac{1}{Z} \psi_V(x_V) \, . \tag{5.3}$$

  This gives no further information upon the n-sample $X_1, ..., X_n$.

5.3 Separation and conditional independence

When the graph is not complete, information lies in the absence of edges. In contrast to the directed case, conditional independence is given by a simpler graph separation in the undirected case. It follows the following characterization of Markov networks. First, let us specify the Markov property *w.r.t.* a graph G which generalizes the usual Markov property used to characterize Markov chains.

Definition 5.2 *We say that p satisfies the Global Markov property w.r.t. G if and only if for all disjoint subsets $A, B, S \subset V$:*
A and B are separated by $S \Rightarrow X_A \perp\!\!\!\perp X_B | X_S$.

When V is finite, checking conditional independences for all $A = \{i\}, S = N(i), B = E\backslash\{i\}$ is enough where $N(i) = \{j \in V, (i,j) \in V\}$ is the set of neighbors of i in G.

The following theorem makes the connection between conditional independences or Markov properties and factorization.

Theorem 5.1 (Hammersley - Clifford) *If $\forall x,\ p(x) > 0$ then $p \in \mathcal{L}(G) \iff p$ satisfies the global Markov property.*

Note that the positivity constraint can be relaxed to a slighty less constraining condition sometimes called hereditary condition but positivity is however the most common due to its link to Gibbs distributions. Indeed, a distribution $p \in \mathcal{L}(G)$ can also be referred to as a Gibbs distribution. It can be represented using the Boltzmann-Gibbs representation: $\Psi_C(x_c) = \exp(-\mathcal{E}(x_c))$ and $p(x) = \frac{1}{Z}\exp(-\mathcal{E}(x))$ where $\mathcal{E}(x) = \sum_c \mathcal{E}_c(x_c)$ is also called the energy function. The minus sign convention in the exponential is not important but common in statistical physics.

Example 5.1 *Pairwise Markov Random Fields (MRF).*

A pairwise MRF admits as cliques only pairs and singletons so that its energy writes:

$$\begin{aligned}\mathcal{E}(x) &= \sum_{i\in V}\mathcal{E}_i(x_i) + \sum_{(i,j)\in E}\mathcal{E}_{ij}(x_i,x_j) \\ &= \sum_{i\in V}(\mathcal{E}_i(x_i) + 1/2\sum_{j\in N(i)}\mathcal{E}_{ij}(x_i,x_j))\ .\end{aligned}$$

Note the $1/2$ in the second expression above to ensure that each edge in counted only once.

Famous such MRFs include:

- *Ising model on $G = (V,E)$: $p(x;\theta) = \frac{1}{Z}\exp(\sum_{i\in V}\theta_i x_i + \sum_{(i,j)\in E}\theta_{ij}x_i x_j)$ where the $x_i \in \{-1,1\}$ are binary variables.*
- *Potts model on $G = (V,E)$: $p(x;\theta) = \frac{1}{Z}\exp(\sum_{i\in V}\theta_i^T x_i + \sum_{(i,j)\in E}\theta_{ij}x_i^T x_j)$ where the x_i are now K-dimensional indicator vectors which components are 0 except 1 which is 1. This generalizes the Ising model to K-ary variables. We can denote by $\mathcal{X}$ this finite set with K elements. Each of them will be represented by a binary vector of length K with one component being 1, all others being 0, so that $\mathcal{X}$ will be seen as included in $\{0,1\}^{K\times K}$ and its elements denoted by $\{e_1,\ldots,e_K\}$.*

  Parameters θ_i and θ_{ij} are often called respectively the external field and interaction parameters.

A typical application of MRF and hidden MRF in noisy settings, is image segmentation or image region labelling. At each pixel i of an image, a value say

a grey level Y_i is observed and the goal is to recover from the observed image a segmentation into regions. This corresponds to assigning a label x_i for each pixel. This task can also be viewed as a clustering of the pixels into a number of classes. For a binary segmentation into two labels or equivalently into two classes, an Ising model can be used as the hidden MRF, while for a more general segmentation a Potts model is necessary. The corresponding graph is that of Figure 10 whose joint distribution can be written as $p(x, y) = \frac{1}{Z} \prod_{i \in V} \Psi_i(x_i, y_i) \prod_{(i,j) \in E} \Psi_{ij}(x_i, x_j)$.

