Spatial statistics and image analysis

Lecture 4

Konstantinos Konstantinou

Mathematical sciences
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden

Lecture's content

Todays lecture will cover

- Computational problems with kriging.
- Gaussian Markov random fields.
- Pattern recognition (LDA, QDA).
- Image moments.

Kriging

So far we looked at statistical models

$$Y_i = B(s_i)\beta + Z(s_i) + \epsilon_i, \quad i = 1, ..., N$$

where $\epsilon_i \sim N(0, \sigma_e^2)$ and $Z(s)$ is a zero mean Gaussian random field.

- $Y = (Y_1, ..., Y_N) \sim N(B\beta, \Sigma)$, with $\Sigma = \Sigma_X + \sigma_e^2 I$
- Kriging: If

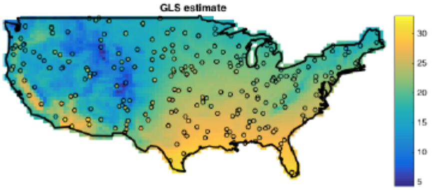
$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix}\right)$$

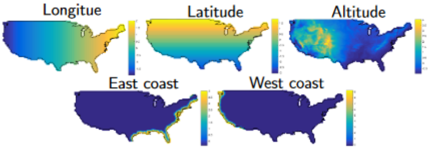
  then

$$X \mid Y \sim N(\mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y), \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX})$$

X is a random field at unobserved locations and Y are the observations.

Temperature example





Implementation aspects

1. Memory to store Σ scales as $\mathcal{O}(N^2)$.
2. The computation time for the kriging predictor scales as $\mathcal{O}(N^3)$.

Example: For an image x of size $N = n \times n$

	Time (s)	Memory (MB)
n = 50	1.1	47.7
n = 100	23.4	762.9
n = 150	272.5	3862.4

For an image of size 2500×2500 we need 20 years and $20GB$!

Sparse matrices

Definition: A Matrix Q is sparse if most of its elements are zero

- Efficient algorithms exist to deal with sparse matrices.
  1. Memory scales as $\mathcal{O}(N)$
  2. Computations scales as $\mathcal{O}(N^{\frac{3}{2}})$

Possible solutions:

- Force Σ to be sparse. This forces independence between variables.
- Force the precision matrix $Q = \Sigma^{-1}$ to be sparse. What does this correspond to?

Conditional independence

Definition: A and B are conditionally independent given C and we write $A \perp\!\!\!\perp B \mid C$, iff conditioned on C, A and B are independent, that is

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

Conditional independence is represented with an undirected graph $G = (V, E)$, where $V = \{1, ..., n\}$ is the set of vertices/nodes and $E = \{\{i, j\} : i, j \in V\}$ is the set of edges in the graph.

The neighbours of a node i are all nodes in G having an edge to i. i.e $N_i = \{j \in V : (i, j) \in E\}$

Gaussian Markov random field

Definition: A random vector X is called a Gaussian Markov random field (GMRF) with respect to the undirected graph $G = (V, E)$ with mean μ and precision matrix Q iff its density has the form

$$f_X(x) = (2\pi)^{-\frac{n}{2}} \mid Q \mid^{\frac{1}{2}} \, exp\left(-\frac{1}{2}(x-\mu)^T Q(x-\mu)\right) \qquad \text{and}$$

$$Q_{i,j} \neq 0 \iff \{i,j\} \in E, \quad \text{for all} \quad i \neq j.$$

Example: The simplest example of a GMRF is the AR(1) process

$$x_0 \sim N(0, \frac{1}{1-\alpha^2}), \qquad \alpha \in (-1, 1)$$

$$x_i = \alpha x_{i-1} + \epsilon_i, \qquad i = 1, ..., n \qquad \epsilon_i \sim N(0, 1)$$

Here Q is a tridiagonal matrix.

Simulating from a GMRF

How can we simulate a zero mean GMRF with precision matrix Q?

