Statistics 352: Spatial statistics

Models for discrete data

Jonathan Taylor
Department of Statistics
Stanford University

April 28, 2009

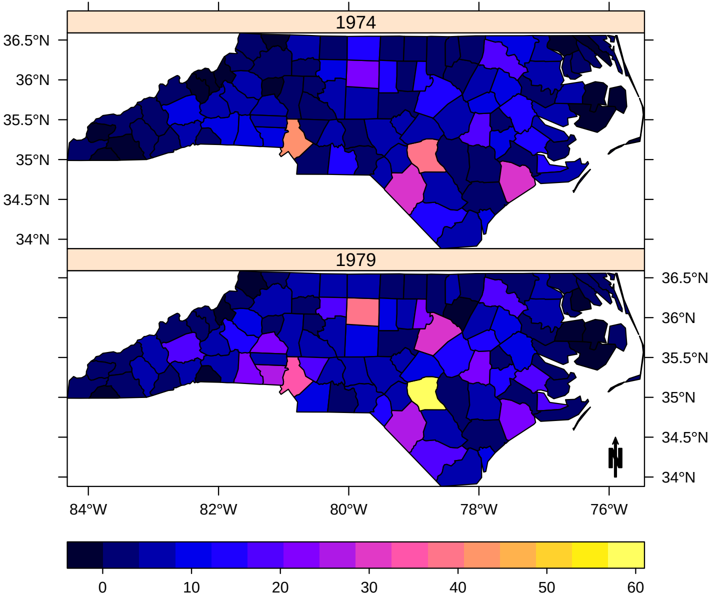
Models for discrete data

Outline

- Dependent discrete data.
- Image data (binary).
- Ising models.
- Simulation: Gibbs sampling.
- Denoising.

Jonathan
Taylor
Department of
Statistics
Stanford
University

SIDS (sudden infant death syndrome) in North Carolina



Discrete data

Description

- Observations:
  - Incidence of SIDS: $\{Z_i : i \text{ is a county in North Carolina}\}$.
  - Births in county: $\{n_i : i \text{ is a county in North Carolina}\}$.
- Natural models:
  - $Z_i \sim \text{Poisson}(\lambda_i)$
  - $Z_i \sim \text{Binomial}(n_i, p_i)$
  - $\lambda_i = \exp(x_i'\beta)$: spatial? other features?
  - $\text{logit}(p_i) = x_i'\beta$?

Discrete data

Jonathan Taylor
Department of Statistics
Stanford University

Models

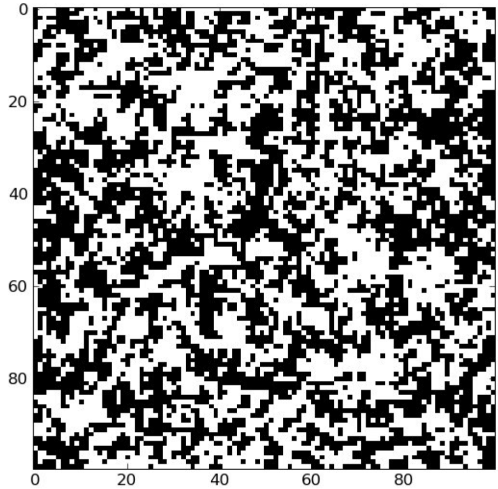
- How do we introduce spatial dependence in the Z_i's?
- If the Z_i's are Gaussian, all we need is a covariance function . . . more complicated for Poisson, Binomial.
- One approach: Markov random fields (a.k.a. graphical models)

Discrete data

Models

- How do we introduce spatial dependence in the Z_i's?
- If the Z_i's are Gaussian, all we need is a covariance function ...more complicated for Poisson, Binomial.
- One approach: Markov random fields (a.k.a. graphical models)
- Instead of defining a general MRF, we'll start with image models.

Simulation of an Ising model



Binary images

Ising model

- Let $L = \{(i,j) : 1 \leq i \leq n_1; 1 \leq j \leq n_2\}$ be our lattice of pixels.
- Binary images Z are elements of $\{-1,1\}^L$.
- Let $i \sim j$ be the nearest neighbours (with periodic boundary conditions).
- Given an inverse temperature $\beta = 1/T$

$$P(Z = z) \propto \exp\left(\beta \sum_{(i,j):i\sim j} z_i z_j\right)$$

$$= \frac{e^{\beta \sum_{(i,j):i\sim j} z_i z_j}}{\sum_{w \in \{-1,1\}^L} e^{\beta \sum_{(i,j):i\sim j} w_i w_j}}$$

Binary images

Ising model

- To a physicist
$$H(z) = \beta \sum_{(i,j): i \sim j} z_i z_j$$
is the potential energy of the system.
- Another representation:
$$H(z) = \beta \sum_{(i,j): i \sim j} ((z_i - z_j)^2 - 2) = \beta z' L z + C$$
where L is the graph Laplacian.