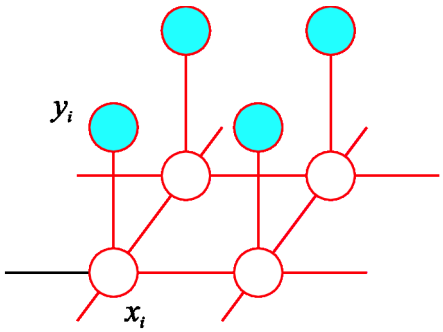
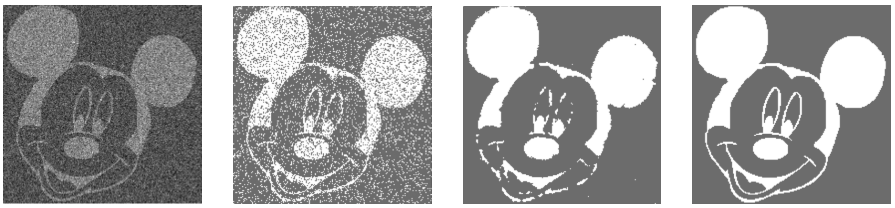


Figure 10. Typical undirected graphical model for image segmentation using an hidden Markov random field (HMRF)



(a) (b) (c) (d)

Figure 11. Illustration of image segmentation: a site/vertex corresponds to a pixel of the image, y_i is the observed grey level at pixel i in image (a). A 2 class segmentation consists in associating a label 0 or 1 to each pixel like in (b) or (c). The ideal ground truth segmentation is shown in (d).

6 Inference and learning

Frequently it is of interest to compute various quantities associated with an undirected graphical model such as the log normalization constant $\log Z$, local marginal distributions $p(x_i)$ or other local statistics, modes and most probable configurations. These tasks may represent challenging computational issues because the complexity often grows rapidly with the graph size and maximum clique size. For

instance, the complexity of computing the normalizing constant for n binary random variables $Z = \sum_{x\in\{0,1\}^n} \prod_{c\in\mathcal{C}} \Psi_c(x_c)$, scales exponentially as 2^n. Inference in graphical models is based on exploiting the graphical structure to find efficient algorithms and to make the structure of these algorithms clear, *e.g.* propagation of local messages around the graph. However, exact inference is not always tractable and a number of approximate inference techniques have been developed. In this chapter, we will focus on estimation in hidden Markov random fields. We assume that we observe a number of measures denoted by $Y = \{Y_i, i \in V\}$ where V is a set of sites that can be associated to vertices of a graph. The goal is to recover from Y a number of hidden variables $X = \{X_i, i \in V\}$ that represent for instance labels and can take a finite number of values. X is assumed to follow a discrete MRF distribution: $p(x) = \frac{1}{Z}\exp(-\mathcal{E}(x))$. The link to the observations Y is specified by a so-called data term that can be written as $p(y|x) = \exp(-\mathcal{E}(y|x))$. It follows that we can compute the conditional distribution which is also a MRF: $p(x|y) = \frac{1}{Z_y}\exp(-\mathcal{E}_y(x))$ with $\mathcal{E}_y(x) = \mathcal{E}(x)+\mathcal{E}(y|x)$. $\mathcal{E}(x)$ acts as a regularization or prior or context term while $\mathcal{E}(y|x)$ acts as a data dependent term.

Let $\tilde{\mathcal{X}}$ denote the set in which X takes values. For general graphical models, not tree-structured, say, $p(x) = \frac{1}{Z}\prod_{c\in\mathcal{C}} \Psi_c(x_c)$, all basic computations are combinatorial for large G and intractable. This is generally the case for the normalization constant $Z = \sum_{x\in\tilde{\mathcal{X}}} \prod_{c\in\mathcal{C}} \Psi_c(x_c)$ and the likelihood, marginals $p(x_j) = \frac{1}{Z}\sum_{x_i, i\neq j} \prod_{c\in\mathcal{C}} \Psi_c(x_c)$, conditionals and modes $\hat{x} = \arg\max_{x\in\tilde{\mathcal{X}}} \prod_{c\in\mathcal{C}} \Psi_c(x_c)$.

Among approximate solutions, we can distinguish two classes, deterministic approaches that involve relaxation, variational approximations (*e.g.* mean field) and stochastic approaches such as Gibbs sampling and simulation methods (Monte-Carlo). In the next section, we detail the variational principle.

6.1 Markov model based segmentation

A typical example in image analysis is the two dimensional lattice with a first-order neighborhood system: for each site, the neighbors are the four sites surrounding it. Let $\mathcal{X}$ be a finite set with K elements denoted by $\{e_1, \ldots, e_K\}$. We define a discrete Markov random field as a collection of discrete random variables, $X = \{X_i,\ i \in V\}$, defined on V, each X_i taking values in $\mathcal{X}$. The joint probability distribution p of X is a Gibbs distribution given by

$$p(x) = Z^{-1}\exp(-\mathcal{E}(x)), \tag{6.1}$$

where $\mathcal{E}$ is the energy function $\mathcal{E}(x) = \sum_c \mathcal{E}_c(x_c)$. The $\mathcal{E}_c$'s are also often called the clique potentials and may depend on parameters, not specified in the notation. $Z = \sum_x \exp(-\mathcal{E}(x))$ is the normalizing factor also called the partition function; $\sum_x$ denotes a sum over all possible values of x. The computation of Z involves all possible realizations x of the Markov field. Therefore, it is, in general, exponentially complex, and not computationally feasible. This can be an issue when using these models in situations where an expression of the joint distribution $p(x)$

is required. We will denote by $\mathcal{D}$ the set of probability distributions on $\tilde{\mathcal{X}}$.

