Undirected Graphical Models

Christophe Ambroise

12/23/2020

Section 1

Markov Random Fields

Problem of directed graph

- Several graphs can induce the same set of conditional independences .
- Is it possible to associate to each graph a family of distribution so that conditional independence coincides exactly with the notion of separation in the graph?

Conditional independence properties of UGMs

UGMs define CI relationships via simple graph separation:

Global Markov property for UGMs

for sets of nodes A, B, and C, we say $x_A \perp_G x_B | x_C$ iff C separates A from B in the graph G.
When we remove all the nodes in C, if there are no paths connecting any node in A to any node in B, then the CI property holds.

Other Markov properties

Local Markov property

A variable is conditionally independent of all other variables given its neighbors:

$$X_v \perp X_{V \setminus \mathrm{N}[v]} \mid X_{\mathrm{N}(v)}$$

Pairwise Markov property

Any two non-adjacent variables are conditionally independent given all other variables:

$$X_u \perp X_v \mid X_{V \setminus \{u,v\}}$$

Links

It is obvious that global Markov implies local Markov which implies pairwise Markov.

Undirected Graph

A Markov Random Field

Given an undirected graph a Markov Random Field is associated with probability distributions obeying the global Markov property:

$$X_A \perp\!\!\!\perp X_B | X_C$$

The distribution $p(x)$ of a Markov Random Field

is given by the Hammersley-Clifford Theorem (1971)

$$p(x) = \frac{1}{Z} \prod_{c \in \mathrm{cl}(G)} \Psi_c(x_c|\theta_c)$$

where $\mathrm{cl}(G)$ is the set of all cliques of G, and

$$Z \triangleq \sum_x \prod_{c \in \mathrm{cl}(G)} \Psi_c(x_c|\theta_c)$$

is the "partition function".

Hammersley-Clifford Theorem

A distribution p (with $p(x) > 0$) for all x satisfies the Global Markov property for graph G iff it is a Gibbs distribution associated with G

$$p(x) = \frac{1}{Z} \prod_{c \in \mathrm{cl}(G)} \Psi_c(x_c|\theta_c)$$

It is easy to check the global Markov property if the distribution is Gibbs but more difficult to do the reverse.

Hammersley-Clifford Theorem

The idea of the demonstration consists in considering a generic form subject to the global global Markov property:

Consider $Q(x) = \ln \frac{p(x)}{p(\mathbf{0})}$ and its unique decomposition on the interaction space (of the n variables)

$$Q(x) = \sum_i x_i G_i(x_i) + \sum_{i<j} x_i x_j G_{ij}(x_i, x_j) + \cdots + x_1 x_2 \ldots x_n G_{12\ldots n}(x_1, x_2, \ldots, x_n)$$

Hammersley-Clifford Theorem

For example $x_i G_i(x_i) = Q(0, ..., 0, x_i, 0, ..., 0) - Q(0)$

Consider for any vectors x and $x^i = (x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n)$

$$\exp\left(Q(x) - Q(x^i)\right) = \frac{p(x)}{p(x^i)} = \frac{p(x_i \mid x_{\mathrm{N}[i]})}{p(0 \mid x_{\mathrm{N}[i]})}$$

Notice that

$$Q(x) - Q(x^1) = x_1(G_1(x_1) + \sum_{j \neq 1} x_j G_{1j}(x_1, x_j)) + \sum_{j \neq 1, j < k} G_{1jk}(x_1, x_j, x_k) + ... + x_2...$$

Suppose I is not a neighbor of 1. All terms (and thus G functions) involving I must be null. The G functions are thus not null only if the variables form a clique on the graph.

Markov Blanket in an undirected graph

Definition

The Markov Blanket $MB(i)$ of a node i is the smallest set of nodes $MB(i)$ such that $X_i \perp\!\!\!\perp X_R | X_{MB(i)}, with R = V \setminus (MB(i) \cup i)$ or equivalently such that $p(X_i|X_{\setminus i}) = p(X_i|X_{MB(i)})$.
For a Markov Random field the Markov blanket of X_i are its neighbors on G:

$$X_{MB(i)} = X_{N[i]}$$

Moralization

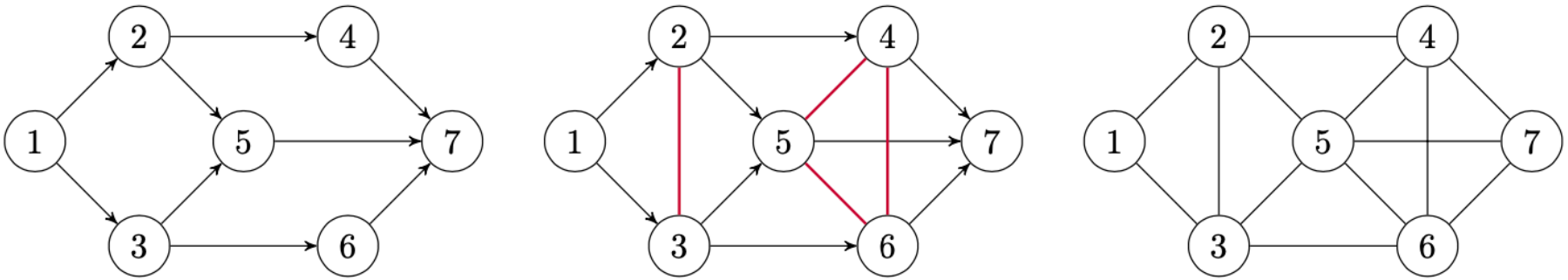
For a given oriented graphical model

- is there an unoriented graphical model which is equivalent?
- is there a smallest unoriented graphical which contains the oriented graphical model?

$p(x) = \frac{1}{Z} \prod_c \psi(x_c)$ vs $p(x) = \prod_i p(x_i \mid x_{\pi(i)})$

Moralization

Given a directed graph G, its moralized graph G_M is obtained by 1. For any node i, add undirected edges between all its parents 2. Remove the orientation of all the oriented edges



Moralization

Proposition

If a probability distribution factorizes according to a directed graph G then it factorizes according to the undirected graph G_M.
A distribution that factorizes according to a directed model is a Gibbs distribution for the cliques $C_i = \{i\} \cup \pi(i)$. As a consequence, it factorizes according to an undirected graph in which C_i are cliques.

Ising model

- The Ising model is an example of an MRF that arose from statistical physics.
- It was originally used for modeling the behavior of magnets.

Let $x_s \in \{-1, +1\}$ represents the spin of an atom, which can either be spin down or up.

In some magnets, called ferro-magnets, neighboring spins tend to line up in the same direction, whereas in other kinds of magnets, called anti-ferromagnets, the spins "want" to be different from their neighbors.

Ising model Gibbs distribution

Consider a graph with pairwise clique potential:

$$\psi_{st}(x_s, x_t) = e^{w_{st} x_s x_t}$$

where w_{st} is the coupling strength between neighboring nodes s and t.

The log probability is then

$$p(\boldsymbol{x}) = \frac{1}{Z} e^{\sum_{s \sim t} w_{st} x_s x_t} = \frac{1}{Z} e^{\frac{1}{2} \boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x}}$$

If $w_{st} = \beta > 0$, we get high probability if neighboring states agree.

Ising model Gibbs distribution with external field

Sometimes there is an external field, which is an energy term which is added to each spin.

This can be modelled using a local energy term of the form $\boldsymbol{b}^T \boldsymbol{x}$, where $\boldsymbol{b} = (b_s)_s$ is sometimes called a bias term.

The modified distribution is given by

$$p(\boldsymbol{x}) = \frac{1}{Z} e^{\frac{1}{2}\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x}}$$

Links with Gaussian

Distribution looks similar to a Gaussian but

Beware normalization constant

- in the case of Gaussians, the normalization constant, $Z = |2\pi\Sigma|$, requires the computation of a matrix determinant, which can be computed in O(D3) time,
- whereas in the case of the Ising model, the normalization constant requires summing over all 2^D bit vectors;

Exercice: Ising Model with $w_{st} = \beta$ and $b_s = \alpha$

Compute the conditional distribution

$$p(x_i = 1 | x_{\setminus i})$$

and use it to design a Gibbs sampler.