Binary images

Ising model

- Yet another representation

$$\sum_{j:i\sim j} z_i z_j = \#\{j : i \sim j, z_j = z_i\} - \#\{j : i \sim j, z_j \neq z_i\}$$
$$= 2\#\{j : i \sim j, z_j = z_i\} - 4$$

- The set

$$\{i : \exists j \sim i, z_j \neq z_i\}$$

can be thought of as the boundary of the black/white interface of z

- Leads to an interpretation

$$\sum_i \#\{j : i \sim j, z_j = z_i\} = \text{boundary length of } z.$$

Binary images

Ising model

- Conditional distributions are simple

$$P(z_i = 1 | z_j, j \neq i) \propto e^{\beta z_i \sum_{j:i \sim j} z_j} = P(z_i = 1 | z_j, j \sim i)$$

- Full joint distribution requires partition function

$$\sum_{w \in \{-1,1\}^L} e^{\beta \sum_{(i,j):i \sim j} w_i w_j}$$

which is complicated . . .

- Simulation of the Ising model is (relatively) easy

Gibbs sampler (Geman & Geman (1984))

Algorithm

```
def simulate(initial, beta, niter=1000):
    Z = initial.copy()
    for k in range(niter):
        for i in L:
            s = sum([Z[j] for j in nbrs(i, L)])
            odds = exp(2*beta*s)
            p = odds / (1 + odds)
            Z[i] = bernoulli(p)
    return Z
```

Gibbs sampler

Jonathan Taylor
Department of Statistics
Stanford University

Convergence

- For whatever initial configuration *initial*, as $niter \to \infty$

$$Z^{niter} \stackrel{niter \to \infty}{\Rightarrow} Z_\beta$$

  where Z_β is a realization of the Ising model.
- In fact, as a process $(Z^i)_{1 \leq i \leq niter}$ is a Markov chain that has stationary distribution Z_β.
- This is the basis of *most* of the MCMC literature . . .
- We'll see the Gibbs sampler again for more general MRFs.

Ising models

Adding an external field

We can also add a "mean" to the Ising model through an external field

$$P(Z = z) \propto \exp\left(\beta \sum_{(i,j):i\sim j} z_i z_j + \sum_i \alpha_i z_i\right)$$

Gibbs sampler with an external field

Algorithm

```
def simulate(initial, beta, alpha, niter=1000):
    Z = initial.copy()
    for k in range(niter):
        for i in L:
            s = sum([Z[j] for j in nbrs(i, L)])
            odds = exp(2*beta*s + 2*alpha[i])
            p = odds / (1 + odds)
            Z[i] = bernoulli(p)
    return Z
```

Denoising

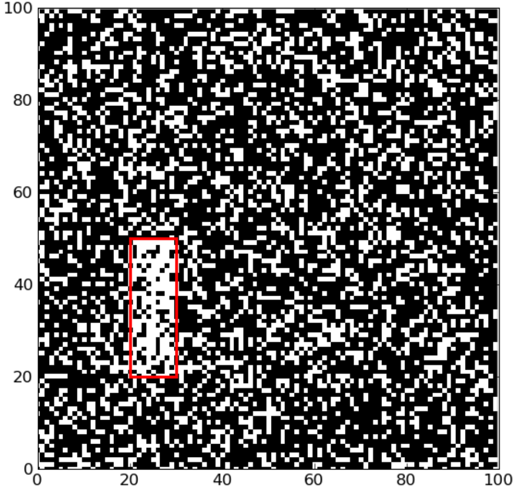
Model

- Suppose we observe a "noisy" image $Y \in \{-1, 1\}^L$ based on a "noise-free" image $Z \in \{-1, 1\}^L$
- A plausible choice for "noise" independent bit-flips

$$
\begin{aligned}
P(Y_i = y_i | Z = z) = P(Y_i = y_i | Z_i = z_i) \\
= \begin{cases} q & y_i = z_i \\ 1 - q & y_i \neq z_i \end{cases}
\end{aligned}
$$

- Goal: recover Z, the "noise-free" image.

Noisy image $q = 0.7$



Denoising

Model

- If Z were continuous, we might put some smoothness penalty on Z ...
- Recall the Laplacian interpretation of the potential in the Ising model

$$H(z) = \beta z' L z$$

- Suggests the following

$$\begin{aligned}\widehat{Z}_\beta &= \underset{Z \in \{-1,1\}^L}{\text{argmin}} \log L(Z|Y) - \beta \sum_{(i,j):i\sim j} z_i z_j \\ &= \underset{Z \in \{-1,1\}^L}{\text{argmin}} \sum_{i \in L} Z_i Y_i \text{logit}(q) - \beta \sum_{(i,j):i\sim j} z_i z_j\end{aligned}$$

Denoising

Model

- Our "suggested" estimator is the mode of an Ising model with parameter β and field $\alpha_i = \text{logit}(q) Y_i \dots$
- In Bayesian terms, if our prior for Z is Ising with parameter β, the posterior is Ising with a field dependent on the observations Y.
- If it's Bayesian, we can sample from the posterior using Gibbs sampler.