In this section, we focus on Markov model-based image segmentation. Image segmentation involves observed variables (*e.g.* noisy image pixels) and unobserved variables (*e.g.* unknown class assignments) which have to be recovered. The hidden variables are modeled as a discrete Markov random field, X, with distribution p as defined in (6.1) and an energy function $\mathcal{E}$ depending on a parameter $\beta \in \mathcal{B} \subseteq \mathbb{R}$ and henceforth denoted by $\mathcal{E}(x;\beta)$. It is assumed that the observations Y are conditionally independent given the Markov random field X, with conditional distribution parameterized by $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$, where n_θ is the dimension of θ depending on the model under consideration. In the general case, the likelihood of (Y, X) called the complete likelihood, is given by

$$p(y, x\ ;\ \theta, \beta) = p(y \mid x; \theta)\ p(x; \beta). \tag{6.2}$$

It is easy to see that, for such a hidden Markov field model, the conditional field X given $Y = y$ is a Markov field as X is with energy function $\mathcal{E}(x;\beta) - \log p(y \mid x; \theta)$. Hereafter, we will refer to the Markov fields X and X given $Y = y$ as the marginal and the conditional fields.
In image segmentation problems, the question of interest is generally to recover the unknown image x, interpreted as a classification into a finite number K of labels. This classification usually requires values for the vector parameter $\phi = (\theta, \beta)$. If unknown, the parameters are usually estimated in the maximum likelihood sense

$$\hat{\phi} = \text{argmax}_{\phi \in \Phi} \log p(y; \phi), \tag{6.3}$$

where $\Phi = \Theta \times \mathcal{B}$ is the parameter space. This optimization is usually solved by the iterative EM procedure (Dempster et al 1977). Any iteration of the algorithm may be formally decomposed into two steps: given the current value of the parameter ϕ^t, the so-called E-step consists in computing the expectation of the complete log-likelihood knowing the observations y and the current estimate ϕ^t. In the M-step, the parameter is then updated by maximizing this expected complete log-likelihood

$$\phi^{t+1} = \text{argmax}_{\phi \in \Phi} \sum_{x \in \tilde{\mathcal{X}}} \log p(y, x; \phi)\ p(x|y; \phi^t). \tag{6.4}$$

It is known that, under mild regularity conditions, EM converges to the set of the stationary points of the incomplete likelihood $\phi \mapsto p(y; \phi)$ (Wu 1983). As discussed in Csiszar and Tusnady 1984 and Neal and Hinton 1998, EM can be viewed as an alternating maximization procedure of a function F defined, for any probability distribution $q \in \mathcal{D}$, by

$$F(q, \phi) = \sum_{x \in \tilde{\mathcal{X}}} \log \left(\frac{p(y, x; \phi)}{q(x)} \right)\ q(x). \tag{6.5}$$

Starting from the current value $(q^t, \phi^t) \in \mathcal{D} \times \Phi$, set

$$q^{t+1} = \text{argmax}_{q \in \mathcal{D}}\ F(q, \phi^t), \tag{6.6}$$

and

$$\begin{aligned} \phi^{t+1} &= \operatorname{argmax}_{\phi\in\Phi} F(q^{t+1},\phi) \\ &= \operatorname{argmax}_{\phi\in\Phi} \sum_{x\in\tilde{\mathcal{X}}} \log p(y,x;\phi)\; q^{t+1}(x). \end{aligned} \tag{6.7}$$

The first optimization (6.6) has an explicit solution $q^{t+1} = p(\cdot|y;\phi^t)$ so that the optimization in (6.4) and (6.7) are equal. Hence the "marginal" sequence $\{\phi^t\}_t$ of the sequence $\{(q^t,\phi^t)\}_t$ produced by the alternating maximization procedure is an EM path. The maximization (6.7) can also be understood as the minimization of a Kullback-Leibler divergence, up to some convention on p thus justifying the name of alternating minimization procedure often found in the literature (e.g. Csiszar and Tusnady 1984, Byrne and Gunawardana 2005).

There exist different generalizations of EM when the M-step (6.4) is intractable; it can be relaxed by requiring just an increase rather than an optimum. This yields Generalized EM (GEM) procedures (McLachlan and Krishnan 1996; see also Boyles 1983 for a convergence result).