Gibbs sampler for ising model

The conditional site probability

can be used to build a Gibbs sampler:

$$p(x_i = 1|x_{N[i]}) = \frac{exp(\alpha + \beta \sum_{j\sim i} x_j)}{exp(\alpha + \beta \sum_{j\sim i} x_j) + exp(-\alpha - \beta \sum_{j\sim i} x_j)}$$

Gibbs sampler

1. Init the random field $x = \{x_i\}$
2. loop through sites

a. Pick a site i at random

b. Simulate from $p(x_i = 1|x_{N[i]})$

R code for fetching neighbors of site x_{ij}

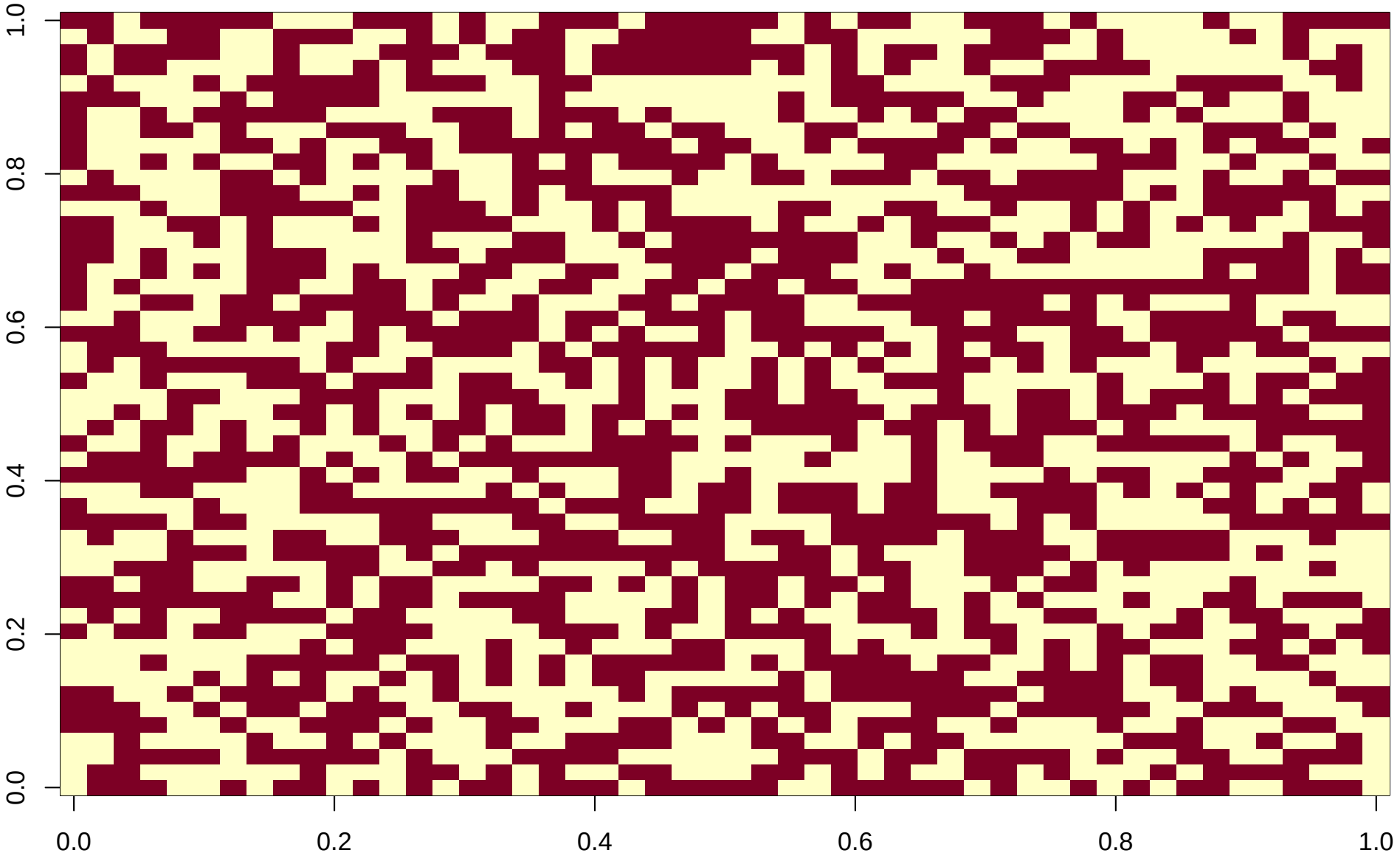
```
get_neighbours<-function(ij,n,p){
  # Get the 4 neighbours of a pixel i,j in a field of size nxp
  #                aj
  #                |
  #         il - ij - ir
  #                |
  #                uj
  j<-(ij-1) %/% n+1 ; i<-ij-(j-1)*n
  u<-ifelse(i+1>n,1,i+1)# V3: u j
  a<-ifelse(i-1<1,n,i-1)# V1: a j
  l<-ifelse(j-1<1,p,j-1)# V3: i l
  r<-ifelse(j+1>p,1,j+1)# V4: i r
  neighbours.coord<-matrix(c(u,a,i,i,j,j,l,r),4,2)
  neighbours.index<-neighbours.coord[,1]+
                    (neighbours.coord[,2]-1)*n
  return(neighbours.index)
}
```

R Gibbs sample for ising model

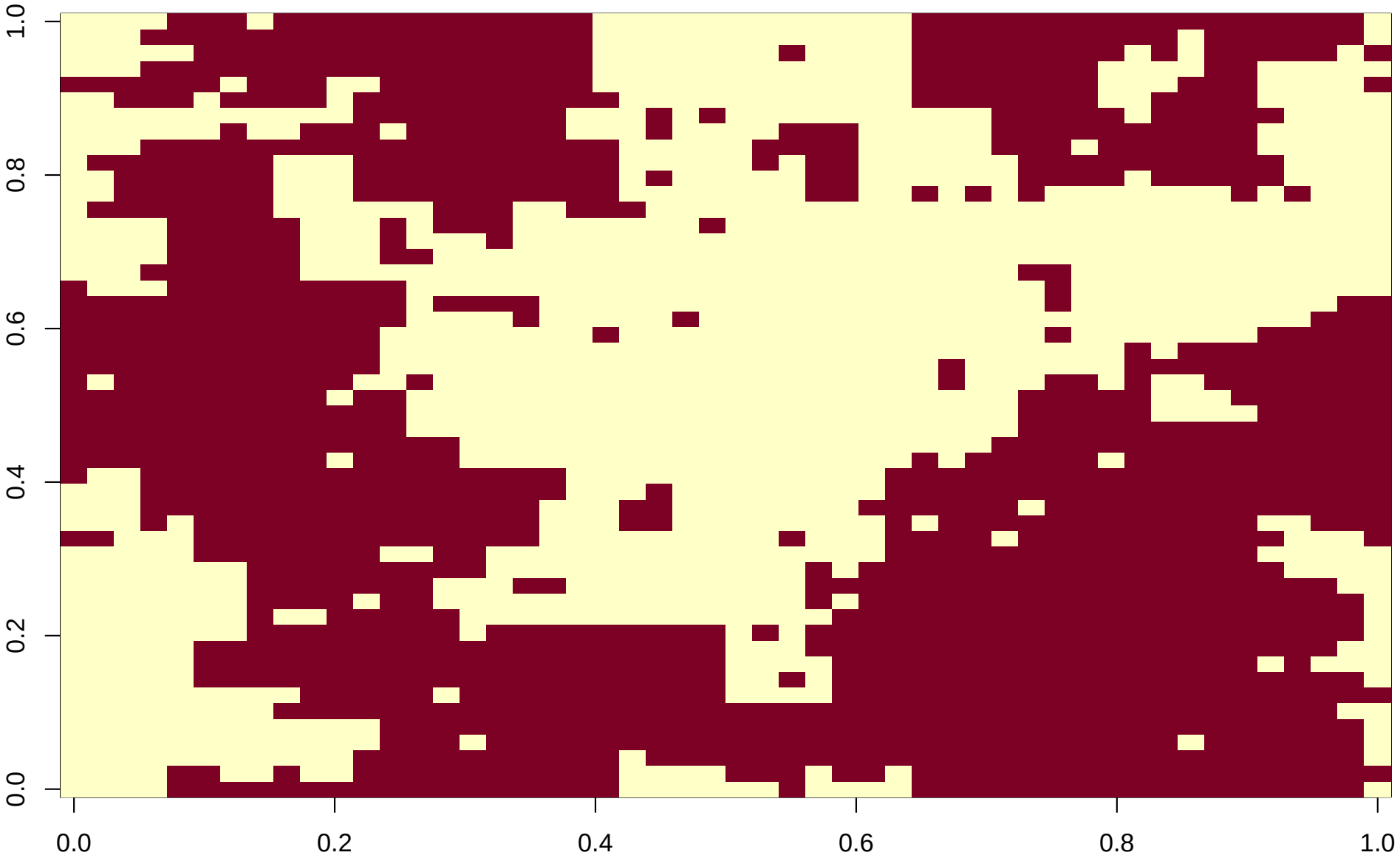
```
gibbs.ising<-function(n=50,p=50,prob=0.5,alpha=0,
                      beta=1/2,nb.cycle=20){
  # Inititialisation
  MRF<-2*matrix(rbinom(n*p,size=1,prob=prob),n,p)-1
  np<-n*p
  cycle<-1
  while(cycle<=nb.cycle){
  cycle<-cycle+1
  walk.order<-sample(1:np,np,replace=FALSE)
  sapply(1:np,function(ij){
     sum.Nij<-sum(MRF[get_neighbours(ij,n,p)])
     pXij.cond.Nij<- exp(alpha+beta*sum.Nij) / (exp(alpha+beta
     MRF[ij]<<- 2*rbinom(1,1,prob=pXij.cond.Nij)-1}
     )
  }
  return(MRF)
}
```

