Comparaison de deux modélisations probabilistes pour la classification de données géostatistiques : construction et analyse des algorithmes d'estimation

Benoît Allemand

Comparaison de deux modélisations probabilistes pour la classification de données géostatistiques : construction et analyse des algorithmes d'estimation

Benoît ALLEMAND

Elève en troisième année d'IUP Génie Mathématique et Informatique, spécialité Optimisation et Aide à la Décision.

Stage du 03/05/04 au 03/09/04

INRA centre d'AVIGNON, unité Biométrie

Sous la direction de Nathalie PEYRARD et Denis ALLARD

Table des matières

Pour commencer mon rapport concernant ce stage, je tiens à remercier tous les gens m'ayant permis de réaliser ma mission de la façon la plus agréable qui soit.

Je remercie Nathalie Peyrard et Denis Allard pour leur sujet de stage qui m'a beaucoup appris, leur convivialité ainsi que pour tout le temps qu'ils m'ont généreusement consacré, surtout N. Peyrard dont le bureau était proche du mien et qui eu la patience de répondre à mes régulières et incessantes interrogations.

Je souhaite de plus remercier toute l'équipe de Biométrie pour son accueil, ses réponses à mes problèmes et sa bonne humeur.

Merci à Rachid Senoussi pour m'avoir permis d'effectuer mon stage dans le Laboratoire Biométrie d'Avignon.

1 Environnement et Sujet

1.1 Cadre de Travail

1.1.1 L'INRA

L'institut National de la Recherche Agronomique fut créé en 1946. C'est un établissement public à caractère scientifique et technologique, placé sous la double tutelle des ministères chargés de la Recherche et de l'Agriculture.

Il a pour mission de :
- Oeuvrer au service de l'intérêt public tout en maintenant l'équilibre entre les exigences de la recherche et les demandes de la société ;
- Produire et diffuser des connaissances scientifiques et des innovations, principalement dans les domaines de l'agriculture, de l'alimentation et de l'environnement ;
- Contribuer à l'expertise, à la formation, à la promotion de la culture scientifique et technique, au débat science/société.

Pour cela d'importants moyens sont à sa disposition :
- 17 départements de recherche touchant l'agriculture, l'alimentation et l'environnement ;
- 21 centres régionaux répartis en près de 200 sites dans toute la France ;
- 257 unités de recherche (dont 128 associées à d'autres organismes) ;
- 80 unités expérimentales ;
- 131 unités d'appui et de service ;
- Un budget de 573 millions d'euros.

1.1.2 Le centre d'Avignon

Il fut fondé en 1953. Il tient aujourd'hui une place importante dans l'organisation de l'INRA tant par sa taille que par les thématiques de recherches qui lui sont confiées. Son implantation vauclusienne l'a, bien entendu, conduit à se pencher majoritairement sur l'agriculture méditerranéenne pour la qualité de la production maraîchère, fruitière et l'industrie régionale de la transformation agroalimentaire.

Ses recherches ont pour but la mise à jour de connaissances et de savoir-faire dans l'ingénierie de la gestion de l'environnement appliquée aux territoires cultivés et à la forêt méditerranéenne, mais aussi dans la maîtrise de la qualité des produits cultivés et transformés pour la santé du consommateur.

1.1.3 La Biométrie

L'unité Biométrie d'Avignon dépend du département Mathématiques et Informatique Appliquées dont les travaux visent à développer et à promouvoir l'utilisation de méthodes originales de statistique, analyse de systèmes et d'intelligence artificielle dans les recherches de l'INRA et la recherche agronomique en général. L'unité se place dans la thématique Espace et environnement, c'est à dire la modélisation spatiale et dynamique des systèmes écologiques et agronomiques.

Les applications à l'INRA couvrent des domaines tels que la science du sol, l'agronomie, la bioclimatologie, l'environnement, la génétique, l'épidémiologie et les forêts.

L'unité fonctionne avec 12 agents permanents, chercheurs, ingénieurs et 6 thésards, 1 équipe : Statistiques et modélisations spatiales.

La mission principale de l'unité est la recherche méthodologique en statistiques spatiales. Il s'agit de développer de nouveaux modèles et de nouvelles méthodes pour l'analyse statistique de données spatiales. Cet objectif est souvent réalisé en coopération avec des départements de l'INRA intéressés par l'introduction de la composante spatiale dans l'exploitation de données expérimentales (unité des "Recherches forestières méditerranéennes" par exemple). Ces recherches sont réalisées avec des équipes universitaires nationales et internationales.

Le champ des statistiques spatiales peut être définie par l'utilisation et le développement de méthodes statistiques pour l'étude de phénomènes où la prise en compte de la composante géographique est jugée à priori pertinente. Ses recherches sont essentiellement organisées autour de 2 axes :

• la statique géométrique avec l'étude et le développement de modèles pour les processus de points (simples ou marqués), les processus de fibres et de tâches, les tessallations aléatoires et les modèles booléens.

• la géostatistique avec en particulier le développement de modèles non stationnaires, l'analyse des données multivariables et spatio-temporelles ou encore les méthodes de classification en contexte spatial.

Le sujet de mon stage entre dans l'un des domaines de recherche de l'unité qui concerne la recherche de méthodes de classification en contexte spatial. Il s'agit de la mise en place d'algorithmes de classification de données spatiales qui peuvent soit être une image (imagerie médicale par exemple) soit des relevés régionaux (pluviométrie par exemple). Le but est de résumer un grand nombre d'informations dans un nombre restreint de classes ayant chacune des caractéristiques distinctes.

Comme on peut le constater, l'unité Biométrie d'Avignon recouvre à la fois les domaines biologiques et mathématiques. étant donné que l'informatique est devenue nécessaire afin de tester les nouveaux modèles mis en place par l'unité, je peux dire que ma formation va dans le sens d'un tel cadre de travail.

1.2 Contexte et Objectifs

1.2.1 Contexte de création du sujet

N. Peyrard travaille entre autre sur :
- La modélisation de données spatiales.
- L'estimation et la simulation de modèles probabilistes.
- La classification de données type image.

Tous ces travaux donnent lieux à des applications en épidémiologie ainsi qu'en analyse d'images. Les deux derniers concernent pleinement mon sujet de stage.

D. Allard travaille lui sur :
- Les données spatiales.
- La classification de données spatiales.
- La géostatistique.

Ainsi, le travail de thèse de N. Peyrard consistait en la modélisation probabiliste (par champ de Markov caché, HMRF en anglais) ainsi que la réalisation d'un algorithme permettant la classification de données spatiales régulièrement réparties (images).

