Spatial Statistics with Image Analysis
Lecture 11

Johan Lindström

December 7, 2018

Lecture L11

General MRF:s
Gibbs distributions

Gibbs sampling
Block updating
Parallel updating
Convergence

Parameter estimation
Notation for discrete MRF:s
Pseudo-likelihood
Example

General Markov random fields

We want models, defined for locations \( s_i \), which fulfil the

Markov condition

\[
p(x_i | \{ x_j : j \neq i \}) = p(x_i | \{ x_j : j \in N_i \})
\]

for the neighbours \( N_i \)

Gibbs distributions

Originally from mathematical physics:

\[
p(x) = \frac{1}{Z} \exp \left( -\frac{W(x)}{kT} \right),
\]

where \( W \) is the energy of a state \( x \), \( T \) is a temperature, \( k \) is some physical constant, and \( Z \) is an (unknown) normalising constant.
Gibbs distributions

- Neighbourhood system \( \mathcal{N}_i = \{ s_j; s_i \text{ is a neighbour of } s_i \} \).
- Cliques \( C \).
- Potentials \( V_C(x; \theta) \).
- Parameters \( \theta \).
- Density/probability function

\[
p(x|\theta) = \frac{1}{Z_\theta} \exp \left( \sum_{C} V_C(x; \theta) \right)
\]

- Conditional distributions

\[
p(x_i|x_j, j \in \mathcal{N}_i, \theta) = \frac{1}{Z_{i,\theta}} \exp \left( \sum_{C; s_i \in C} V_C(x; \theta) \right)
\]

Gibbs Normalizing constant

- The normalizing constant \( Z_\theta \) sums over possible combination of \( x \).
- For a \( 3 \times 3 \) image with \( x_{(i,j)} \in \{0, 1\} \) there exists \( 2^9 \) possible combinations.
- For a \( 10 \times 10 \) image there exists \( 2^{100} \approx 10^{30} \) possible combinations. Thus in reality not possible to compute \( Z_\theta \).

Example: Integer valued fields

- Integer pixels, when only first and second order cliques are used:

\[
V_{(i)}(x) = \alpha_k, \text{ when } x_i = k \\
V_{(i,j)}(x) = \beta_{kl}, \text{ when } x_i = k \text{ and } x_j = l.
\]

- Simple case preferring equal neighbours:

\[
\beta_{kl} = \begin{cases} 
\beta_k, & \text{if } k = l \\
0, & \text{otherwise}
\end{cases}
\]

- If each pixel takes a value in the set \( \{1, 2, \ldots, K\} \), the space of possible outcomes for \( N \) pixels \( x \) is

\[
\Lambda = \{1, 2, \ldots, K\}^N
\]
The global probability function for the simple integer field model is:

\[ p(x) \propto \exp \left( \sum_{i=1}^{N} \alpha x_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in N_i} \beta I(x_j = x_i) \right) \]

The conditional distribution for a pixel \( x_i \) given the rest of the field is:

\[ p(x_i|x_j, j \neq i) = \frac{p(x)}{p(\{x_j, j \neq i\})} = \frac{p(x)}{\sum_{k=1}^{K} p(x = k, \{x_j, j \neq i\})} \]

If \( \beta > 0 \) we obtain fields where neighbouring pixels strive to be equal.

Gibbs sampling

- Sampling from the distribution \( p(x) \) is hard.
- However the conditional distributions

\[ p(x_i|x_j, j \neq i) = p(x_i|x_j, j \in N_i) \]

are straightforward to obtain and simulate from.

Gibbs sampling says that we can simulate from a complex distribution \( p(x) \) by repeatedly simulating from the conditionals \( p(x_i|x_j, j \neq i) \).
Block-update Gibbs-sampling

- Sequential Gibbs-sampling is often slow.
- Can we use the Markov property better?
- Compute the conditional distribution for an entire block of neighbouring pixels, allowing us to update larger sets of blocks.
- But, computing the conditional for a large enough block may be difficult.