Ising illustration $\beta = 0$

```
image(gibbs.ising(beta=0))
```

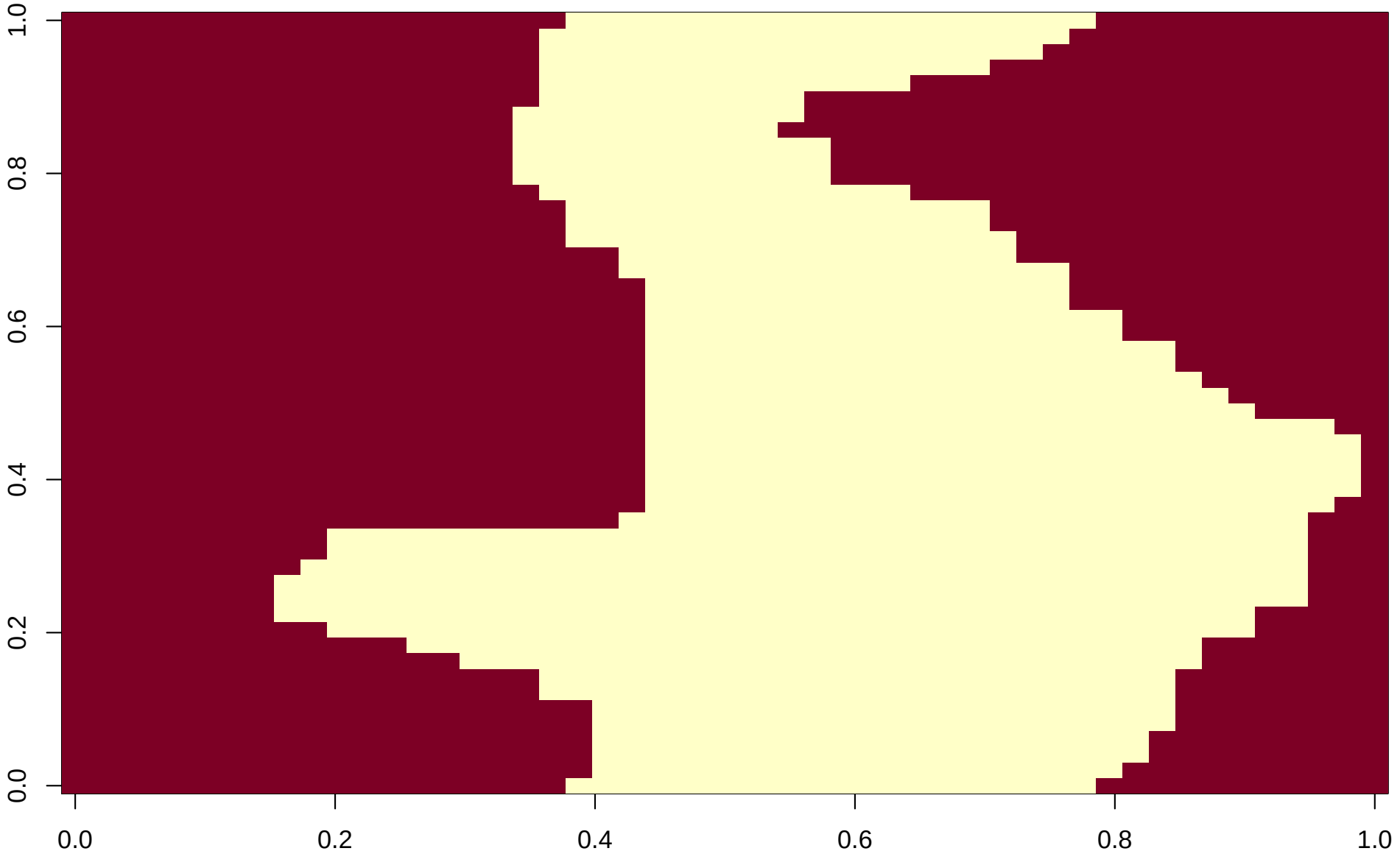


Ising illustration $\beta = 0.5$



Ising illustration $\beta = 3$

```
image(gibbs.ising(beta=3))
```



Hopfield networks

A Hopfield network (Hopfield 1982) is a fully connected Ising model with a symmetric weight matrix, $\boldsymbol{W} = \boldsymbol{W}^T$

Boltzmann machine

A fully connected graph with Bernoulli random variable X_i at each node i whith parameter θ_i given by a logistic regression on the other variables:

$$\text{logit}\,(\theta_i) = \alpha_i + \sum_{i \neq j} \beta_{ij} X_j$$

The weight are symetric and $\beta_{ii} = 0$.

The joint distribution is

$$p(x) = \frac{1}{Z} \exp \sum_i \alpha_i x_i + \sum_{i<j} \beta_{ij} x_i x_j$$

- Boltzmann machine are used to "learn" distributions for prediction or summary.
- Estimation of the parameters is not trivial because of the partition function.

Boltzmann machine estimation of the parameters when the graph is known

- Assuming observation $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{ip}) \in \{0,1\}^p$, with $i = 1, ..., N$. The log-likelihood is
- to avoid handling the bias terms α_i we assume a vetex 0 and condition w.r.t. $x_0 = 1$

$$L(\beta) = \sum_i \log P_\beta(X_i = \boldsymbol{x}_i), \quad (1)$$

$$= \sum_i \sum_{(j,k)\in E} \beta_{jk} x_{ij} x_{ik} - \log Z(\beta) \quad (2)$$

The gradient of the log-likelihood

$$\frac{\partial L(\beta)}{\partial \beta_{jk}} = \sum_i x_{ij} x_{ik} - N \frac{\partial \log Z(\beta)}{\partial \beta_{jk}}$$

where $\frac{\partial \log Z(\beta)}{\partial \beta_{jk}} = \frac{1}{Z(\beta)} \frac{\partial Z(\beta)}{\partial \beta_{jk}} = \sum_{x\in\mathcal{X}} x_j x_k P_\beta(x) = E_\beta[X_J X_K]$

Boltzmann machine estimation of the parameters when the graph is known

Setting the gradient to zero gives

$$\hat{E}(X_j X_k) - E_\beta(X_j X_k) = 0$$

where $\hat{E}(X_j X_k) = \frac{1}{N} \sum_i x_{ij} x_{ik}$

To find the maximum likelihood estimates,

- we can use gradient search or Newton methods.
- However the computation of the expectation is usually not possible

Boltzmann machine estimation of the parameters when the graph is known

- The mean field approximation estimates $E_\beta(X_j X_k)$ by $E_\beta(X_j)E_\beta(X_k)$, and replaces the input variables by their means, leading to a set of nonlinear equations for the parameters β_{jk}.
- To obtain near-exact solutions, Gibbs sampling (Section is used to approximate $E_\beta(X_j X_k)$ by successively sampling from the estimated model probabilities $P\beta(X_j|X_{-j})$.
- ...