6.2 Variational EM algorithm

Unfortunately, EM (or GEM) is not appropriate for solving the optimization problem (6.3) in Hidden Markov Random Field due to the complex structure of the hidden variables X; the distribution $p(x;\beta)$ is only known up to a multiplicative constant (the partition function) that depends upon the parameter of interest β and the domain $\tilde{\mathcal{X}}$ is too large so that the E-step is intractable. Alternative approaches were proposed and they can be understood as generalizations of the alternating maximization procedures mentioned above : the optimization (6.6) is solved over a restricted class of probability distribution $\tilde{\mathcal{D}}$ on $\tilde{\mathcal{X}}$ and the M-step (6.7) remains unchanged. This yields the Variational EM (VEM) algorithms (Jordan et al. 1998). VEM can also be introduced as resulting from a relaxation of a convex optimization problem; the objective function $p(y;\cdot)$ is re-written as the ratio of two partition functions and VEM results from the approximation of one of them using the notion of conjugate duality in convex analysis (see Wainwright and Jordan 2003 and Wainwright and Jordan 2005 for details).

Byrne and Gunawardana 2005 proved that, under mild regularity conditions, VEM converges to the set of the stationary points of the function F in $\tilde{\mathcal{D}}$. Here again, generalizations of VEM can be defined by requiring an increase rather that an optimum in the M-step (6.7) thus defining generalized VEM procedures. These relaxation methods are part of the Generalized Alternating Minimization procedures (Byrne and Gunawardana 2005).

The most popular form of VEM is the case when $\tilde{\mathcal{D}}$ is the set of the independent probability distributions on $\tilde{\mathcal{X}}$ so that $q^{t+1}(x)$ is a factorized distribution $\prod_{i\in V} q_i^{t+1}(x_i)$. Optimizing (6.6) with regards to $q_i^{t+1}(e_k)$, $i\in V$ and $e_k\in\mathcal{X}$ leads

to a fixed point equation:

$$\forall i \in V, \forall e_k \in \mathcal{X}, \log q_i^{t+1}(e_k) = c_i + \sum_{x \in \tilde{\mathcal{X}}} \log p(x|y;\phi^t) \ \{\delta_{e_k}(z_i) \ \prod_{j \neq i} q_j^{t+1}(z_j)\} \quad (6.8)$$

where c_i is the normalizing constant and δ_e denotes the Dirac mass at point e. The Markov property implies that the right-hand side of the equation only involves the probability distributions q_j, $j \in N(i)$. Existence and uniqueness of a solution to (6.8) are properties that have not yet been fully understood and will not be discussed here. We refer to Tanaka 2001 for a better insight into the properties of the (potentially multiple) solutions of the mean field equations. Such solutions are usually computed iteratively (see Wu and Doerschuk 1995 and Ambroise and Govaert 1998, Zhang 1996 and an erratum in Fessler 1998).

Despite the relaxation which may make the summation of the VEM E-step explicit for a convenient choice of $\tilde{\mathcal{D}}$ (*i.e.* the computation of $F(q^{t+1}, \phi)$ in (6.7)), VEM remains intractable for hidden Markov random fields. From (6.2) and (6.7), θ and β are updated independently, given q^{t+1}. Under additional commonly used assumptions on p, θ^{t+1} is computed in closed form. The issue is the update of β since it requires an explicit expression of the partition function or an explicit expression of some related quantities (its gradient for example).

To overcome this difficulty, different approaches have been proposed. The *Mean Field* , *Modal field* and *Simulated Field* algorithms proposed in Celeux et al. 2003 are alternatives to VEM that propagate the approximation q^{t+1} of $p(x|y, \phi^t)$ to $p(x; \beta)$. Another simple presentation of the variational principle can be found in Bishop 2006.

7 Practical work in R: Image segmentation

As mentioned earlier, image segmentation can be seen as a spatial clustering task that can be solved using undirected graphical models (Hidden MRF) on a regular grid. The segmentation can also be performed without accounting for spatial information with a standard EM for mixtures of Gaussians. In this section, we propose a simple segmentation task to illustrate the gain in accounting for interaction. The R commands that can be used to answer the questions below are given in section 8. However, they should not be considered as a model of R implementation that would be much better written and optimized in a genuine R package. At last in section 9, we mention a link to the SPACEM^3 software that can be used for spatial clustering and classification tasks with additional features such as those related to multimodal, high dimensional and partially missing or incomplete data.

7.1 Non spatial segmentation

The file "mickey.dat" contains a 200 x 200 grey level image. Each pixel can take a value between 0 and 255. The pixels are ordered in the file line by line.

Read the file and plot the image.

Plot a 20 class histogram of the pixels grey levels. Use the standard EM algorithm to find the best mixture of two Gaussians that fits the data.

Initially this image was a binary image made of two colors that we will code as 0 and 1. The goal is to recover the original color of each pixel.