1. Compute the Cholesky factorization $Q = LL^T$.
2. Solve $L^T x = z$, where $z \sim N(0, \mathcal{I})$

Then x is a zero mean GMRF with precision matrix Q

Proof:

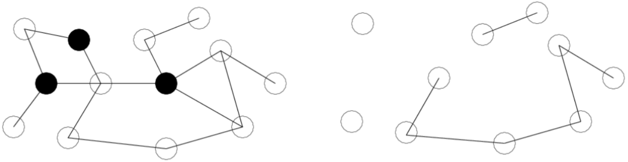
$$E(x) = E(L^{-T}z) = 0$$
$$Cov(x) = Cov(L^{-T}z) = L^{-T}Cov(z)L^{-1} = L^{-T}\mathcal{I}L^{-1} = (LL^T)^{-1} = Q^{-1}$$

Subgraph G^A

Definition: Let $A \subset V$, the subgraph G^A is the graph restricted to A.

- Remove all nodes not belonging to A and
- Remove all edges where at least on node is not A.

Example:



Conditional distributions

Theorem: Let $V = A \cup B$ where $A \cap B = \emptyset$, and let x be a GMRF wrt G with

$$X = \begin{bmatrix} X_A \\ X_B \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \quad Q = \begin{bmatrix} Q_{AA} & Q_{AB} \\ Q_{BA} & Q_{BB} \end{bmatrix}$$

then $X_A \mid X_B$ is a GMRF wrt to the subgraph G^A with $\mu_{A|B}$ and $Q_{A|B} > 0$ where

$$\mu_{A|B} = \mu_A - Q_{AA}^{-1} Q_{AB}(X_B - \mu_B) \quad \text{and} \quad Q_{A|B} = Q_{AA}$$

.

Note that

- $Q_{A|B} = Q_{AA}$ is known
- If Q_{AA} is sparse then $\mu_{A|B}$ is the solution of a sparse linear system.

Theorem: If $x \sim N(\mu, Q^{-1})$, then for $i \neq j$

$$x_i \perp\!\!\!\perp x_j \mid x_{-ij} \quad \Longleftrightarrow \quad Q_{ij} = 0$$

Corollary of the theorem

If we take $A = \{i\}$ and $B = \{-i\} := \{j : j \neq i\}$ then

$$\mu_i \mid \mu_{-i} = \mu_i - \sum_{j \in N_i} \frac{Q_{ij}}{Q_{ii}} (X_j - \mu_j) = \mu_i + \sum_{j \in N_i} \beta_{ij} (X_j - \mu_j)$$

$$Q_i \mid Q_{-i} = Q_{ii} = Var(X_i \mid X_{-i})^{-1} = \kappa_i$$

- The expectation of X_i is a weighted mean of the neighbouring X_j with weights β_{ij}.
- It is common to specify the GMRF through the full conditionals $P(X_i \mid X_{-i})$. These models are called Conditional autoregressions (CAR models).
- Since Q should be symmetric we require that $\kappa_i \beta_{ij} = \kappa_j \beta_{ji}$
- Also Q should be positive definite. We often deal with that issue by forcing Q to be diagonal dominant i.e

$$Q_{ii} > \sum_{j \in N_i} \mid Q_{ij} \mid \qquad \forall i$$

An example

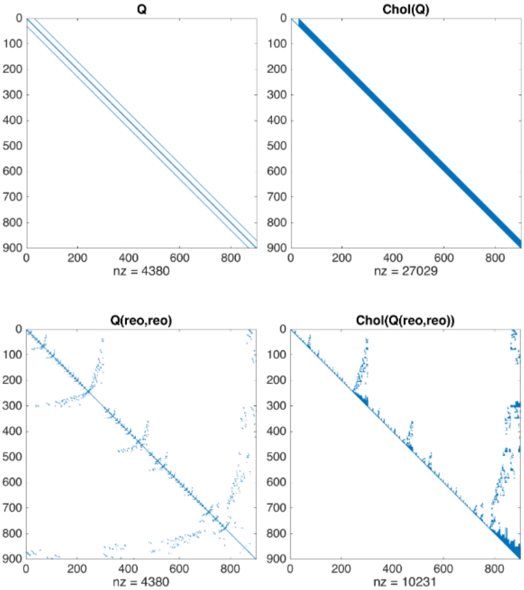
- Let $\epsilon > 0$, $\kappa_i = 4 + \epsilon^2$ $\forall i$ and $\mu = \mathbf{0}$.
- Now assume that the neighbourhood of a pixel i is given by the 4 nearest pixels with equal weights given by $\beta_{ij} = \frac{1}{4}$
- Then the precision matrix Q is given by

$$Q_{ij} = \begin{cases} 4 + \epsilon^2 & \text{if } j = i \\ -1 & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Sparsity of Q and L