Pour expliquer la problématique à résoudre, voilà un exemple :

Lors de l'analyse de sols, des spécialistes récoltent des données (niveau de pluviométrie, par exemple) qu'ils enregistrent. Ensuite, ils désirent classifier ce sol en différentes zones de façon automatique et de sorte que des points d'une même classe aient des données proches et que des points proche spatialement soient dans la même classe. Une façon de faire est de modéliser les données

par un modèle probabiliste. On va alors demander à notre algorithme de reconstituer un certain nombre de classes (3 par exemple) pour donner 3 régions avec 3 caractéristiques différentes au sein desquelles les points classifiés dans une même classe répondent tous à une même loi probabiliste. Chaque classe se vera attribué certaines caractéristiques telles qu'une moyenne 'm' et une variance 'v' de niveau de pluie de façon à ce que les données de cette classe répondent à une loi normale de paramètres 'm' et 'v'.

Au départ, un algorithme réalisé par N. Peyrard existait et était capable de classifier des données grâce à un l'algorithme SF-EM basé sur le modèle HMRF. De plus, cet algorithme n'était utilisable que sur des données spatiales régulièrement réparties (type image).

D. Allard se posait le même problème de classification spatiale mais pour des données géostatistique donc irrégulièrement réparties. Pour cela il a fait un autre choix de modélisation, un modèle géostatistique et s'est posé la question encore non résolue d'un algorithme d'estimation de ce modèle.

Ce qui précède est l'objet principal de mon stage mais le but final de ce sujet de recherche est d'arriver à partir des deux premiers modèles à créer un dernier modèle ainsi qu'un algorithme répondant à ce modèle que nous appellerons modèle complet. Le but est de réaliser un modèle qui semble plus proche de la réalité en combinant les avantages des deux premiers.

Enfin, il faut bien comprendre que les données réelles ne sont jamais issues de l'un de ces 3 modèles. Ces 3 modèles complémentaires sont là afin de se rapprocher au mieux de la réalité tout en rendant possible la réalisation d'algorithmes liés à ces modèles. L'utilisateur pourra ensuite choisir quel modèle semble le plus performant pour représenter ces données et ainsi choisir l'algorithme le plus apte à les classifier.

Pour résumer le but du stage était de disposer de différents algorithmes propres à ces modèles bien définis et de les comparer soit sur des simulations, soit sur des cas réels afin de connaître leurs avantages et leurs inconvénients.

Afin de distinguer les modèles en cours, voyons les carctéristiques propres à chacun :
- Le modèle HMRF est basé sur la corrélation spatiale des classes, c'est à dire que deux points proches spatialement ont de fortes chances d'être de la même classe.
- Le modèle Géostatistique est basé sur la corrélation spatiale des données, c'est à dire que deux points proches spatialement et de même classe ont de fortes chances d'avoir des valeurs très proches.
- Le modèle Complet est basé sur la corrélation spatiale des classes et des données, c'est à dire que deux points proches spatialement ont de fortes chances d'être de la même classe et ils ont aussi de fortes chances d'avoir des valeurs très proches.

1.2.2 Cahier des charges

Avant même de commencer ce stage, lors d'une rencontre avec N. Peyrard, nous avions déjà discuté du sujet et du cahier des charges à suivre. étant donné que ce stage est un sujet de recherche, le cahier des charges fut défini de façon la plus basique possible, les thèmes étaient bien définis mais nous ne pouvions pas anticiper les résultats et les tournures à prendre dans le futur.

Le cahier des charges :

• Assimilation des modèles statistiques ainsi que de l'algorithme SF-EM puis du code. Ce travail a été continu tout au long du stage, j'ai beaucoup appris la première semaine, ensuite j'ai assimilé ce qui était nécessaire tout au long du stage.

• Application de l'algorithme SF-EM à des données irrégulièrement réparties. Pour cela, il m'a fallu assimiler le code afin de le modifier correctement. Le but étant de pouvoir appliquer SF-EM à des données géostatistiques.

• Simulation des 2 nouveaux modèles (Géostatistique et Complet) dans le sens de tester ensuite les performances des algorithmes correspondants à ces modèles. Ici, c'est l'assimilation des modèles qui fut importante, j'ai du comprendre les modèles afin de les insérer dans le code.

• Recherche et mise en place d'un algorithme (Geo-EM) afin de classifier des données selon le modèle géostatistique.

• Analyse du nouvel algorithme et comparaison avec l'algorithme existant. Ce travail fut en fait la mise en place de batteries de simulation afin de tester l'efficacité et la robustesse du nouvel algorithme.

• Recherche et la mise en place d'un algorithme basé sur le modèle complet (CP-EM).

• Analyse du nouvel algorithme (CP-EM) et comparaison avec l'algorithme existant.

1.3 Moyens

Pour la réalisation de mon stage, différents moyens ont été mis à ma disposition, je peux dire qu'ils on étés suffisants et m'ont permis d'être efficace.

1.3.1 Matériel

Durant tout mon stage, je me suis retrouvé dans un sas regroupant deux bureaux, j'ai eu la chance de travailler sur un ordinateur très récent (pentium 4) ainsi que dans une pièce calme.

Tout était installé sur cet ordinateur, je n'ai eu aucun manque matériel, d'autant plus que l'unité Biométrie possède deux imprimantes dont une couleur.

1.3.2 Logiciel

Sur l'ordinateur, Windows et Linux étaient installés, pour plus de simplicité je n'ai travaillé que sous Linux. Même si je n'avais pas eu beaucoup l'occasion de m'en servir précédemment, connaissant le système Unix, il n'y eu aucun problème.

En ce qui concerne le codage, tout était et est resté en C (déjà connu, proche du C++). De plus quelques routines d'analyses ont été réalisées sous R.

Enfin, étant donné que tout le monde utilise Latex pour faire les rapports (il est fait pour créer des formules mathématiques contrairement à Word), tous mes rapports ont été faits sous Latex que je n'avais jamais utilisé avant cela. L'apprentissage de Latex me pris donc aussi un certain temps (en cumulé, environ 2 jours de travail).

Enfin, l'unité dispose d'une bibliothèque très complète tant au niveau mathématique qu'informatique ce qui m'a permit de trouver facilement des réponses à toutes mes questions.

1.3.3 Organisation / Encadrement

Mon bureau étant proche de N. Peyrard et pour différentes autres raisons, j'ai principalement travaillé avec elle. D. Allard intervenait principalement lors de réunions de bilans ou quand seul lui pouvait donner une réponse du fait de son expérience en géostatistique.

De plus, tout le laboratoire est très soudé et si un problème se pose, il suffit de poser une question (lors d'un café ou d'un repas, ou encore directement à un spécialiste) pour qu'un grand nombre de réponses arrivent.

Pour tout ce qui concerne l'avancement du stage, il m'est arrivé de prendre seul des décisions qui ne se sont pas toujours révélées bonnes mais qui m'ont permis de comprendre mes erreurs. Malgré tout, la plus part du temps, on décidait ensemble du travail à fournir. Ensuite, les résultats étaient analysés en priorité par moi puis avec mes maîtres de stages.