Use the result of the previous EM algorithm to partition the pixels into two groups.

7.2 Spatial segmentation.

The image can be seen as a regular 2D grid with a neighborhood structure of order 1 or 2. Use the following Hidden Markov Random Field (HMRF) model with two classes to partition the image into two groups.

For all $i \in [1:n], x_i \in \{0,1\}$,

Data term:

$$p(y|x) = \prod_{i=1}^{n} p(y_i|x_i) \text{ with for } k = 1,0 \;\; p(y_i|x_i = k) = f(y_i|\mu_k, \sigma_k^2)$$

$$\begin{aligned} \text{and } f(y_i|\mu_0, \sigma_0^2) &= \mathcal{N}(y_i|\mu_0, \sigma_0^2) \\ f(y_i|\mu_1, \sigma_1^2) &= \mathcal{N}(y_i|\mu_1, \sigma_1^2) \end{aligned}$$

where $\mathcal{N}(.|\mu, \sigma^2)$ is the density of the univariate Gaussian distribution with mean μ and variance σ^2.

Hidden MRF:

$$p(x) = \frac{1}{Z} \exp(\mathcal{E}(x)) \text{ with } \mathcal{E}(x) = \beta \sum_{(i,j) \in V} (2x_i - 1)(2x_j - 1)$$

Z is the normalizing constant, β is a positive scalar regulating interaction between neighboring pixels.

7.2.1 Inference and learning

Estimate the Gaussian parameters and recover the two class segmentation using,

- a Variational EM algorithm.
- the ICM algorithm.
- a Gibbs sampler (optional).

The β parameter can be first set to a positive value, *e.g.* 0.5. The boundary conditions can also be fixed to simplify the code, *i.e.* to induce a constant number of neighbors for every pixels.

7.2.2 Comparison

Plot and compare the obtained segmentations. What is the effect of β? Of the neighborhood order and structure? What happens when β is large? When β is negative? When two different β values are used for horizontal and vertical neighbors?

7.2.3 External field

Add an external field to the previous model (optional).

8 R commands

The R function detailed in section 8.3 implements a Mean Field approximation of EM for a HMRF with a 2 color Potts model (0,1) and ICM algorithm (which can be seen as a modal field algorithm).

8.1 Input of the R function

imgobs : Matrix of the observed (noisy) image to be segmented into two classes (x=0 or x=1), eg a greylevel image ;
meaninit and varinit are both vectors of size 2 containing resp. initial values for the means and variances of the 2 Gaussian distributions (noise model);
beta : MRF interaction spatial parameter, here beta is fixed by the user and is scalar and the same for all pairs;
maxite: number of iterations, no convergence criterion in this function;
imginit : matrix, either a hard segmentation used for initialize the posterior probabilities or some soft segmentation, here we can for instance run the function with beta=0 (standard EM) and use the estimated parameters and segmentation to initialize.

8.2 Output of the R function

seg: matrix of the MAP segmentation;
mean: estimation of the 2 means;
var: estimation of the 2 variances;
prob1: matrix of the final posterior probabilities of being in class 1.
Note: all variables ending with ICM are similar definition but for the ICM algorithm.

8.3 R function

The VarEMbin function below implements VEM for a binary hidden MRF. Comments are outside the grey blocks and can be removed to obtain a single R code. Some examples are given in the next section.

```
VarEMbin<-function(imgobs, meaninit, varinit, beta, maxite,
imginit){
n_c <- ncol(imgobs)
n_l <- nrow(imgobs)
imglabel<-matrix(0, nrow = n_c, ncol = n_l)
```

ICM:

```
imglabelICM<-matrix(0, nrow = n_c, ncol = n_l)
```

Initialisation:

1) Posteriors. In the binary case, only the probabilities to be in class 1 need to be computed. Boundary conditions are set to 0 (binary case).

```
prob1init <- matrix(0, nrow = (n_c+2), ncol = (n_l+2))
prob1init[2:(n_l+1),2:(n_c+1)]<-imginit
prob1 <-prob1init
```

ICM: prob1 is not a probability but a label.

```
prob1ICM<-matrix(0, nrow = (n_c+2), ncol = (n_l+2))
prob1ICM[2:(n_l+1),2:(n_c+1)]<-imginit
# ok if imginit is a hard clustering image
```

2) Parameters.

```
bimean<-meaninit
bivar<-varinit
```

ICM:

```
bimeanICM<-meaninit
bivarICM<-varinit
```

Mean Field EM:

```
for (ite in 1:maxite){
        #Estep
mean1<-bimean[2]
sd1<-sqrt(bivar[2])
mean0<-bimean[1]
sd0<-sqrt(bivar[1])

        #ICM
mean1ICM<-bimeanICM[2]
sd1ICM<-sqrt(bivarICM[2])
mean0ICM<-bimeanICM[1]
sd0ICM<-sqrt(bivarICM[1])
```