Boltzmann machine with Hidden Nodes

- We assume 2 types of variables ($\mathcal{V}$ visible and $\mathcal{H}$ hidden)

The Log-likelihood for K samples is

$$L = \sum_i \log P(X_{\mathcal{V}} = x_{\mathcal{V}_i}) = \sum_i \log \sum_{x_{\mathcal{H}}} P(X_{\mathcal{V}} = x_{\mathcal{V}_i}, X_{\mathcal{H}} = x_{\mathcal{H}})$$

$$L = \sum_i \left(\log \sum_{x_{\mathcal{H}}} \exp \sum_{(j,k) \in E} \beta_{jk} x_{ij} x_{ik} - \log Z \right)$$

where the sum over $x_{\mathcal{H}}$ means that we are summing over all possible $\{0, 1\}$ values for the hidden units.

Boltzmann machine with Hidden Nodes

The gradient is

$$\frac{\partial L(\beta)}{\partial \beta_{jk}} = \sum_i \frac{\sum_{x_{\mathcal{H}}} x_{ij} x_{ik} \exp \sum_{(j,k)\in E} \beta_{jk} x_{ij} x_{ik}}{\sum_{x_{\mathcal{H}}} \exp \sum_{(j,k)\in E} \beta_{jk} x_{ij} x_{ik}} - N \frac{\partial \log Z(\beta)}{\partial \beta_{jk}}$$

Noticing that the numerator is

$$\sum_{x_{\mathcal{H}}} x_{ij} x_{ik} \frac{\exp \sum_{(j,k)\in E} \beta_{jk} x_{ij} x_{ik}}{Z} = E_\beta[X_j X_k \mathbb{I}_{(X_{\mathcal{V}} = x_{\mathcal{V}_i})}] = P(X_j = 1, X_k = 1, X_{\mathcal{V}} =$$

and the denominator is

$$\frac{\sum_{x_{\mathcal{H}}} \exp \sum_{(j,k)\in E} \beta_{jk} x_{ij} x_{ik}}{Z} = P(X_{\mathcal{V}} = x_{\mathcal{V}_i})$$

Boltzmann machine with Hidden Nodes

$$\frac{\partial L}{\partial \beta_{jk}} = \sum_i \Big(\sum_i P(X_j = X_k = 1 \mid X_{\mathcal{V}} = x_{\mathcal{V}_i}) - P(X_j = X_k = 1) \Big) \quad (3)$$

$$= \sum_i E_\beta(X_j X_k | X_{\mathcal{V}_i}) - E_\beta(X_j X_k) \quad (4)$$

Boltzmann machine parameter estimation with hidden nodes

Gibbs sampling

Each part of the sum can be estimated via simulation (e.g. Gibbs sampler):

- unconditionned network
- network with fixed $x_{\mathcal{V}_k}$ (clamped nodes)

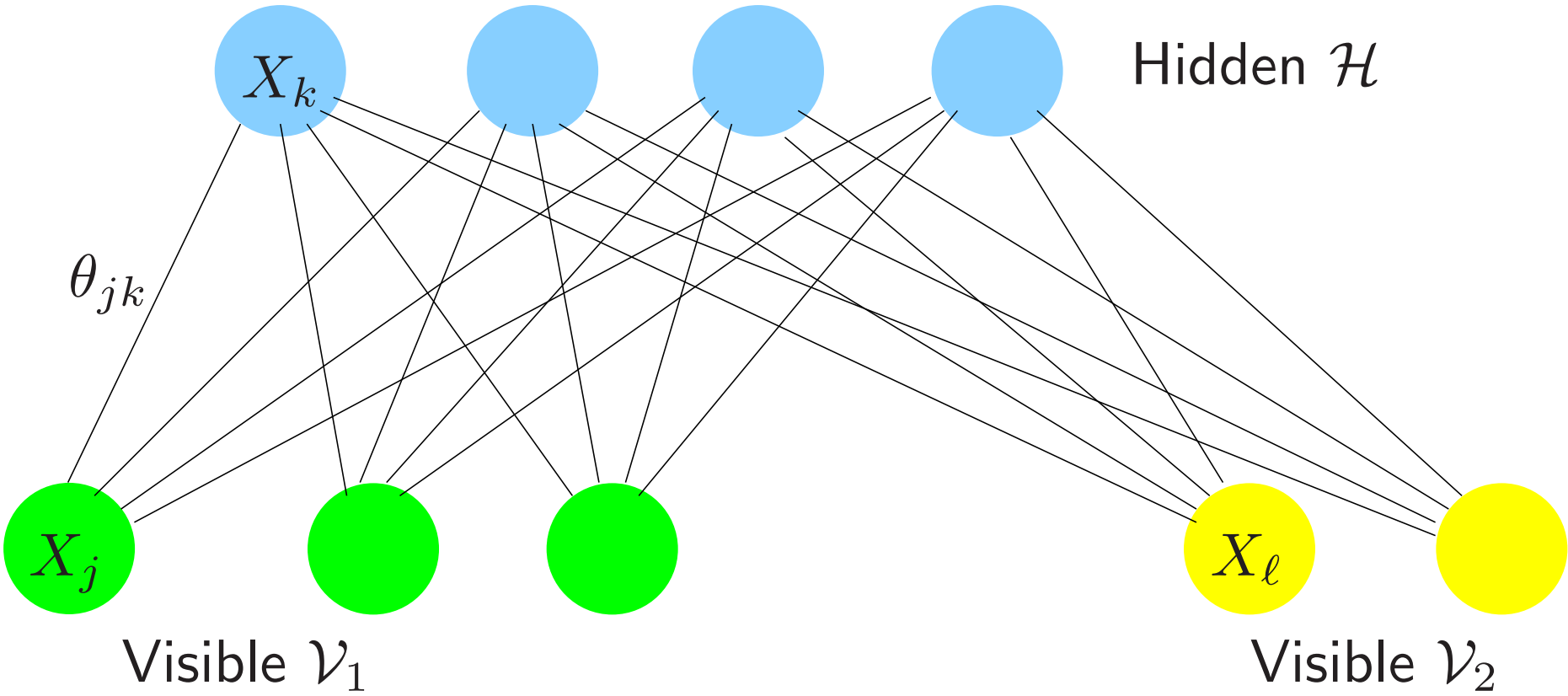
The gradient is used for small steps before re-estimation

Variational approach (mean field approximation)

- Noting $\theta_k = P(X_k = 1)$ the joint distribution $P(X_j = X_k = 1)$ is approximated by $\theta_j \theta_k$
- Problem reduces in estimating θ_k
- Replacing input variables by their means
- And solving the system of non linear equations
  - $\text{logit}(\theta_k) = \alpha_k + \sum_{j \neq k} \beta_{jk} \theta_j \theta_k$ in β_{jk}

Restricted Boltzmann machine

- one layer of visible units and one layer of hidden units with no connections within each layer.
- same generic form as a single hidden layer neural network



Restricted Boltzmann machine

Parallel Gibbs sampling

- the variables in each layer are independent of one another, given the variables in the other layers.
- Hence they can be sampled together, using the conditional probabilities $p(x_i = 1|x_{N[i]})$

Training RBM using Gibbs sampling

The gradient of the likelihood for one obs. is

$$E_\beta(X_j X_k | X_{\mathcal{V}_i}) - E_\beta(X_j X_k)$$

Let denote $v = X_{\mathcal{V}_i}$ and h the observed value of the hidden nodes conditionnaly to v.