Lors de ce stage, j'ai du apprendre l'autonomie, je me suis rendu compte que je demandais trop facilement de l'aide dans des cas où je pouvais en cherchant trouver seul la solution. Je pense que cela est aussi important que la communication, un stage sert aussi à mieux se connaître et trouver ses limites.

Sinon, il n'y a eu aucun problème de communication et d'organisation. La communication avec mes maîtres de stage et le reste de l'équipe c'est toujours bien passée ce qui m'a permis d'avancer sans trop d'embûches.

2 Présentation des modèles spatiaux

Pour commencer le stage, il a fallu que je passe un certain temps à assimiler les 3 modèles ainsi que l'algorithme SF-EM, tant au niveau mathématique qu'informatique.

Je vais exposer toutes les bases de modélisation probabilistes nécessaires à la compréhension de mon stage, en conclusion seulement, je parlerais de mes difficultés et facilités d'adaptation à ce stage.

2.1 Approche probabiliste pour la classification de données spatiales

On peut définir le problème de classification de données spatiales de la façon suivante :

- Soit une région spatialisée au sein de laquelle nous avons n points. Ces points sont soit répartis sur une grille (images), soit irrégulièrement répartis (données géostatistiques).

- Chaque point i est défini par :
  - ◦ Des coordonnées (x_i, y_i) afin de le repérer dans l'espace.
  - ◦ Ses voisins qui peuvent être soit les pixels (4 ou 8 plus proches) sur une image, soit les points appartenant à un cercle de voisinage de rayon r en géostatistique.
  - ◦ Une(des) donnée(s) observée(s) X_i qui peuvent être soit une valeur indiquant une intensité de réaction, soit une mesure dans le sol de cuivre, de cobalt, de pluviométrie....
  - ◦ Une donnée inconnue Y_i qui est la classe du point i. Y_i est définie comme un vecteur de taille K(nombre de classes) avec des 0 partout et un 1 à la ligne correspondant à la classe de i. ainsi, $Y_i{}^*Y_j = 1$ si i et j sont de même classe, 0 sinon.

Le but est ici de retrouver la classe initiale Y_i de chaque point i en fonction des paramètres connus comme la valeur X_i, la valeur des voisins, les classes estimées des voisins... Nous effectuons la classification par une approche probabiliste :

- On choisit un modèle (HMRF, géostatistique ou complet) définissant la façon dont les données X_i et Y_i sont modélisées.

- On connaît le nombre de classes K.

- On cherche à maximiser la vraisemblance des données pour estimer les paramètres du modèle.

- On calcule la classification qui maximise la probabilité à posteriori des classes pour les valeurs estimées des paramètres.

Voyons maintenant de façon approfondie quels sont les différents modèles et comment fonctionnent les algorithmes, lorsqu'ils existent, permettant de résoudre les problèmes définis par ces modèles.

2.2 Les différents modèles

Pour classifier les données, nous avons retenu trois modèles pour représenter les données observées et celles inconnues.

Pour comparer ces 3 modèles et leurs algorithmes de classification associés, nous partons d'un modèle de base, le modèle de mélange associé à l'algorithme EM.

Pour présenter ces modèles, définissons :
- e_l = matrice de taille K avec des 0 partout et un 1 à la ligne l.
- n_k = nombre de points de la classe k.
- a_k = poids de la classe k.
- π_k = proportion de la classe k.
- $N(i)$ = ensemble des voisins de i
- $h_{i,j}$ = distance du point i au point j.

2.2.1 Modèle de mélange et algorithme EM

Le modèle mélange ne suppose aucune dépendance spatiale tant au niveau des données que des classes. C'est donc le modèle le plus simple.

Pour mieux comprendre le modèle, je vous présente un exemple sur la mesure de taille des ailes d'oiseaux afin de classifier les mâles des femelles car identifier leur sexe est compliqué donc les perturbe trop. On voit sur les graphiques que deux classes distinctes se forment, ceci est un modèle mélange à 2 classes :

Longueur	82	83	84	85	86	87	88	89	90
Fréquence	5	3	12	36	55	45	21	13	15
Longueur	91	92	93	94	95	96	97	98	
Fréquence	34	59	48	16	12	6	0	1	

TAB. 8.1 – Données passereau

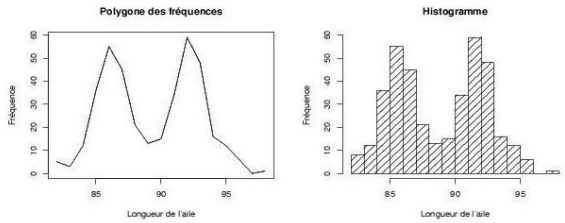


FIG. 8.1 – Longueurs des ailes en mm de 381 passereaux

L'algorithme classiquement utilisé pour estimer le modèle est appelé EM.

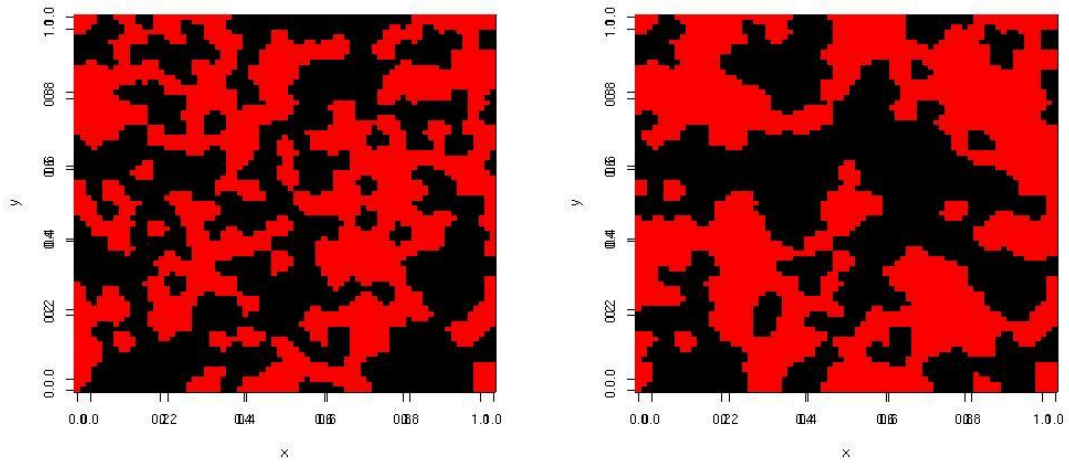
EM signifie Expectation, Maximisation car le fonctionnement d'un tel algorithme est basique :
◦ On itère les deux étapes suivantes :
- (E) Calcul des probabilités d'appartenance aux classes de chaque point en fonction des paramètres estimés actuellement. C'est ce qu'on appellera la classification floue.
- (M) Maximisation de l'estimation de la logvraisemblance des paramètres sachant les données en fonction de la classification floue actuelle.