- The main computation tasks in the GMRFs approach are
  1. Compute the Cholesky factorisation of $Q = LL^T$, and
  2. Solve $Lz = Q_{AB}(X_B - \mu_B)$ and $L^T x = z$.
- The crucial aspect of computations with GMRFs is that the Cholesky factor L is sparse, but it is less sparse than Q.
- The additional non-zero nodes are called fill-in.
- We can reduce the fill-in by reordering the nodes.
- Finding the optimal reordering is an NP-hard problem, but there are many fast methods for finding good reorderings. For example, the approximate minimum degree reordering is generally a good option.
- If you use reorderings, you should also reorder the observations, covariates, etc. using the same reordering.

An example


Q
0
100
200
300
400
500
600
700
800
900
0
200
400
600
800
nz = 4380
Chol(Q)
nz = 27029
Q(reo,reo)
nz = 4380
Chol(Q(reo,reo))
nz = 10231

Implementation aspects

- Image reconstruction using GMRF is more efficient than working with Σ.

Example: For an image x of size $N = n \times n$

	Time (s)	Memory (MB)
n = 50	0.012	0.21
n = 100	0.054	0.83
n = 150	0.177	1.88

Optimal discrimination with K=2 and $X \in \mathcal{R}$

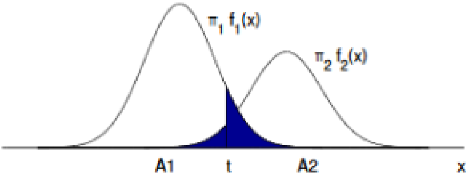
- Suppose we have two classes: Class 1 and Class 2
- A real valued feature variable X for each object to be classified.
- Let π_i be the prior probability of class i, i=1,2.
- Let f_i be the probability density of X for and observation from class i. Then we should choose class i over j if

$$\pi_i f_i(x) > \pi_j f_j(x)$$

Optimal discrimination with K=2 and $X \in \mathcal{R}$

"Proof": Choose the threshold t that minimizes the probability of misclassification

$$Pr(Misclassification) = \pi_1 \int_{A_2} f_1(x)dx + \pi_2 \int_{A_1} f_2(x)dx$$



Pr(misclassification) is given by the coloured area, and is minimized when t is the point where the curves intersect. Hence we should choose class i over j if

$$\pi_i f_i(x) > \pi_j f_j(x)$$

Discriminant analysis

- Suppose we have K classes
- Let X be a $d-$dimensional feature vector for each object to be classified and $f_i(x)$ the probability density for an observation from class i.
- Let π_i be the prior probabilities of class i

Then the posterior class probabilities are given by

$$P(Class = m \mid X = x) = \frac{P(Class = m)P(X = x \mid m)}{\sum_{j=1}^{K} P(Class = j)P(X = x \mid j)} = \frac{\pi_m f_m(x)}{\sum_{j=1}^{K} \pi_j f_j(x)}$$

We shall then prefer class i to class j when

$$\pi_i f_i(x) > \pi_j f_j(x)$$

Quadratic discriminant analysis

- Assume X is a d-dimensional feature vector with multivariate normal distribution $N(\mu_i, C_i)$ in class i, $i = 1, ..., k$
- Then we shall prefer class i to j if

$$\frac{1}{2}x^T(C_j^{-1} - C_i^{-1})x + (\mu_i^T C_i^{-1} - \mu_j^T C_j^{-1})x + \frac{1}{2}(\mu_j^T C_j^{-1}\mu_j - \mu_i^T C_i^{-1}\mu_i)$$
$$> ln\left(\frac{\pi_j \mid C_i \mid^{\frac{1}{2}}}{\pi_i \mid C_j \mid^{\frac{1}{2}}}\right)$$

- Since the border between the two regions in d-dimensional space where we should or should not prefer i to j is given by a quadratic surface we call this case ***Quadratic discriminant analysis(QDA).***

Linear discriminant analysis

- If $C_i = C$, for $i = 1, ..., k$ then we shall prefer class i to j if

$$(\mu_i - \mu_j)^T C^{-1}(x - \frac{1}{2}(\mu_i + \mu_j)) > ln\frac{\pi_j}{\pi_i}$$

- Proof: Set $C_i = C_j = C$ in the expression derived for QDA.
- As the expression above is linear in x this case is called ***linear discriminant analysis (LDA)***.