- The first exectation could be roughly approximated by the product $v_j h_k$
- The second expectation could be approximated with one round of Gibbs sampling by the product $v'_j h'_k$

Training RBM using Gibbs Sampling

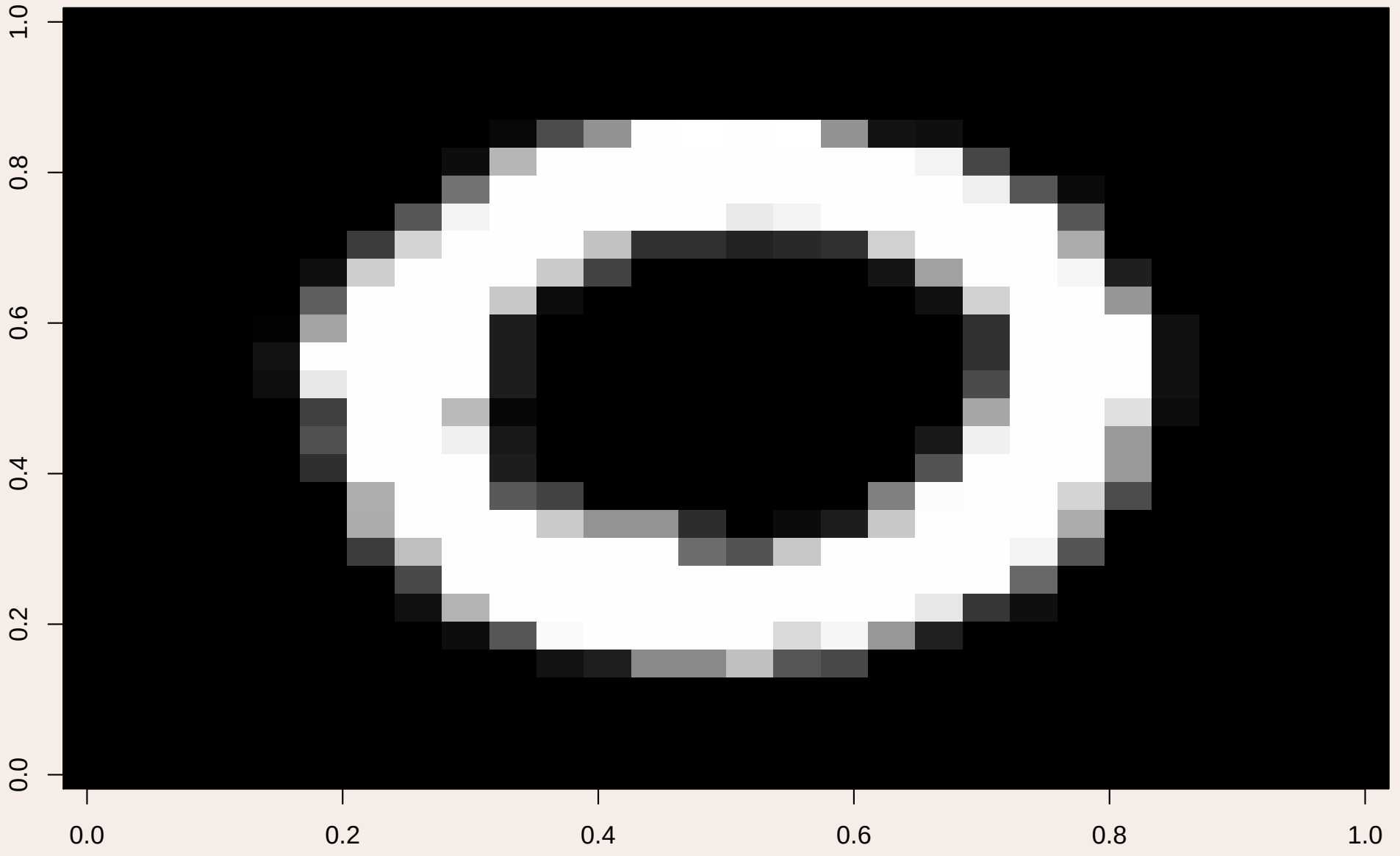
1. Take a training sample v, compute the probabilities of the hidden units and sample a hidden activation vector h from this probability distribution.
2. Compute the outer product of v and h and call this the positive gradient.
3. From h, sample a reconstruction v' of the visible units, then resample the hidden activations h' from this. (Gibbs sampling step)
4. Compute the outer product of v' and h' and call this the negative gradient.
5. Update to the weight matrix and the bias

$$\Delta\beta_{jk} = \epsilon(v_j h_k - v'_j h'_k)$$

$$\Delta\alpha_j = \epsilon(v_j - v'_j),\ \Delta\alpha_h = \epsilon(h_k - h'_k),$$