La classification floue signifie que chaque point a une probabilité d'appartenir à chaque classe et ce n'est pas forcément 1 ou 0. Soit $t_{i,k}$ cette probabilité pour le point i et la classe k, alors $t_{i,k} \in [0,1]$.

Le modèle de mélange, estimé par EM, sera la référence pour comparer les performances des autres modèles et algorithmes, car c'est le plus simple.

2.2.2 Le modèle HMRF

Le modèle HMRF est basé sur la corrélation spatiale des classes. Ainsi, plus un point a de voisins d'une même classe, plus il a de chances d'appartenir à cette classe, le tout proportionnellement à un coefficient de régularité spatiale noté β. Plus β est grand, plus la corrélation spatiale est forte et les classes sont regroupées.



(a) (b)

Figure 1: Un exemple graphique de deux jeux de données issus d'un modèle HMRF, (a) $\beta = 0.1$ et (b) $\beta = 0.6$.

Pour définir ce modèle, étudions les lois des X_i et des Y_i.

Définissons la probabilité que le point i appartienne à la classe k sachant les classes des autres points :

$$f(Y_i = k/Y_{-i}) = \frac{e^{[a_k + \beta * \sum_{j \epsilon N(i)} e_k * Y_j]}}{\sum_{l=1}^{K} e^{[a_l + \beta * \sum_{j \epsilon N(i)} e_l * Y_j]}} \tag{1}$$

On voit donc bien ici l'action de β et des voisins du point i. Ce modèle s'appelle champ de Markov.

On peut voir sur la figure 1 l'effet du β, lorsque $\beta = 0.1$, les classes ne sont pas lisses alors que quand $\beta = 0.6$, les classes sont plus regroupées et donnent donc un ensemble plus lisse.

Définissons la loi des X sachant Y : la loi X_i sachant que Y_i=k est une loi normale de moyenne m_k et de variance V_k :

$$f(X_i/Y_i = k, Y - i) = f(X_i/Y_i = k) \sim \mathcal{N}(m_k, V_k) \tag{2}$$

De plus, il n'y a aucune corrélation spatiale des données.

Ici, il y a donc une corrélation spatiale entre les classes des points et non entre leurs valeurs. On associe à ce modèle l'algorithme SF-EM (Simulated Field - Estimation Maximisation) dont le fonctionnement est relativement proche de EM avec l'estimation du coefficient de régularité spatiale β en plus.

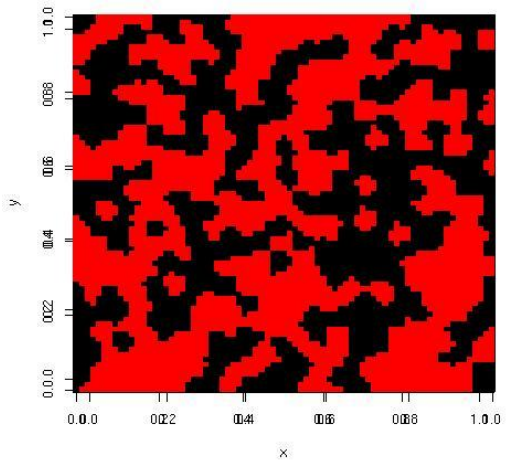


Figure 2: Un exemple graphique d'un jeu de données issus d'un modèle Géostatistique.

2.2.3 Le modèle Géostatistique

Le modèle Géostatistique est basé sur la corrélation spatiale des valeurs des points. Ainsi, plus deux points d'une même classe sont proches, plus ils ont de chances d'avoir des valeurs proches, le tout proportionnellement pour chaque classe k à un paramètre de corrélation spatiale des données noté b_k. Plus b_k est grand, plus la corrélation spatiale des données est forte.

Pour définir ce modèle, étudions les lois des X_i et des Y_i.

Définissons la loi des Y. La probabilité que le point i appartienne à la classe k est :

$$f(Y_i = k/Y_{-i}) = f(Y_i = k) = \pi_k \quad (3)$$

$$= \frac{e^{a_k}}{\sum_{l=1}^{K} e^{a_l}} \quad (4)$$

Ici, il n'y a aucune corrélation spatiale des classes, on détermine juste les probabilités d'appartenance aux classes en fonction des proportions de chacune. On remarque que l'équation (4) est identique à l'équation (1) avec $\beta = 0$.

On peut voir sur la figure 2 qu'avec un modèle géostatistique, les classes ne sont pas lisses, c'est l'équivalent au niveau des classes du modèle HMRF avec $\beta = 0$.

Définissons la loi des X sachant Y. La densité des données X sachant les classes :

$$f(X|Y) = \frac{e^{\frac{-1}{2}}(X-\mu)^t \mathbf{\Sigma}^{-1}(X-\mu)}{(2\pi)^{n/2}\sqrt{det(\mathbf{\Sigma})}} \quad (5)$$

où $\mathbf{\Sigma}$ est la matrice de variance/covariance :

$$\mathbf{\Sigma} = \sum_k V_k * R_k \quad (6)$$

$$R_k = \{(R_k)_{i,j}\}\ \forall_{i,j} \quad (7)$$

$$(R_k)_{i,j} = \begin{cases} t_{i,k} \ \forall_{i=j}. \\ \\ t_{i,k} * t_{j,k} * \rho_k(h_{i,j}) \ \forall_{i \neq j}. \end{cases} \tag{8}$$

où les $t_{i,k}$ sont durs :

$$t_{i,k} \epsilon \{0,1\} \ \forall_{i,k}, \ \sum_k t_{i,k} = 1 \ \forall_i. \tag{9}$$

Avec :

$$\rho_k(h_{i,j}) = e^{\frac{-h_{i,j}}{b_k}} \tag{10}$$

C'est ici, que de manière relativement complexe apparaît la corrélation spatiale des données pour des points d'une même classe.

On voit donc qu'ici, il n'y a que corrélation entre les valeurs des points et non les classes. Nous verrons en 3.1 que l'algorithme Geo-EM a été choisi afin de classifier au mieux des données répondant à ce modèle.

2.2.4 Le modèle Complet

Le modèle complet se définit comme un mélange des deux précédents, il est donc bas? sur la corrélation spatiale des classes et des données. paragraph La loi des Y est donné par l'équation (1) et la loi des X sachant Y par l'équation (5). paragraph On associe à ce modèle l'algorithme CP-EM dont le fonctionnement serra entre celui de SF-EM (estimation de β) et celui de Geo-EM (estimation des b_k).

