Krigeage de données spatiales de température

On dispose des températures dans 150 villes de France un jour de canicule en août 2003. L'objectif est de compléter ces données sur toute la France, ici sur une grille de points 100x100. Et en particulier d'estimer la température à Rennes (longitude=-1.7, latitude=48)

On suppose que le processus des températures à cette date est un vecteur gaussien de moyenne m (indépendante de l'endroit) et de covariance

$$R_{vw} = ae^{-d(v,w)/b} \tag{1}$$

où a et b sont des paramètres et $d(v,w)$ est la distance entre les deux villes. Cette modélisation est à peu près la plus simple qui soit. La prédiction sur les points de la grille ne pourra se faire qu'après estimation de m, a et b.

Rq: Un théorème assure que la matrice des R_{vw} sera toujours $\geqslant 0$, ce qui n'a rien d'évident a priori. La condition de positivité est une restriction importante pour la modélisation des matrices de covariance.

(1) Estimation. Soit $X = (X_v)_{1\leqslant v\leqslant 150}$ le vecteur des températures observées, et $R_{XX} = R_{XX}(a,b)$ sa matrice de covariance, donnée par la formule (1), le logarithme de vraisemblance est

$$\mathscr{L}(m,a,b) = -\tfrac{1}{2}(X - m\mathbf{1})^T R_{XX}^{-1}(X - m\mathbf{1}) - \tfrac{1}{2}\ln(\det(R_{XX})). \tag{2}$$

Si l'on fait $v = w$ dans (1), on voit que a est la variance commune aux composantes de X, on va donc estimer m et a directement par la moyenne et la variance empiriques. L'optimisation de $\mathscr{L}$ en b se fera ensuite en utilisant la méthode BFGS proposée dans la routine `optim()` de R (il se trouve qu'il n'y a plus qu'un paramètre et donc une méthode plus simple fonctionnerait).

(2) Prédiction. Soit w une nouvelle ville, la température prédite est donnée par

$$E[Y|X] = m + R_{YX}R_{XX}^{-1}(X - m).$$

où R_{YX} est le vecteur des $ae^{-bd(w,v)}$, v variant parmi les villes observées.

(3) Comparaison des covariances fournies par le modèle estimé et d'estimées non-paramétriques.

(4) Estimation de l'erreur de prédiction du modèle par validation croisée.

(5) Estimation 2. Ici on se propose d'estimer $\mathscr{L}$, en minimisant directement $\mathscr{L}$ en (m,a,b). On compare cette méthode à la précédente en calculant par leave one out l'erreur de prédiction moyenne.

(6) Etude de l'erreur d'estimation par bootstrap par simulation.

Le programme.

`source("./KrigRout.R")`:

> Chargement de la routine `lldist(Long1,Lat1,Long2,Lat2)` qui le tableau de distances entre deux groupes de villes données par leur longitude et latitude. Cette routine renvoie donc une matrice de dimension `length(Long1)`×`length(Long2)`.
> Chargement `plotpoint(long,lat,temp)`: Une fonction de tracé des températures par des points de couleur.
> Chargement `covariogram(Dist,temp)`: Estimation non paramétrique de la covariance entre les températures de deux villes en fonction de leur distance. `Dist` est le tableau des distances et `temp` le vecteur des valeurs.

`read.table("./FranceTemp.txt",h=T)`: Charge la table individus/variables D contenant longitudes, lattitudes (degrés) et températures des villes observées, variables sont `lon, lat, data`.

`Dist=...`: Calcul de la matrice des distances. Calcul de m (ici `MT`), et de a (ici `VarT`).

`cost = function(par){...}` fonction de `par` (le paramètre b) qui renvoie $\mathscr{L}(m, a, b)$.

Suivent l'estimation et la prédiction dans une syntaxe standard.

Puis l'estimation de l'erreur de prédiction par leave one out, et la comparaison avec l'algorithme d'estimation des trois paramètres par maximum de vraisemblance.

Puis l'étude de l'erreur d'estimation par bootstrap par simulation.