Github example from TimoMatzen

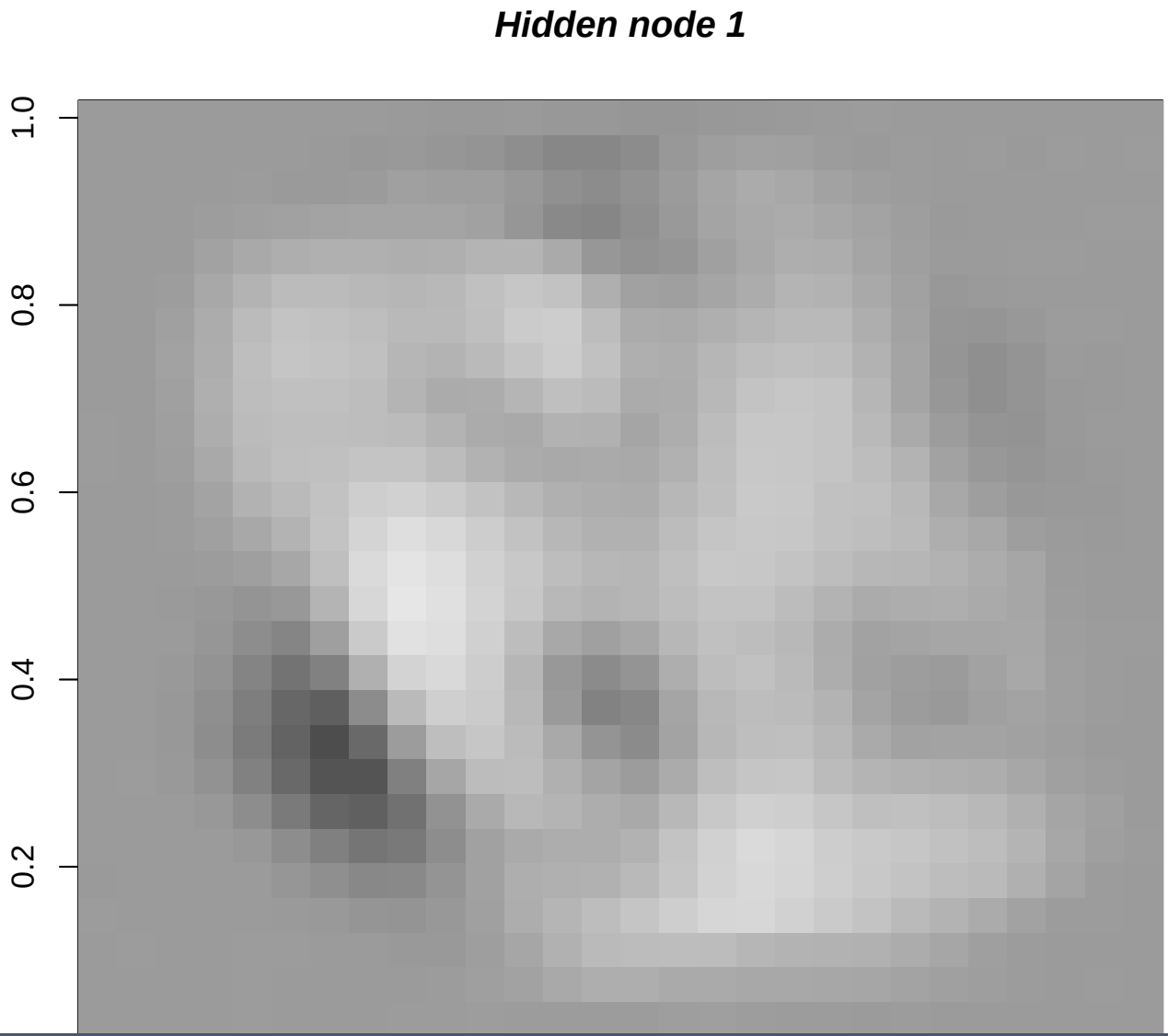
The MNIST Data



RBM: Github example from TimoMatzen: Learning

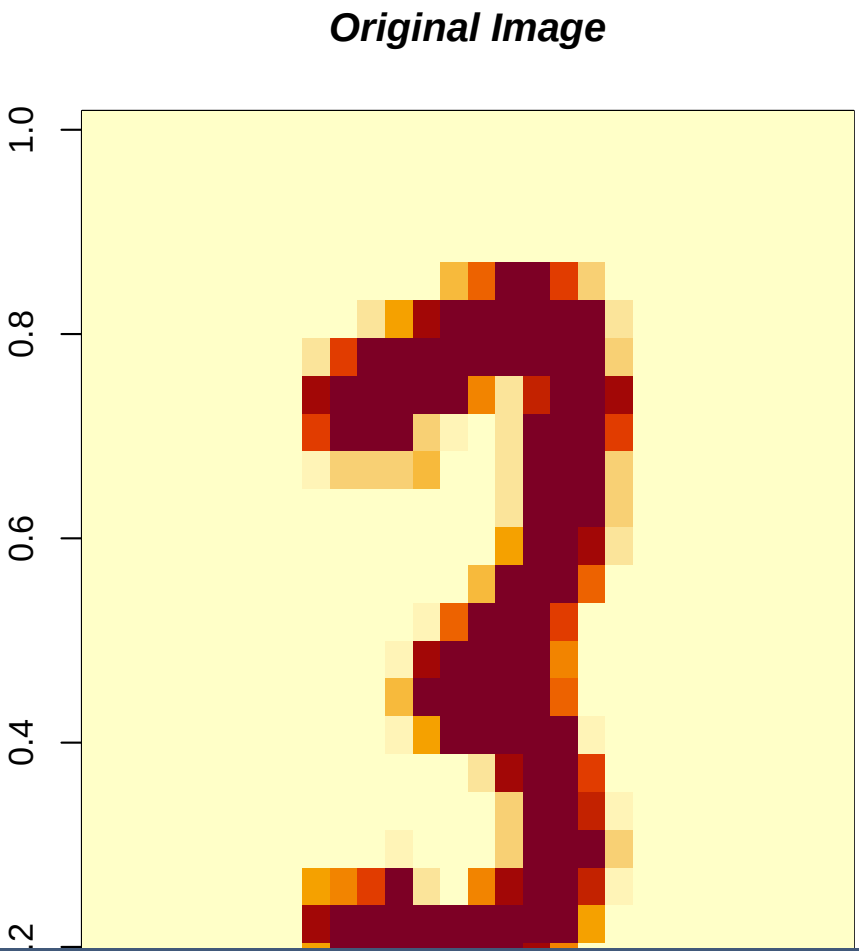
```
train <- MNIST$trainX
nb.hidden.units<-10
modelRBM <- RBM(x = train, n.iter = 1000,
                n.hidden = nb.hidden.units,
                size.minibatch = 10)
```

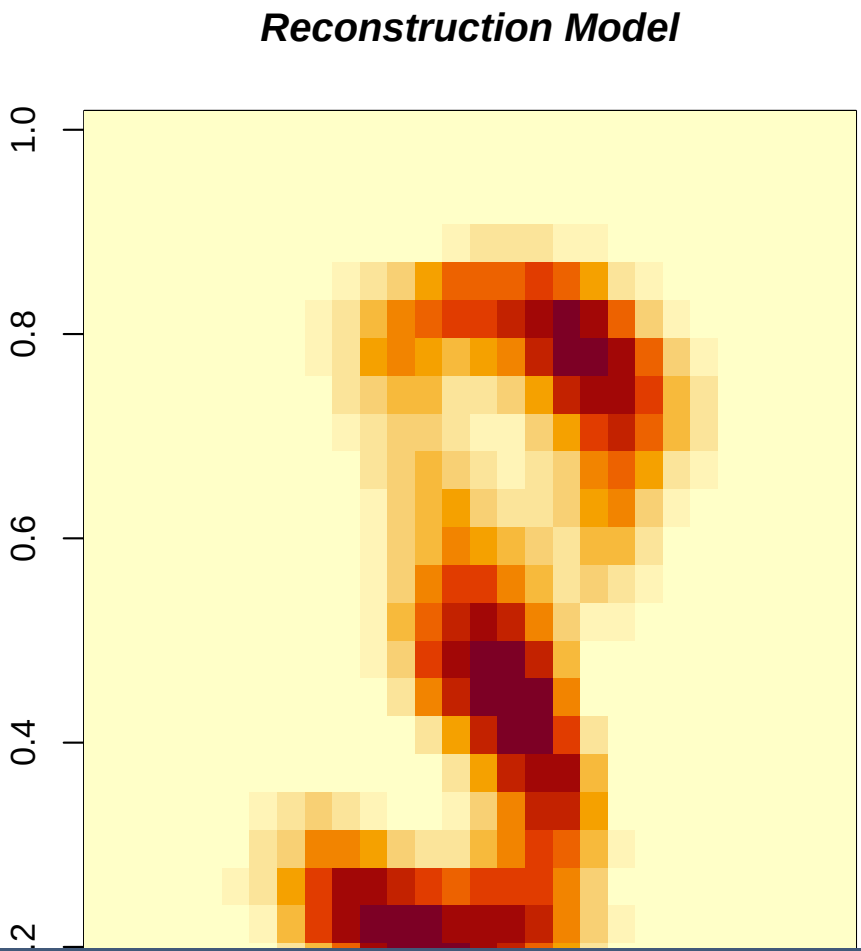
RBM: Reconstruction example from TimoMatzen Hidden units



Reconstruction example from TimoMatzen

```
# Get the test data from MNIST
test <- MNIST$testX
# Reconstruct the image with modelRBM
ReconstructRBM(test = test[6, ], model = modelRBM)
```

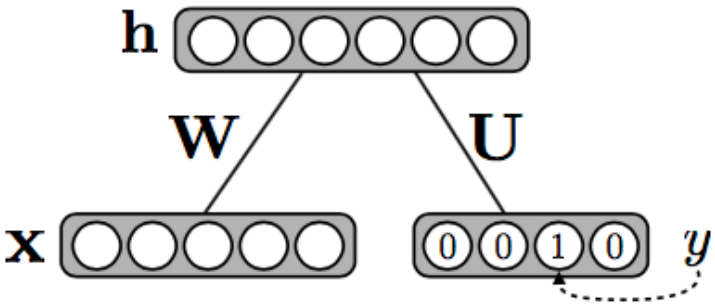




Classification with Boltzmann Machine

Using 2 type of visible units:

- the pixel values
- the (binarized) labels.