2.3 Les algorithmes de simulation

Afin de mesurer l'efficacité des algorithmes de classifications, il fallait être capable de simuler des données répondant à ces modèles et y comparer les algorithmes dessus. C'est pourquoi j'ai implémenté dans le programme les algorithmes de simulations des modèles Géostatistique et Complet, le modèle HMRF pouvant déjà être simulé.

Nous représentons ici les méthodes de simulation des ces 3 modèles afin d'aider à leur compréhension.

2.3.1 Simulation pour un modèle HMRF

Voyons comment sont données les valeurs X_i et Y_i pour chaque point i lors d'une simulation de modèle HMRF.

Pour fonctionner, ce modèle doit disposer :

- D'un coefficient de régularité spatiale noté β qui, plus il est fort, plus il augmente la probabilité que des voisins soient de la même classe.
- Des caractéristiques spatiales de la simulation (nombre de points, coordonnées des points...).
- Des caractéristiques de chaque classe (moyenne, variance, proportion...).

L'algorithme de simulation :
- Donner une classe à chaque point en fonction des poids des classes afin d'initialiser le processus.
- Pour un nombre N de boucles, lancement de l'algorithme de Gibbs :
  ◦ Pour chaque point i :
    • Calculer la probabilité d'appartenance à chaque classe k en fonction de son voisinage (équation 1).
    • Tirer un nombre suivant une loi uniforme sur [0, 1].
    • Associer la classe de i à ce nombre en fonction des probabilités calculées précédemment. Pour cela, ranger les probabilités dans un tableau allant de 0 à 1 et regarder dans quelle intervalle de probabilité se trouve q.
- Une fois que chaque point appartient à une classe.
- Donner les valeurs X_i à chaque point i suivant une loi normale de paramètres la moyenne et la variance de la classe k à laquelle appartient i (équation 2).

2.3.2 Simulation pour un modèle Géostatistique

Voyons comment sont données les valeurs X_i et Y_i pour chaque point i lors d'une simulation de modèle Géostatistique.

Pour fonctionner, ce modèle doit disposer :

- D'un paramètre de corrélation spatiale pour chaque classe k noté b_k qui, plus il est fort, plus il augmente la probabilité que des voisins aient des valeurs proches.
- Des caractéristiques spatiales de la simulation (nombre de points, coordonnées des points...).
- Des caractéristiques de chaque classe (moyenne, variance, proportion...).

L'algorithme de simulation :
- Donner une classe de façon aléatoire à chaque point (équation 4).
- Une fois que chaque point appartient à une classe.
- Fabriquer la matrice de variance/covariance $\mathbf{\Sigma}$, à partir de cette matrice et suivant un algorithme rmultnorm relativement complexe donner une valeur X_i à chaque point i en fonction de son voisinage de même classe que lui (équation 5).

2.3.3 Simulation pour un modèle Complet

Voyons comment sont données les valeurs X_i et Y_i pour chaque point i lors d'une simulation de modèle Complet.

Pour fonctionner, ce modèle doit disposer :

- D'un coefficient de régularité spatiale noté β qui, plus il est fort, plus il augmente la probabilité que des voisins soient de la même classe.
- D'un paramètre de corrélation spatiale pour chaque classe k noté b_k qui, plus il est fort, plus il augmente la probabilité que des voisins aient des valeurs proches
- Des caractéristiques spatiales de la simulation (nombre de points, coordonnées des points...).
- Des caractéristiques de chaque classe (moyenne, variance, proportion...).

L'algorithme de simulation :
- Donner une classe à chaque point en fonction des poids des classes afin d'initialiser le processus.
- Pour un nombre N de boucles, lancement de l'algorithme de Gibbs :
  ◦ Pour chaque point i :
    • Calculer la probabilité d'appartenance à chaque classe k en fonction de son voisinage (équation 1).
    • Tirer un nombre suivant une loi uniforme sur [0, 1].

• Associer la classe de i à ce nombre en fonction des probabilités calculées précédemment. Pour cela, ranger les probabilités dans un tableau allant de 0 à 1 et regarder dans quelle intervalle de probabilité se trouve q.

- Une fois que chaque point appartient à une classe.
- Fabriquer la matrice de variance/covariance $\mathbf{\Sigma}$, à partir de cette matrice et suivant un algorithme rmultnorm relativement complexe donner une valeur X_i à chaque point i en fonction de son voisinage de même classe que lui (équation 5).

2.3.4 Résumé

Pour résumer toutes ces informations, on peut donc poser le problème sous forme d'un tableau :

problème	Classification de données spatiales géostatistiques		
	Résolution du problème, choix du modèle		
Modèle	HMRF (Ch 2.2.2)	Géostatistique (Ch 2.2.3)	Complet (Ch 2.2.4)
Algorithme	SF-EM	Geo-EM	CP-EM

Pour chaque modèle, il y a un algorithme de classification et pour le tester, un algorithme de simulations et une boite à outil (programme existant) dans laquelle on retrouve les deux.

3 Présentation de l'algorithme de classification de données géostatistiques et de ses résultats

3.1 Geo-EM sous ses différentes versions

Ceci est la partie principale de mon stage, ici il a fallu "inventer" un algorithme capable de classifier des données par un modèle géostatistique. Nous verrons qu'ici, il n'y a pas un algorithmes mais 7. En fait ces algorithmes sont des variations plus ou moins importante d'un algorithme de base, seront définis :

- Geo-EM-Vrai.
- Geo-EM-Approximé versions 1 et 2.
- Geo-EM-B versions 1 et 2.

voilà donc une présentation de Geo-EM sous trois de ses formes, elles correspondent aux trois principales étapes dans le développement de l'algorithme. On aura tout d'abord Geo-EM-Vrai qui est l'algorithme répondant le plus formellement au modèle, ensuite Geo-EM-Approximé qui permet une accélération de l'algorithme et enfin Geo-EM-B1 qui se sera révèle le plus efficace.

Avant de commencer le découpage du fonctionnement de l'algorithme, nous présentons les notations :

i et j représenteront des points.
k et l représenteront les classes.
m représentera les tranches de distances h entre les points.
y_i = observation au point i.
-i en indice = tout sauf i (ex : $t_{-i,k}$ = probabilités pour tous les points sauf i et $t_{-i} = \{t_{-i,k}, \forall k\}$)
$h_{i,j}$ = distance du point i au point j.
les paramètres à estimer sont :
V_k = variance de la classe k.
μ_k = moyenne de la classe k.
π_k = proportion de la classe k dans le modèle.
b_k = paramètre pour la création de $\mathbf{\Sigma}$, matrice de variance/covariance.
K = nombre de classes, n = nombre de points, n_h = nombre de découpages des distances h (h_1=0.1, h_2=0.2, ...).

Soit aussi la matrice de variance/covariance du modèle géostatistique $\mathbf{\Sigma}$ défini dans l'équation (6), chapitre 2.2.3.