In MATLAB:
templateDiscriminant('DiscrimType','Linear') for LDA and
templateDiscriminant('DiscrimType','Quadratic') for QDA.

Parameter estimation

Suppose that we have a training set with n_i objects from class i.
Let the observation vectors be denoted $X_{im}, m = 1, ..., n_i, i = 1, ..., K$
Then

- $\hat{\pi}_k = \frac{n_k}{\sum_{i=1}^{K} n_i}, \quad k = 1, ..., K$
- $\hat{\mu}_k = \frac{1}{n_k} \sum_{m=1}^{n_k} X_{im}, \quad k = 1, ..., K$
- $\hat{C}_k = \frac{1}{n_k - 1} \sum_{m=1}^{n_k} (X_{im} - \hat{\mu}_i)(X_{im} - \hat{\mu}_i)^T, \quad k = 1, ..., K$

If we assume that the covariance matrices are equal then

- $\hat{C} = \frac{1}{\sum_{i=1}^{K}(n_i - 1)} \sum_{i=1}^{K} (n_i - 1) \hat{C}_i$

Moment features

Let $f = (f_{ij})$ be a binary/grey level image and A be a subset of pixels. The moment of order (p, q) in A is defined as

$$\mu_{pq}(A) = \sum_{(i,j)\in A} i^p j^q f_{ij}, \qquad p, q = 0, 1, \dots$$

Examples:

- μ_{00}: area = number of white pixels in A
- μ_{01}: sum over y
- μ_{10}: sum over x

centroid(A) $= \left(\frac{\mu_{10}}{\mu_{00}}, \frac{\mu_{01}}{\mu_{00}}\right) = (\bar{x}, \bar{y})$

Translation invariant moments

Image moments with respect to the centroid can be defined as

$$\mu_{pq}(A) = \sum_{(i,j)\in A} (i-\bar{x})^p (j-\bar{y})^q f_{ij} \qquad p+q>1$$

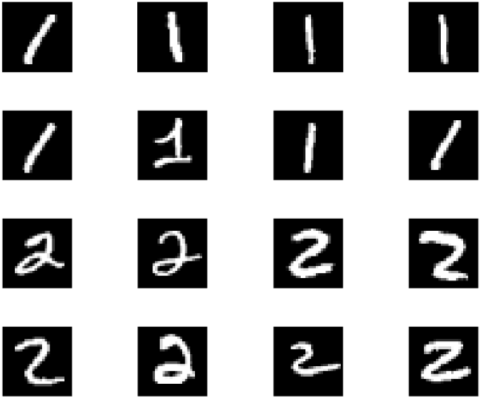
Central moments are invariant under translations.
Hu moments are translation, rotation and scale invariant moments. There are 8 such moments, the first two are

- $\mu_{02} + \mu_{20}$
- $(\mu_{20} - \mu_{02})^2 + 4\mu_{11}$

Invariant moments are useful for image classification.

Example: Handwritten digits 1 and 2. Moment features.

Aim: Classify the handwritten digits using the image moments μ_{11} and μ_{20}.



Example: Handwritten digits 1 and 2. Moment features.

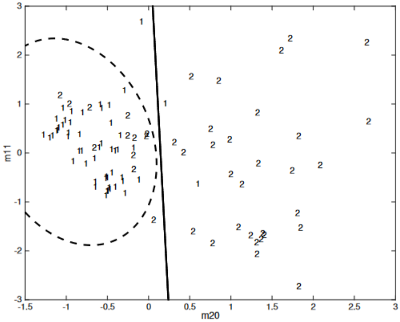


Figure: Plot of standardized moments μ_{11} versus μ_{20} for handwritten digits 1 and 2 among the first 400 digits in the MNIST data base together with the class boundaries corresponding to linear and quadratic discrimination.