Classification example from TimoMatzen

Training

```
data(MNIST)
# First get the train labels of MNIST
TrainY <- MNIST$trainY
# This time we add the labels as the y argument
modelClassRBM <- RBM(x = train, y = TrainY,
                     n.iter = 3000, n.hidden = 200,
                     size.minibatch = 10)
```

Classification example from TimoMatzen

Testing

```
# First get the test labels of MNIST
TestY <- MNIST$testY
# Give our ClassRBM model as input
PredictRBM(test = test, labels = TestY, model = modelClassRBM)
#[1] 0.852
```

Reference about RBM

- Hinton, G. A Practical Guide to Training Restricted Boltzmann Machines (2010) : https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf

Exponential family

Discrete Case

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^t \phi(\boldsymbol{x}) - A(\boldsymbol{\theta})\right)$$

where $A(\boldsymbol{\theta}) = \log(Z(\boldsymbol{\theta})) = \log\left(\sum_{\boldsymbol{x}} \exp\left(\boldsymbol{\theta}^t \phi(\boldsymbol{x})\right)\right)$

Examples

Gaussian, Bernoulli, Binomial, Poisson, Exponential, Weibull, Laplace, gamma, beta, multinomial, Wishart distributions

Derivatives of the log partition function

$$\frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{x}} \phi(\boldsymbol{x}) \exp\left(\boldsymbol{\theta}^t \phi(\boldsymbol{x})\right) = E[\phi(\boldsymbol{x})]$$

$$\begin{aligned}
\frac{\partial^2 A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^t} &= \frac{\partial}{\partial \boldsymbol{\theta}^t} \sum_{\boldsymbol{x}} \phi(\boldsymbol{x}) \exp\left(\boldsymbol{\theta}^t \phi(\boldsymbol{x}) - A(\boldsymbol{\theta})\right) \\
&= \sum_{\boldsymbol{x}} p(\boldsymbol{x}; \boldsymbol{\theta}) \phi(\boldsymbol{x}) (\phi(\boldsymbol{x})^t - E[\phi(\boldsymbol{x})]^t) \\
&= E[\phi(\boldsymbol{x}) \phi(\boldsymbol{x})^t] - E[\phi(\boldsymbol{x})]^t E[\phi(\boldsymbol{x})] \\
&= var[\phi(\boldsymbol{x})]
\end{aligned}$$

Gradient ascent for maximizing the log-likelihood

The Log-likelihood for K samples is

$$L = \sum_k \log p(\boldsymbol{x}_k; \boldsymbol{\theta})$$

Gradient ascent

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\theta}} &= \sum_k \frac{\partial}{\partial \boldsymbol{\theta}}((\boldsymbol{\theta}^t \phi(\boldsymbol{x}_k) - A(\boldsymbol{\theta})) \\
&= \sum_k (\phi(\boldsymbol{x}_k) - E[\phi(\boldsymbol{x})]) \\
&= K\left(\hat{E}[\phi(\boldsymbol{x})] - E[\phi(\boldsymbol{x})]\right)
\end{aligned}$$

$$\boldsymbol{\theta}^{q+1} = \boldsymbol{\theta}^{q+1} + \epsilon \frac{\partial L}{\partial \boldsymbol{\theta}}$$

Undirected models are members of exponential family.

If we consider $\phi(\boldsymbol{x}) \propto \prod_{c \in \mathcal{C}} \Psi(\boldsymbol{x}_c)$

it can be rewritten as exponential family

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^t \phi(\boldsymbol{x}) - A(\boldsymbol{\theta})\right)$$

where

$$\phi(\boldsymbol{x}) = \{\mathbb{1}_{\{\boldsymbol{x}_c = \boldsymbol{x}_c^*\}} \mid \forall c \in \mathcal{C}, \forall \textit{ possible } x_c^*\}$$

In that case the derivative of the partition function are the marginals:

$$E[\mathbb{1}_{\{\boldsymbol{x}_c = \boldsymbol{x}_c^*\}}] = p(\boldsymbol{x}_c = \boldsymbol{x}_c^*; \boldsymbol{\theta})$$

Section 2

Gaussian Graphical Model

Gaussian Graphical Model

Density

A random vector $\boldsymbol{x} \in \mathbb{R}^p$ is distributed according to the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ the covariance matrix is defined by

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$$

Precision matrix

The inverse covariance matrix, also known as the precision matrix or the concentration matrix, $K = \Sigma^{-1}$

Link to Exponential family

Canonical parameters and sufficient statistics

$$\boldsymbol{\theta} = \{-\boldsymbol{\mu}^t K, K\}$$

$$\phi(\boldsymbol{x}) = \{\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^t\}$$

Link to Markov Random Field

Factorization

$$f(\boldsymbol{x}) \propto \prod_j \Psi(x_j) \prod_{j<k} \Psi(x_{jk})$$

where $\Psi(x_j) = (exp(-\sum_k \mu_k K_{kj})x_j$ and $\Psi(x_{jk}) = exp(x_j K_{jk} x_k)$

Markov property

From the factorization it is straightforward that

$$K_{ij} = 0 \Leftrightarrow X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}}$$

Graph $G = \{V, E\}$ where $K_{ij} = 0 \Leftrightarrow \forall (i,j) \notin E$ describes the sparsity pattern of the concentration matrix.

Gaussian distribution and Conditional independence

$$\begin{bmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \end{bmatrix} \sim \mathcal{N}_p \left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}; \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \right)$$

We have the following property

$$(\boldsymbol{x}_A \mid \boldsymbol{x}_B = b) \sim \mathcal{N}_p \left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(b - \mu_B); \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA} \right).$$

The idea of the proof consists in computing the conditional density $f(\boldsymbol{x}_A|\boldsymbol{x}_B = b) = f_{A,B}(\boldsymbol{x}_A, b)/f_B(b)$ knowning that both f_B and $f_{A,B}$ are multivariate gaussian.

Concentration matrix

The concentration matrix is $K := \Sigma^{-1}$. Using the partition of the multivariate vector in A and B, the Schur complement allows to compute

$$K_{AA} = (\Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA})^{-1}$$

which is exactly the inverse of the conditional covariance of $A|B$.

If $A = (x_1, x_2)^t$ and $B = (x_3, ..., x_p)^t$ then

$$K_{AA} = \begin{pmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{pmatrix} = \Sigma_{A|B}^{-1}$$

Concentration matrix

Thus the conditional covariance of $A|B$ expressed in terms of concentration becomes

$$\Sigma_{A|B} = \frac{1}{det(K_{AA})} \begin{pmatrix} k_{22} & -k_{12} \\ -k_{12} & k_{11} \end{pmatrix}$$

and the correlation of $x_1 x_2 | x_3 \ldots x_p$ is

$$\frac{-k_{12}}{\sqrt{k_{11} k_{22}}}.$$

Covariance selection

The task of computing the MLE for a (non-decomposable) GGM is called covariance selection (Dempster 1972).

$$\log L(K) = \log detK - tr(SK)$$

where $S = \frac{1}{N}\sum_{i=1}^{N}(xi - x)(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$ is the empirical covariance matrix.

Exercice

Derive the equation of the log-likelihood

The gradient

$$\nabla \log L(K) = K^{-1} - S$$

Estimation of K when the graph structure is known

Interestingly, one can show that the MLE must satisfy the following property:

- $\Sigma_{st} = S_{st}$ if $G_{st} = 1$ or $s = t$
- $K_{st} = 0$ if $G_{st} = 0$, by definition of a GGM, i.e., the precision of a pair that are not connected must be 0.

Σ is a positive definite matrix completion of S

it retains as many of the entries in S as possible:

- corresponding to the edges in the graph
- subject to the required sparsity pattern on K, corresponding to the absent edges;
- the remaining entries in Σ are filled in so as to maximize the likelihood.

Estimation of K when the graph structure is known (Example)

Let us consider the example from (Hastie et al. 2009, p652) representing the cyclic structure, $X_1 - -X_2 - -X_3 - -X_4 - -X_1$, and the following empirical covariance matrix:

$$\mathbf{G} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

The MLE is given by

$$\mathbf{\Sigma} = \begin{pmatrix} 10.00 & 1.00 & \mathbf{1.31} & 4.00 \\ 1.00 & 10.00 & 2.00 & \mathbf{0.87} \\ \mathbf{1.31} & 2.00 & 10.00 & 3.00 \\ 4.00 & \mathbf{0.87} & 3.00 & 10.00 \end{pmatrix}, \quad \mathbf{\Omega} = \begin{pmatrix} 0.12 & -0.01 & \mathbf{0} & -0.05 \\ -0.01 & 0.11 & -0.02 & \mathbf{0} \\ \mathbf{0} & -0.02 & 0.11 & -0.03 \\ -0.05 & \mathbf{0} & -0.03 & 0.13 \end{pmatrix}$$

Estimation of K when the graph is structure is unknown

By analogy to lasso one can define the following ℓ_1 penalized criterion:

$$J(K) = -\log detK + tr(SK) + \lambda \|K\|_1$$

where $\|K\|_1 = \sum_{st} |k_{st}|$

Several algorithms have been proposed for optimizing this objective (Yuan and Lin 2007; Banerjee et al. 2008; Duchi et al. 2008), although arguably the simplest is the one in (Friedman et al. 2008), which uses a coordinate descent algorithm similar to the shooting algorithm for lasso.