Tout d'abord, une idée du fonctionnement de l'algorithme avant de rentrer dans les détails. On désire "mimer" EM pour modèle de mélange, c'est à dire avec des $t_{i,k}$ flous. Pour cela, on va remplacer dans les formules de EM pour mélange les probabilités pour des données spatialisées. Cependant, il faut trouver une façon d'estimer b_k car ce paramètre n'existe pas dans le mélange.

L'algorithme agit en un certain nombre d'itérations qui lui permettent de converger vers une solution. Avant de commencer ces itérations, il faut initialiser l'algorithme. Pour cela, on range les points par ordre croissant de valeurs, ensuite on les découpe en K classes de tailles équivalentes. De cette façon, l'algorithme part d'une classification initiale dure ($t_{i,k} \epsilon \{0, 1\}$) pour sa première itération.

Suite à cela, on peut estimer les autres paramètres (variance, moyenne ..) à partir des $t_{i,k}$ connus.

fonctionnements de l'algorithme à l'étape q+1 :

- Pour tout $i \in \{1, n\}$, $k \in \{1, K\}$:
  - Estimer $t_{i,k}^{q+1}$ (I)
- Pour toute classe $k \in \{1, K\}$:
  - Estimer π_k^{q+1} (II)

- Estimer μ_k^{q+1} (III)
- Estimer V_k^{q+1} (IV)
- Estimer b_k^{q+1} (V)

Tous les algorithmes explorés fonctionnent de la même façon pour estimer π_k^{q+1} et b_k^{q+1}, pour le reste, cela varie.

Etudions plus précisément le fonctionnement de ces algorithmes au cas par cas.

3.1.1 Geo-EM-V

- Estimation (I) :

Ici, on utilise une méthode de krigeage afin d'estimer les $t_{i,k}^{q+1}$. Cela fonctionne de la façon suivante :

$$t_{i,k}^{q+1} = \frac{\pi_k^q * f(y_i|y_{-i}, t_{i,k} = 1, t_{-i}^q, \theta^q)}{\sum_l \pi_l^q * f(y_i|y_{-i}, t_{i,l} = 1, t_{-i}^q, \theta^q)} \tag{11}$$

Avec

$$f(y_i|y_{-i}, t_{i,k} = 1, t_{-i}^q, \theta^q) \sim \mathcal{N}(m_{i,k}, \sigma_{i,k}^2) \tag{12}$$

obtenue par krigeage comme suit :
On définit :

$$\mathbf{\Sigma}_{-i}^q = \{(\mathbf{\Sigma}_{-i}^q)_{i',j'}\} \ \forall_{i' \neq i, j' \neq i} \tag{13}$$

$$(\mathbf{\Sigma}_{-i}^q)_{i',j'} = \begin{cases} \sum_k V_k^q * t_{i',k}^q \ \forall_{i'=j'}. \\ \sum_k V_k^q * t_{i',k}^q * t_{j',k}^q * \rho_k^q(h_{i',j'}) \ \forall_{i' \neq j'}. \end{cases} \quad \forall_{i' \neq i, j' \neq i} \tag{14}$$

$$\Sigma_{i,k}^q = \{t_{j,k}^q * V_k^q * \rho_k^q(h_{i,j})\} \ \forall_{j \neq i} \tag{15}$$

μ_{-i}^q est le vecteur des moyennes de chaque point sans la ligne i.

$$\mu_{-i}^q = \{\sum_{k=1}^{K} t_{j,k}^q * \mu_k^q\} \ \forall_{j \neq i} \tag{16}$$

$$m_{i,k} = \mu_k^q + (\Sigma_{i,k}^q)^t * (\mathbf{\Sigma}_{-i}^q)^{-1} * (y_{-i} - \mu_{-i}^q) \tag{17}$$

$$\sigma_{i,k}^2 = V_k^q - (\Sigma_{i,k}^q)^t * (\mathbf{\Sigma}_{-i}^q)^{-1} * (\Sigma_{i,k}^q) \tag{18}$$

Enfin :

$$f(y_i|y_{-i}, t_{i,k} = 1, t^q_{-i}, \theta^q) \approx \frac{e^{\frac{-(y_i - m_{i,k})^2}{2\sigma^2_{i,k}}}}{\sigma_{i,k} * \sqrt{2\pi}} \ \forall_{i,k} \tag{19}$$

Définissons :

$$n_k^{q+1} = \sum_{i=1}^{n} t_{i,k}^{q+1} \tag{20}$$

- Estimation (II) :

$$\pi_k^{q+1} = \frac{n_k^{q+1}}{n} \tag{21}$$

La proportion de la classe k est donc la somme des probabilité d'appartenance des points à k divisée par le nombre de points.

- Estimation (III) :

$$\mu_k^{q+1} = \frac{\sum_{i=1}^{n} y_i * t_{i,k}^{q+1}}{n_k^{q+1}} \tag{22}$$

La moyenne de la classe k est donc la somme des probabilité d'appartenance des points à k multiplié par leur valeur, le tout divisé par (équation 37).

- Estimation (IV) :

$$V_k^{q+1} = \frac{\sum_{i=1}^{n} (y_i - \mu_k^{q+1})^2 * t_{i,k}^{q+1}}{n_k^{q+1}} \tag{23}$$

La variance de la classe k est donc la somme des probabilité d'appartenance des points à k multiplié par l'écart au carré entre leur valeur et la moyenne de la classe k, le tout divisé par (équation 37).

- Estimation (V) :

Afin d'estimer b_k^{q+1}, on essai de fiter au mieux le variogramme calculé à partir des $t_{i,k}^{q+1}$:

$$\hat{\gamma}_k(h_m) = \frac{1}{2} \frac{\sum_{i,j:h_{i,j} \approx h_m} t_{i,k}^{q+1} * t_{j,k}^{q+1} * (y_i - y_j)^2}{\sum_{i,j:h_{i,j} \approx h_m} t_{i,k}^{q+1} * t_{j,k}^{q+1}} \ \forall_{0<m\leq nh} \tag{24}$$

Ensuite, on estime b_k^{q+1} en minimisant l'écart entre le variogramme estimé et le théorique :

$$\gamma_k(h_m) = \hat{V}_k(1 - \rho_k(h_m)) = \hat{V}_k(1 - e^{\frac{-h_m}{b_k}}) \tag{25}$$

Pour cela, on utilise la méthode des moindres carrés et on essaye de minimiser :

$$Q_k = \sum_m (\hat{\gamma}_k(h_m) - \gamma_k(h_m))^2 W_{m,k} \tag{26}$$

avec :

$$W_{m,k} = \frac{N_{m,k}}{\gamma_k(h_m)^2} \tag{27}$$

et :

$$N_{m,k} = \sum_{i,j:h_{i,j}\approx h_m} t_{i,k}^{q+1} * t_{j,k}^{q+1} \tag{28}$$

Fonctionnement :

- $\forall_k$:
  - On pose Qmin = ∞
  - Boucler sur $\hat{b}_k$ ($\hat{b}_k$ = 0.02, 0.03, ..., 2)
    - Trouver $\hat{V}_k$ avec $\hat{b}_k$ comme valeur du paramètre.
    - Estimer Q_k.
    - si Q_k est plus petit que Qmin, Qmin = Q_k et $b_k^{q+1} = \hat{b}_k$!