For simplicity, only the interior is updated, borders are set to label 0. Attention boundaries are not taken into account, the number of neighbors is constant (either 4 or 8).

```
for (i in 2:(n_l+1)) {
    for (j in 2:(n_c+1)){
```

2 neighbors (directional):

```
#sumvois1<-2*(prob1[i,(j-1)]+prob1[i,(j+1)])- 2
```

4 neighbors:

```
#sumvois1<-2*(prob1[i,(j-1)]+prob1[i,(j+1)]+prob1[(i-1),j]
+prob1[(i+1),j])-4
```

8 neighbors:

```
sumvois1<-2*(prob1[i,(j-1)]+prob1[i,(j+1)]+prob1[(i-1),j]
+prob1[(i+1),j]+prob1[(i-1),(j-1)]+prob1[(i-1),(j+1)]
+prob1[(i+1),(j-1)]+prob1[(i+1),(j+1)])- 8
```

dnorm with log=TRUE computes the log density

```
temp<-dnorm(imgobs[i-1,j-1],mean0, sd0,log=TRUE)
- dnorm(imgobs[i-1,j-1],mean1, sd1,log=TRUE)
- (2*beta*sumvois1)
prob1[i,j]<-1/(1+exp(temp))
```

ICM:

```
sumvois1ICM<-2*(prob1ICM[i,(j-1)]+prob1ICM[i,(j+1)]
+prob1ICM[(i-1),j]+prob1ICM[(i+1),j]
+prob1ICM[(i-1),(j-1)]+prob1ICM[(i-1),(j+1)]
+prob1ICM[(i+1),(j-1)]+prob1ICM[(i+1),(j+1)])- 8
tempICM<-dnorm(imgobs[i-1,j-1],mean0ICM, sd0ICM,log=TRUE)
- dnorm(imgobs[i-1,j-1],mean1ICM, sd1ICM,log=TRUE)
 - (2*beta*sumvois1)
prob1ICM[i,j]<-(tempICM <0)+0
}}
```

M- step:

```
prob1temp<-prob1[2:(n_l+1),2:(n_c+1)]
n1<-sum(prob1temp)
n0<-n_c*n_l-n1
prob0temp<-1-prob1temp
```

Update the 2 means:

```
bimean[2]<-sum(prob1temp*imgobs)/n1
bimean[1]<-sum((prob0temp)*imgobs)/n0
```

Update the 2 variances:

```
bivar[2]<-sum(prob1temp*(imgobs-bimean[2])^2)/n1
bivar[1]<-sum(prob0temp*(imgobs-bimean[1])^2)/n0
# beta is fixed for now
```

ICM:

```
prob1ICMtemp<-prob1ICM[2:(n_l+1),2:(n_c+1)]
n1ICM<-sum(prob1ICMtemp)
n0ICM<-n_c*n_l-n1ICM
prob0ICMtemp<-1-prob1ICMtemp
```

Update the 2 means:

```
bimeanICM[2]<-sum(prob1ICMtemp*imgobs)/n1ICM
bimeanICM[1]<-sum((prob0ICMtemp)*imgobs)/n0ICM
```

Update the 2 variances:

```
bivarICM[2]<-sum(prob1ICMtemp*(imgobs-bimeanICM[2])^2)/n1ICM
bivarICM[1]<-sum(prob0ICMtemp*(imgobs-bimeanICM[1])^2)/n0ICM
}
```

Compute final MAP:

```
imglabel[prob1temp>0.5]<-1
imglabelICM<-prob1ICM[2:(n_l+1),2:(n_c+1)]
list(seg=imglabel,mean=bimean,var=bivar,prob1=prob1temp,
segICM=imglabelICM)
}
```

8.4 Example of use

```
mick<-matrix(scan("mickey.dat"), ncol=200, byrow=T)
image(mick)
```

The last command above plots the image in Figure 11 (a).

Plot the histogram of the data imgobs=mick to find initial values for mean and var:

```
hist(mick)
```

1) Run the algorithm with beta=0 (VEM corresponds then to regular EM) and use the output label and parameters to set meaninit, varinit, imginit:

```
resmick<-VarEMbin(mick,c(90,170),c(400,1064),0,10,
matrix(0,200,200))
image(resmick$seg) # to check if ok
imginit<-resmick$seg
meaninit<-resmick$mean
varinit<-resmick$var
```

The plotted image is close to the one in Figure 11 (b). The absence of spatial interaction implies the classification of each pixel in one of the two classes independently. As a result, it remains a salt and pepper effect in the obtained binary segmentation.