Graphical Lasso

The subgradient equation is

$$K^{-1} - S - \lambda Sign(K) = \mathbf{0},$$

where $Sign(K_{jk}) = sign(K_{jk})$ if $K_{jk} \neq 0$, else $Sign(K_{jk}) \in [-1, 1]$ if $K_{jk} = 0$.

The graphical Lasso use regression to solve for K and its inverse $W = K^{-1}$ one row and column at a time.

Graphical Lasso

If we consider a partition of columns in two

1. $p-1$ first colums
2. last column p

we have by definition

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^t & w_{22} \end{pmatrix} \begin{pmatrix} K_{11} & K_{12} \\ K_{12}^t & k_{22} \end{pmatrix} = I$$

show that w_{12} can be regressed from W_{11}

$$w_{12} = -W_{11}\frac{K_{12}}{k_{22}} = W_{11}\beta$$

Graphical Lasso (from Hastie & Tibshirani)

1. Initialize $W = S + \lambda I$. The diagonal of W remains unchanged in what follows.
2. Loop through columns until convergence
   a. Partition the matrix W into part 1: all but the jth row and column, and part 2: the jth row and column.
   b. Solve the lasso type problem $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$ using the cyclical coordinate-descent algorithm.
   c. Update $w_{12} = W_{11}\beta$
3. In the final cycle (for each j) solve for $K_{12} = -\beta K_{22}$, with $1/K_{22} = w_{22} - w_{12}^T\beta$

Shooting

Section 3

Appendix

Gibbs Sampling

Multi-stage Gibbs sampler: One step of the algorithm has p stages

1. Given $(X_1^n, \cdots, X_p^n)$ we sample X_1^{n+1} from $P(.|X_1^n, \cdots, X_p^n)$
2. Then sample X_2^{n+1} from $P(.|X_1^{n+1}, X_3^n, \cdots, X_p^n)$

j. Continuing we sample X_j^{n+1} from $P(.|X_1^{n+1}, \cdots, X_{j-1}^{n+1}, X_{j+1}^n, \cdots, X_p^n)$

p. In the last step we sample X_p^{n+1} from $P(.|X_1^{n+1}, \cdots, X_{p-1}^{n+1})$

Transition Matrix

Let A_j be the transition matrix corresponding to the j^{th} step of the multi-stage Gibbs sampler

$$A_j(x_1, x_2, \cdots, x_p; x_1', x_2', \cdots, x_p') = P(x_j'|x_1, x_2, \cdots, x_{j-1}, x_{j+1}, \cdots, x_p) \prod_{(i \neq j)} \delta(x$$

The $\prod_{(i \neq j)} \delta(x_i - x_i')$ ensure that only site j can be different between the origin state $x_1, x_2, \cdots, x_p$ and the arrival state $x_1', x_2', \cdots, x_p'$.

For a **randomized Gibbs sampler**, fix some probability distribution q_i on $\{1, 2, \cdots, p\}$. Given that we are in state $(X_1^n, \cdots, X_p^n)$, we first pick $i \in \{1, 2, \cdots, p\}$ according to this distribution. Then we sample X_i^{n+1} from $P(\mathring{u}|X_1^n, \cdots, X_{i-1}^n, X_{i+1}^n, \cdots, Xp^n)$. The transition matrix for this algorithm is

$$A = \sum_j q_j A_j$$

Stationary distribution

Proposition: $P(x_1, x_2, \cdots, x_p)$ is the stationary distribution of the multi-stage Gibbs sampler and of the randomized Gibbs sample for any choice of the distribution q_i.

We only need to show that for all j, $A_j^T(x_1', x_2', \cdots, x_p', \bullet)P = P$

$$
\begin{aligned}
A_j^T(x', \bullet)P &= \sum_{x_1, x_2, \cdots, x_p} \cdots \sum_{x_1, x_2, \cdots, x_p} P(x_1, x_2, \cdots, x_p) A_j(x_1, x_2, \cdots, x_p; x_1', x_
\\
&= \sum_{x_1, x_2, \cdots, x_p} P(x_1, x_2, \cdots, x_p) P(x_j' | x_1, x_2, \cdots, x_{j-1}, x_{j+1}, \cdots, x_p)
\\
&= P(x_j' | x_1', x_2', \cdots, x_{j-1}', x_{j+1}', \cdots, x_p') \sum_{x_j} P(x_1', x_2', \cdots, x_{j-1}', x_j;
\\
&= P(x_j' | x_1', x_2', \cdots, x_{j-1}', x_{j+1}', \cdots, x_p') P(x_1', x_2', \cdots, x_{j-1}', x_{j+1}', \cdot
\\
&= P(x_1', x_2', \cdots, x_p')
\end{aligned}
$$

Section 4

Exercices

Exercices Unidirected Graphical Model

Conditional independence

Let consider three sets of discrete variables X, Y, Z. Show that if there exist two function F and G such that

$$P(X, Y, Z) = F(X, Z)G(Y, Z)$$

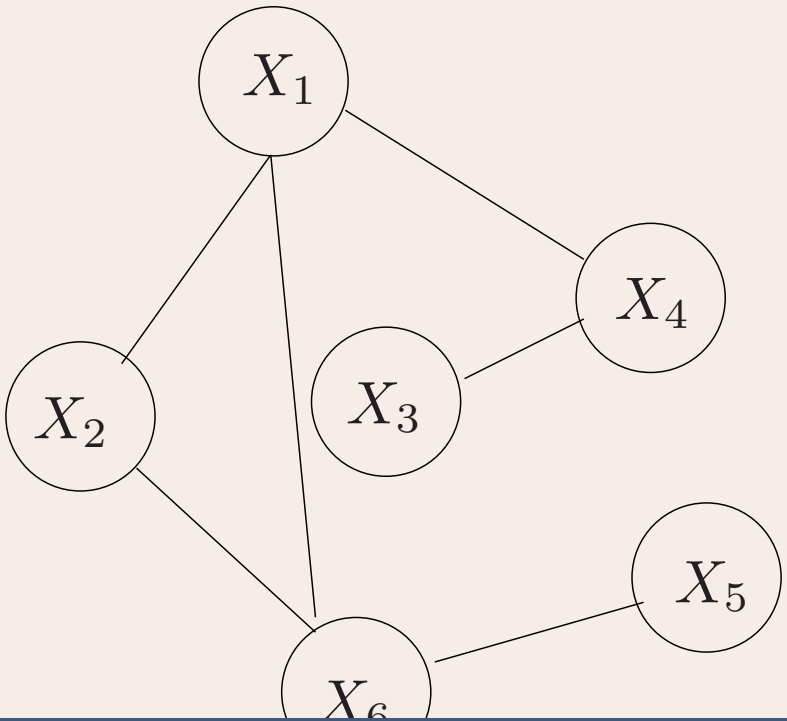
then

$$X \perp\!\!\!\perp Y \mid Z$$

Exercices Unidirected Graphical Model

Conditional independence (Exo 17.1 Elements of Stat)

For the Markov graph follow, list all of the implied conditional independence relations and find the maximal cliques.



Exercices Unidirected Graphical Model

Independences to graph (Exo 17.2 Elements of Stat)

Consider random variables X_1, X_2, X_3, X_4. In each of the following cases draw a graph that has the given independence relations:

a. $X_1 \perp\!\!\!\perp X_3 \mid X_2$, and $X_2 \perp\!\!\!\perp X_4 \mid X_3$.
b. $X_1 \perp\!\!\!\perp X_4 | X_2, X_3$ and $X_2 \perp\!\!\!\perp X_4 \mid X_1, X_3$.
c. $X_1 \perp\!\!\!\perp X_4 \mid X_2, X_3, X_1 \perp\!\!\!\perp X_3 | X_2, X_4$ and $X_3 \perp\!\!\!\perp X_4 \mid X_1, X_2$.