Denoising

Estimating Z

- Based on outputs of the Gibbs sampler (samples from the posterior), we can compute

$$\widehat{Z}_i = 1_{Z_i > 0}$$

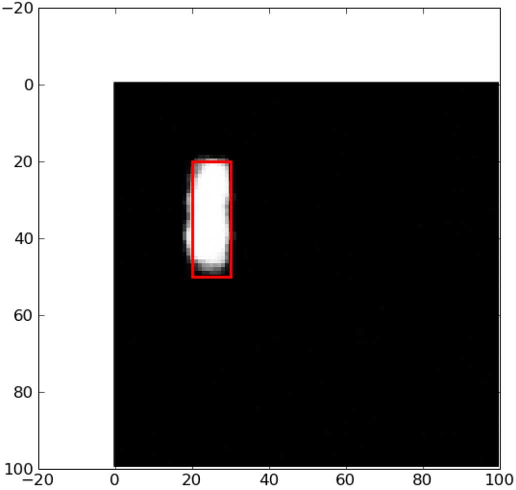
where

$$\bar{Z} = \sum_{j=l}^{niter} Z^j$$

is the posterior mean after throwing away l samples.

Posterior mean, $q = 0.7$

Jonathan Taylor
Department of Statistics
Stanford University



Denoising

Computing the MAP

- Our original estimator was the MAP for this prior.
- Finding the MAP is a combinatorial optimization problem: $\{-1, 1\}^L$ possible configurations to search through.
- A general approach is based on "simulated annealing". (Geman & Geman, 1984).

Denoising

Simulated annealing

- Basic observation: $\widehat{Z}_\beta(Y)$ is also the mode of

$$P_{T,\beta,Y}(z) \propto \exp\left(-\frac{1}{T}\left(\beta \sum_{(i,j):i\sim j} z_i z_j - \text{logit}(q)\sum_{i\in L} Y_i z_i\right)\right)$$

- BUT, for $T > 1$, $P_{T,\beta,Y}$ is more sharply peaked than $P_{1,\beta,Y}$.
- Depends on a choice of temperature schedule ...
- To prove things, one often has to assume $T_{iter} = \log(iter)$.

Simulated annealing

Algorithm

```
def simulate(initial, beta, alpha, temps):
    Z = initial.copy()
    for t in temps:
        for i in L:
            s = sum([Z[j] for j in nbrs(i, L)])
            odds = exp((2*beta*s + 2*alpha[i])/t)
            p = odds / (1 + odds)
            Z[i] = bernoulli(p)
    return Z
```

Denoising

Additive Gaussian noise

- Let the new noisy data be

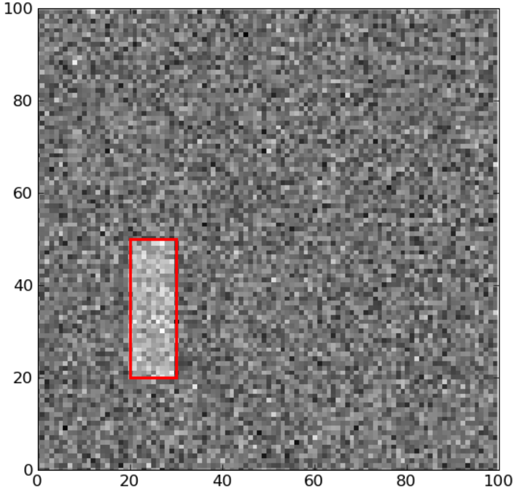
$$Y_i = Z_i + \epsilon_i$$

  with $\epsilon_i \overset{IID}{\sim} N(0, \sigma^2)$
- The field is now

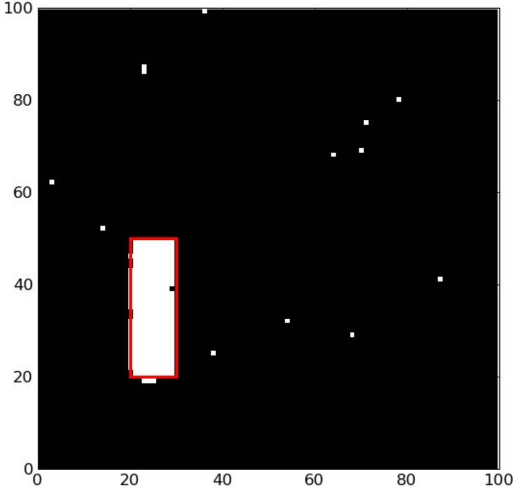
$$\alpha_i = Y_i z_i$$

  and $i \sim i$ (i.e. a new term in the neighbourhood relation).