2) Run VEM with $\beta = 0.4$ and 10 iterations and initial values $meaninit = c(85.14, 148.71)$ and $varinit = c(402.97, 1234.07)$:

```
resmick<-VarEMbin(mick,meaninit,varinit,0.4,10,imginit)
image(resmick$seg)
image(resmick$segICM)
# almost the same for high beta
```

The obtained segmentation is shown in Figure 12 (a). Note that β should not be too above the phase transition value (0.36 for 4 neighbors, about 0.88 for 8). The run below produces a segmentation for $\beta = 1$. The effect is visible on Figure 12 (b): when the spatial interaction is too strong, pixels tend to all agree to be in the same class, which results in a almost monocolor segmentation.

```
resmick<-VarEMbin(mick,meaninit,varinit,1,10,
matrix(0, 200,200))
image(resmick$seg)
#almost monocolor
```

Some experiments to make:
Negative β:

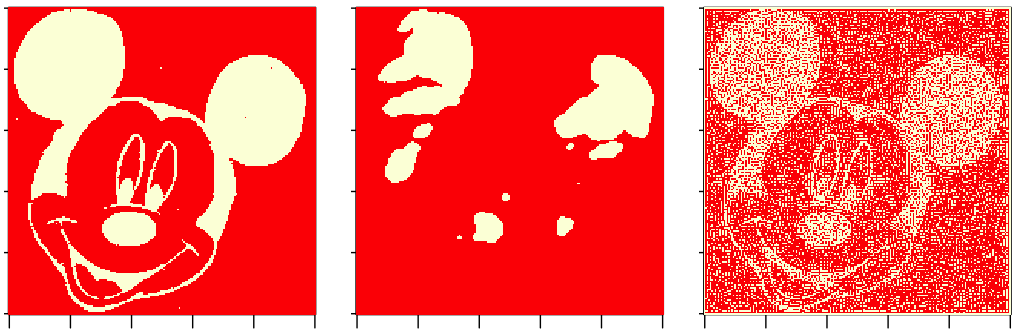
```
resmick<-VarEMbin(mick,meaninit,varinit,-0.15,50,
matrix(0,200,200))
# or
resmick<-VarEMbin(mick,meaninit,varinit,-0.18,10,
matrix(0,200,200))
```

The obtained segmentation is shown in Figure 12 (c). All previous segmentations were made with 8 neighbors on a 2D grid. Try with only 2 or 4 neigbors by changing the lines referring to the computation of *sumvois1* in the code.

For more sophisticated methods and applications, we include in the next section a reference to the SPACEM^3 software that implements a number of spatial clustering methods.

9 The SPACEM^3 software

The SpaCEM^3 software is dedicated to Spatial Clustering with EM and Markov Models. It proposes a variety of algorithms for supervised and unsupervised classification of multidimensional and spatially-located data. The main techniques use the EM algorithm for soft clustering and Markov Random Fields (MRF) for spatial modelling. The learning and inference parts are based on developments in mean field-like approximations. Its applications range from image segmentation



(a) (b) (c)

Figure 12. Output segmentations of the practical work: (a) VEM with 8 neighbors and $\beta = 0.4$; (b) VEM with 8 neighbors and $\beta = 1$ and (c) VEM with 8 neighbors and $\beta = -0.18$.

(e.g. tissue detection in MRI, retrieval of planet surface properties from hyperspectral satellite images) to gene clustering (e.g. biological module detection), remote sensing and mapping epidemics of ecological species. The main functionalities of the program include:

- Model-based unsupervised segmentation including the standard EM algorithm for mixtures and Hidden Markov Random Field models
- Model selection for the Hidden Markov Random Field model
- Simulation of commonly used Hidden Markov Random Field models
- Simulation of independent Gaussian noise for noisy images
- Non standard Markov models including various extensions of the Potts model and triplet Markov models
- Additional treatment of very high dimensional data using dimension reduction techniques within a classification framework
- Models and methods allowing supervised classification with original learning and test steps
- Integrated treatment of missing observations
- Summary statistics of the data and visualization

The interface is shown in Figure 13 with an example of hyperspectral image segmentation into 4 classes. The data to be segmented are spatially localized 184-dimensional spectra on the Mars's surface. More details on the models and algorithms implemented and on possible applications of the software can be

found in Forbes and Peyrard 2003, Celeux et al. 2003 , Blanchet and Forbes 2008, Blanchet et al 2009, Blanchet and Vignes 2009 and Vignes and Forbes 2009.

The software can be downloaded at `http://spacem3.gforge.inria.fr/`.