Parallel Gibbs-sampling

- Can we update more than one pixel at the same time, without having them interfere with each other?
- In the 4-neighbour system, each pixel is conditionally independent of its diagonal "neighbours", given its 4-neighbours.
- Looping over conditionally independent pixels will give exactly the same result as updating them simultaneously.
- Partition the pixels into conditionally independent sets, and update all pixels in each set in parallel. Loop over these sets, forwards and backwards, or choose a set at random.
Convergence diagnostics

- How can we tell whether the simulation has converged to the correct distribution or not? How many iterations are required?
- This is a difficult problem, and no clear answers exist.
- Various heuristics can be applied:
  - Plot the trajectory of the (unnormalised) log-likelihood
  - Plot the trajectory of various functionals of the field, such as the relative amount of the different classes.
  - When the trajectories have oscillated around a constant long enough, the simulation is assumed to have converged.

Parameter estimation for discrete MRF:s

- Maximum likelihood:
  - The normalising factor $Z_\theta$ depends on the parameters.
  - $Z_\theta$ is typically troublesome or impossible to calculate.
  - This makes direct Maximum likelihood estimation hard/impossible.
- Pseudo-likelihood estimation:
  - The normalisation constants $Z_{\theta, j}$ for the conditional distributions are usually simple to compute.
  - Maximise the product of all the full conditional distributions
    \[
    \hat{\theta} = \arg \max_{\theta} \prod_i p(x_i | x_j, j \in N_i; \theta)
    \]
    with respect to the parameters $\theta$.
  - It can be shown that Pseudo-likelihood estimates share many of the properties fond in maximum likelihood estimation.
Useful notation for a discrete MRF

To simplify the notation, we rewrite the MRF using a multi-dimensional indicator field:

\[ z_{ik} = \mathbb{I}(x_i = k) \]

\[ f_{ik} = \sum_{j \in N_i} z_{jk} \]

\[ P(x_i = k) = E(z_{ik}) \]

For the discrete model we have

\[ P(x_i = k | x_j, j \in N_i, \alpha, \beta) = E(z_{ik} | x_{ij}, \alpha, \beta) = \frac{\exp(\alpha_k + \beta f_{ik})}{\sum_{l} \exp(\alpha_l + \beta f_{il})} = E(z_{ik} | f_i, \alpha, \beta) \]

Pseudo-likelihood estimation — Identifiability

For the discrete model the log-pseudo-likelihood is

\[ \log PL_z(\alpha, \beta) = \sum_k \alpha_k \sum_i z_{ik} + \beta \sum_k \sum_i z_{ik} f_{ik} \]

\[ - \sum_i \log \left( \sum_k \exp(\alpha_k + \beta f_{ik}) \right) \]

\( \alpha_k \) might not be identifiable. Take:

\[ a_k = \exp(\alpha_k), \quad \text{and require that} \quad \sum_k a_k = 1 \]

\( PL_z \) is not guaranteed to be bounded for large \( \beta \). Introduce a prior on \( \beta \)

\[ \beta \in \mathbb{N}(0, \sigma^2) \]

Simulated fields and parameter estimation \((\beta; \alpha = 0)\)
Pseudo-likelihood estimation — Example

Given a $4 \times 4$-image, 

\[
\begin{pmatrix}
1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0
\end{pmatrix}
\]

and a 4-neighbourhood we want to estimate $\beta$ (assuming $\alpha = 0$). The pseudo-likelihood for the × conditionally independent pixels,

\[
\begin{array}{cccc}
\times & - & \times & - \\
- & \times & - & \times \\
\times & - & \times & - \\
- & \times & - & \times \\
\end{array}
\]

is

\[
L = e^{7\beta}(1 + e^{2\beta})^{-4}(1 + e^{\beta})^{-4}
\]

with maximum for $\hat{\beta} \approx 0.202$ ($\beta \approx 0.343$ for the other set of conditionally independent pixels).

Pseudo-likelihood estimation

For the estimation the set of points, $I$, over which we compute the approximate likelihood

\[
\log PL_z(\alpha, \beta) = \sum_k \alpha_k \sum_{i \in I} z_{ik} + \beta \sum_k \sum_{i \in I} z_{ik} f_{ik} - \sum_{i \in I} \log \left( \sum_k \exp(\alpha_k + \beta f_{ik}) \right)
\]

can be selected as either a set of conditionally independent points (as in the example above) leading to one estimate for each set; or as all points in the field.

Conditionally independent points:

- Called coding-method.
- Final estimate obtained by averaging each individual estimate.
- Lower bias.

All points:

- Called pseudo-likelihood.
- Nice asymptotic properties (consistent, Normal).
- Lower variance.