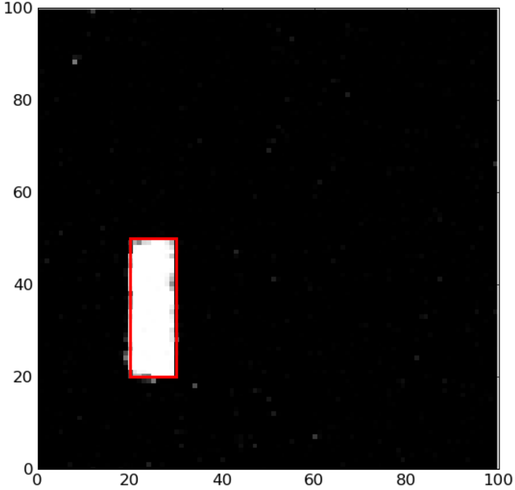
Additive noise



Additive noise



Additive noise

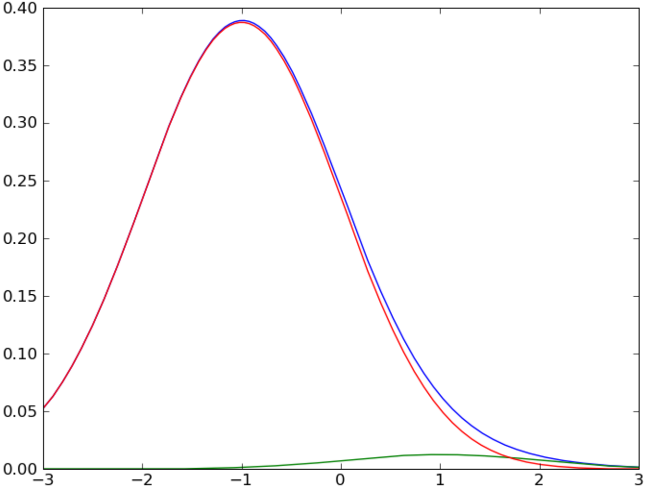


Denoising

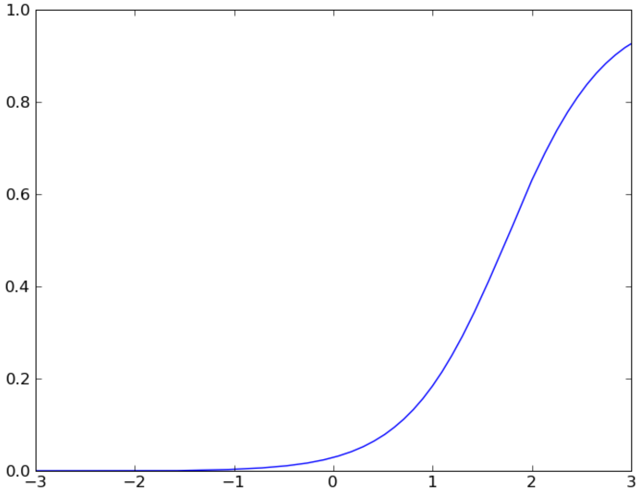
Mixture model

- The additive noise example can be thought of as a two-class mixture model.
- Suggests using LDA (or QDA if variances unequal) to classify.
- Problem: the mixing proportion is 0.03...

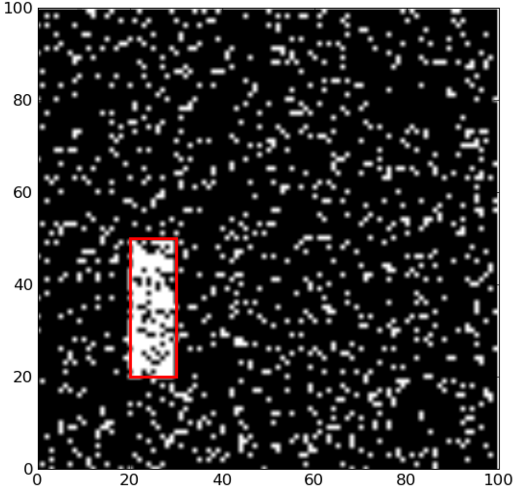
Mixture density



FDR curve



Thresholded



Where this leaves us

Markov random fields

- MRFs generally have distributions “like” an Ising model.
- Gibbs sampler can be used to simulate MRFs.
- Simulated annealing can also generally be used to find MAP (i.e. modes of an MRF).
- Bayesian image analysis is a *huge* field.