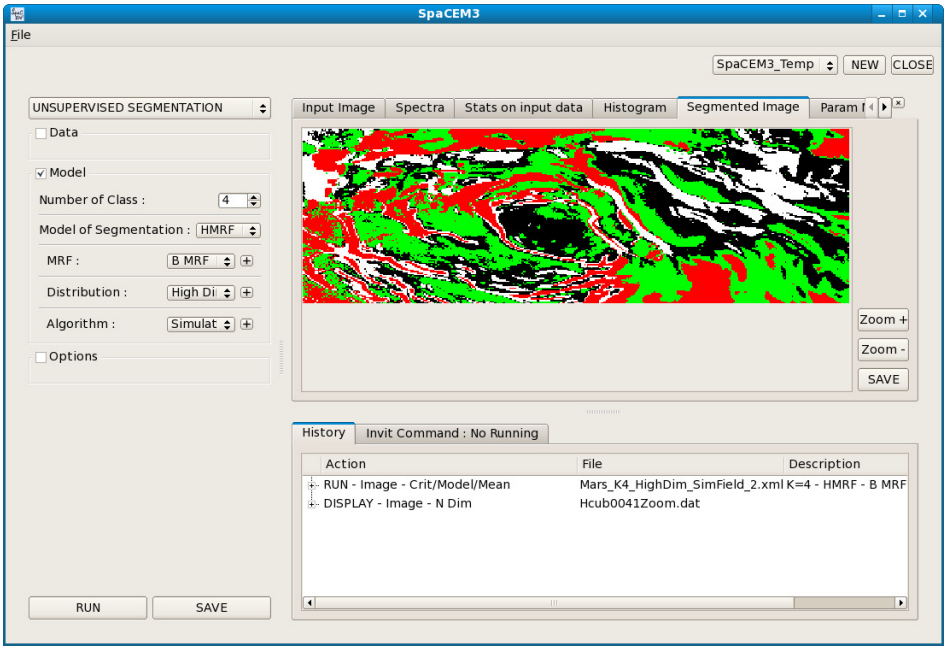


Figure 13. SPACEM^3 interface: illustration of an hyperspectral image segmentation.

References

C. Ambroise and G. Govaert. Convergence proof of an EM-type algorithm for spatial clustering. *Pattern Recognition Letters*, 19:919–927, 1998.

C. M. Bishop. Pattern recognition and machine learning. Springer, 2006.

J. Blanchet and F. Forbes. Triplet Markov fields for the supervised classification of complex structure data. IEEE PAMI, 30(6), pp. 1055-1067, 2008.

J. Blanchet, F. Forbes, S. Chopart and L. Azizi. Le logiciel SpaCEM3 pour la classification de donnees complexes. La Revue MODULAD, 40, pp.147-166, 2009. [in French]

J. Blanchet and M. Vignes,. A model-based approach to gene clustering with missing observation reconstruction in a Markov Random Field framework. Journal of Computational Biology, 16(3), pp. 475-486, 2009.

R.A. Boyles. On the convergence of EM algorithms. *J. Roy. Statist. Soc. Ser. B*, 45(1):47–50, 1983.

W. Byrne and A. Gunawardana. Convergence theorems of Generalized Alternating Minimization Procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.

G. Celeux, F. Forbes, and N. Peyrard. EM procedures using mean-field approximations for Markov model-based image segmentation. *Pattern Recognition*, 36:131–144, 2003.

I. Csiszar and G. Tusnady. Information geometry and alternating minimization procedures. *Stat. & Dec.*, (1):205–237, 1984. Supp. Iss.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977.

J.A. Fessler. Comments on "The convergence of mean field procedures for MRF's". *IEEE Transactions on Image Processing*, 7(6):917, 1998.

F. Forbes and N. Peyrard. Hidden Markov Random Field selection criteria based on Mean Field-like approximations. IEEE trans. on Pattern Analysis and Machine Intelligence, 25(8), 2003.

M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 105–162. 1998.

D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT press, 2009.

G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1996.

K. Murphy. Machine learning, a probabilistic perspective. MIT press, 2012.

R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. 1998.

T. Tanaka. Information geometry of mean-field approximation. In M. Opper and D. Saad, editors, *Advanced Mean Field Methods*, chapter 17. MIT Press, 2001.

M. Vignes and F. Forbes. Gene clustering via integrated Markov models combining individual and pairwise features. IEEE/ACM trans. on Computational Biology and Bioinformatics, 6(2), pp.260-270, 2009.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Department of Statistics, 2003.

M.J. Wainwright and M.I. Jordan. A variational principle for graphical models. In S. Haykin, T. Principe, T. Sejnowski, and J. McWhirter, editors, *New Directions in Statistical Signal Processing*, chapter 11. MIT Press, 2005.

C-H. Wu and P. C. Doerschuk. Cluster Expansions for the Deterministic Computation of Bayesian Estimators Based on Markov Random fields. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 17(3):275–293, 1995.

C.F.J. Wu. On the convergence properties of the EM algorithm. *Ann. Stat.*, 11(1):95–103, 1983.

J. Zhang. The convergence of mean field procedures for MRF's. *IEEE Transactions on Image Processing*, 5(12):1662–1665, 1996.