3.1.2 Geo-EM-A

Etant donné que Geo-EM-V demandait de nombreuses inversions de matrice, il s'est révélé lent, nous avons donc décidé de réaliser une méthode de classification approximée minimisant le nombre d'inversions de matrices.

- Estimation (I) :

$$t_{i,k}^{q+1} = \frac{\pi_k^q * f(y_i|y_{-i}, t_{i,k} = 1, t_{-i}^q, \theta^q)}{\sum_l \pi_l^q * f(y_i|y_{-i}, t_{i,l} = 1, t_{-i}^q, \theta^q)} \tag{29}$$

$f(y_i|y_{-i}, t_{i,k} = 1, t_{-i}^q, \theta^q)$ = probabilité que la valeur du point i soit y_i sachant les valeurs des autres points, les t_{-i}^q ... et $t_{i,k}$=1.

Or, ce calcul est compliqué (il demande l'inversion de n matrices de tailles (n-1)*(n-1) pour estimer tous les $t_{i,k}$), c'est pourquoi, nous avons décidé d'approximer cette valeur par :

$$f(y_i|y_{-i}, t_{i,k}, t_{-i}^q, \theta^q) \tag{30}$$

On définit alors

$$\lambda_{i,j} = \begin{cases} \frac{1}{(\Sigma^q)_{i,i}^{-1}} \forall_{i=j}. \\ \\ \frac{-(\Sigma^q)_{i,j}^{-1}}{(\Sigma^q)_{i,i}^{-1}} \ \forall_{i\neq j}. \end{cases} \tag{31}$$

Ainsi :

$$m_{i,k} = \mu_k^q + \sum_{j\neq i} \lambda_{i,j} * (y_j - \mu_j^q) \ \forall_{i,k} \tag{32}$$

$$\mu_j^q = \sum_{k=1}^{K} t_{j,k}^q * \mu_k^q \tag{33}$$

et, pour Geo-EM-A1 :

$$\sigma_{i,k}^2 = \lambda_{i,i} \ \forall_{i,k} \tag{34}$$

Pour Geo-EM-A2 :
En remarquant que l'algorithme avait tendance à surestimer la variance, nous avons cherché une façon de réduire son estimation dans l'estimation des $t_{i,k}$, N. Peyrard a alors proposé :

$$\sigma_{i,k}^2 = t_{i,k} * \lambda_{i,i} \ \forall_{i,k} \tag{35}$$

Enfin :

$$f(y_i|y_{-i}, t_{i,k} = 1, t_{-i}^q, \theta^q) \approx \frac{e^{\frac{-(y_i - m_{i,k})^2}{2\sigma_{i,k}^2}}}{\sigma_{i,k} * \sqrt{2\pi}} \ \forall_{i,k} \tag{36}$$

Définissons :

$$n_k^{q+1} = \sum_{i=1}^{n} t_{i,k}^{q+1} \tag{37}$$

- Estimation (II) : identique à "Geo-EM-V".
- Estimation (III) : identique à "Geo-EM-V".
- Estimation (IV) : identique à "Geo-EM-V".
- Estimation (V) : identique à "Geo-EM-V".

3.1.3 Geo-EM-B

Suite à des analyses des résultats de Geo-EM-A et de Geo-EM-V, nous nous sommes rendus compte qu'un des facteurs rendant ces algorithmes peu efficaces était la surestimation de la variance estimée des classes. J'ai ainsi proposé une modification à l'algorithme :

- Estimation (I) : identique à "Geo-EM-A".
- Estimation (II) : identique à "Geo-EM-V".
- Estimation (III) : identique à "Geo-EM-V".

- Estimation (IV) :

$$V_k^{q+1} = \frac{\sum_{i=1}^{n}(y_i - \mu_i^{q+1})^2 * t_{i,k}^{q+1}}{n_k^{q+1}} \quad (38)$$

$$= \frac{\sum_{i=1}^{n}[(y_i - \mu_k^{q+1}) - (\mu_i^{q+1} - \mu_k^{q+1})]^2 * t_{i,k}^{q+1}}{n_k^{q+1}} \quad (39)$$

$$= (23) - \frac{\sum_{i=1}^{n}(\mu_i^{q+1} - \mu_k^{q+1})[(y_i - \mu_i^{q+1}) + (y_i - \mu_k^{q+1})] * t_{i,k}^{q+1}}{n_k^{q+1}} \quad (40)$$

$$= (23) - \frac{\sum_{i=1}^{n} d_{i,k} * t_{i,k}^{q+1}}{n_k^{q+1}} \quad (41)$$

Ainsi, si $t_{i,k}$ est proche de 0 ou de 1, $\mu_i^{q+1} \approx \mu_k^{q+1}$, donc $d_{i,k} \approx 0$. Dans ce cas là, l'apport du point i n'est pas modifié par rapport à (23). Si $t_{i,k}$ loin de 0 ou de 1 :

• Si $y_i \approx \mu_i$, $d_{i,k} \approx (y_i - \mu_k^{q+1})^2 > 0$, on tend donc à annuler l'apport de tels points pour le calcul de la variance de la classe k (ils ne sont pas assez proches des caractéristiques de la classe k).

• Soit, $y_i \approx \mu_k$ et ainsi $d_{i,k} \approx -(y_i - \mu_i^{q+1})^2 < 0$, on tend donc à renforcer l'apport de tels points pour le calcul de la variance de la classe k (ils sont très proches des caractéristiques de la classe k).

Cela permet donc de réellement prendre en compte des points qui ont de forte chance d'appartenir à une classe, ceux pour lesquels on hésite beaucoup et qui augmentent la variance calculée d'après (23) ne sont donc plus pris en compte dans (38).

- Estimation (V) : identique à "Geo-EM-V".

3.2 Commentaires sur les mathématiques

Après avoir assimilé le modèle géostatistique, il a fallu passer à la création de l'algorithme qui lui est associé, Geo-EM. Pour cette partie, N. Peyrard et D.Allard ont cherché le modèle et l'algorithme avec moi, seul je n'y serais pas parvenu. Mon travail fut plus de l'ordre de la compréhension au début, ensuite seulement j'ai pu donner mon avis et tenter des modifications du modèle, ce qui c'est révélé fructueux.

De plus, étant donné que ce stage donnait suite à la thèse de N. Peyrard, je me suis beaucoup servi de son mémoire afin de comprendre les modèles mathématiques.

Pour comprendre la problématique et les modèles, il a donc fallu que je comprenne ce qu'étaient :

- Les notations ainsi que les conventions mathématiques.
- La segmentation/classification de données spatiales.
- Les modèles existants (mélange, HMRF, géostatistique et Complet) et la façon de les simuler.
- Le fonctionnement des algorithmes existants.
- Un modèle mélange.

Pour cela, je me suis aidé de :

- Mes maîtres de stage qui sont restés très disponibles.
- La thèse de N. Peyrard.
- Statistics for Spatial Data (N. Cressie) qui est un livre sur les statistiques spatiales.
- Internet, recherches Google.

Au départ, suite à des tests et des analyses nous nous sommes rendu compte que Geo-EM donnait de mauvais résultats. J'ai donc analysé les résultats afin de comprendre d'où venaient les erreurs et d'y trouver des solutions. Ensuite, avec N. Peyrard et D. Allard, nous avons organisé des réunions ou chacun a proposé une/des solution(s) pour remédier aux problèmes. Cela explique donc les nombreuses versions de Geo-EM.

Après avoir assimilé les modèles et avoir programmé les algorithmes de simulation et de classification du modèle géostatistique, j'ai fait un certains nombre de tests de cet algorithme afin de l'analyser le mieux possible. Les séries de tests n'ont bien sur pas été faites au hasard, nous en avons décidés avec N. Peyrard et D. Allard puis je me suis chargé de les lancer et de les analyser.

L'analyse de tels résultats m'a permis de comprendre où se trouvaient les failles de l'algorithme Geo-EM, voilà comment je procédais :
- Lancement de simulations avec Geo-EM, SF-EM et EM.
- Analyse des résultats et comparaison des algorithmes de classification.
- Analyse des cas où Geo-EM donne de bons résultat et des cas où il a des difficultés.
- Analyser de quelle façon Geo-EM réagi et pourquoi (avec l'aide de mes tuteurs).
- Proposition de modifications afin d'améliorer les résultats de Geo-EM (avec l'aide de mes tuteurs).
- Programmation des modifications et lancement de nouvelles simulations.

Cela a permis de réaliser sept versions de Geo-EM et de comprendre pourquoi certaines sont plus efficaces que d'autres. Ici aussi j'ai du acquérir une certaine autonomie.

Etudions maintenant les résultats des simulations qui nous ont permis d'analyser les résultats de Geo-EM.

3.3 Les résultats sur simulations

Quand l'algorithme Geo-EM fut terminé, afin de décider quelle(s) version(s) de Geo-EM il fallait garder, nous avons décidé de faire des séries de simulations. Ainsi, nous pouvions comparer la classification des algorithmes par rapport à la classification juste. De plus, on a pu comparer les algorithmes entre eux et comprendre les raisons de leur échecs ainsi que de leurs réussites.

voilà donc des résultats sur des simulations du modèle complet ou géostatistique avec des situations simples à classifier et d'autres où les classes sont plus confondues.

On notera :
- μ_k = moyenne de la classe k.
- V_k = variance de la classe k.
- σ_k = écart type de la classe k. ($V_k = \sigma_k^2$)
- b_k = paramètre géostatistique de la classe k.
- β = coefficient de régularité spatiale, s'il est fort on a des classes denses, sinon elles sont 'éparpillées'.
- Err_k = pourcentage des points qui devraient être dans la classe k et qui n'y sont pas. Par exemple, si la classe k compte 50 points et que l'algorithme en retrouve 60 dans la classe k dont 40 qui sont justes, l'erreur pour cette classe est de 20%. (10 sur 50 mal classés.)
- $MErr$ = erreur moyenne = $\sum_k \pi_k * Err_k$

3.3.1 Comparaisons des algorithmes sur des situations simples

Voici le tableau des résultats sur une carte de 100 points. A chaque fois, les résultats sont acquis sur 100 simulations pour lesquelles les algorithmes effectuent au plus 150 itérations.

De plus, dans chaque simulation, $\mu_1 = 0$, $V_1 = V_2 = 1$, les valeurs des autres paramètres varient d'un jeu de simulations à l'autre.

Les trois première lignes sont des simulations de modèle géostatistique, les trois dernières, des simulations de modèle complets avec $\beta = 0.15$.

				$MErr$						
				Algorithme Geo-EM						
β	b_1	b_2	μ_2	A1	A2	B1	B2	V	SF-EM	EM
0	0.1	0.1	2	27.5%	16%	**15.8%**	17.8%	28.9%	23.1%	20.9%
0	0.1	0.1	3	9.7%	**7.2%**	7.3%	6.9%	14.6%	8.6%	9.6%
0	0.2	0.2	3	10.1%	6.5%	6.7%	**5.9%**	11.5%	10%	8.9%
0.15	0.1	0.1	2	32.6%	**16.3%**	16.4%	17.5%	33.6%	26.2%	22.7%
0.15	0.1	0.1	3	9.6%	7.3%	7.4%	**6.8%**	16.5%	10.4%	8.8%
0.15	0.2	0.2	3	10.3%	6.8%	6.8%	**6%**	14.1%	11.8%	9%

Lorsque $\beta = 0$, seul Geo-EM est un algorithme basé sur le même modèle que celui qui sert à simuler (quand $\beta \neq 0$, aucun). Malgré tout, SF-EM et EM donnent de bons résultats alors qu'ils ne sont pas basés sur ce modèle.

Avec ces résultats, on peut remarquer que bizarrement, Geo-EM-V donne des résultats relativement moins bons que Geo-EM-A1, la raison est resté pour nous relativement floue.

Enfin, on peut voir que Geo-EM-B1 a l'air d'être l'algorithme le plus efficace, on évite ainsi le problème de la surestimation de la variance comme on le vera plus loin.

Dans la suite nous n'étudions que les versions B1 et A2 de Geo-EM car les autres sont moins performantes.

3.3.2 Comparaisons des algorithmes sur des simulations complexes avec 2 classes

Ici, toutes les simulations sont issus d'un modèle géostatistique avec :
$\mu_1 = 0$, $V_1 = V_2 = 1$, $b_1 = b_2 = 0.1$, $a_1 = 1$, 150 itérations, 100 simulations.

Les poids a_k agissent de façon exponentielle, ainsi, on peut estimer les proportions des classes lorsque β=0 :

$a_1 = 1$ et $a_2 = 2$ donne une proportion de 0.27 pour la classe 1 et de 0.73 pour la classe 2.
$a_1 = 1$ et $a_2 = 3$ donne une proportion de 0.13 pour la classe 1 et de 0.87 pour la classe 2.

Algo	a_2	μ_2	$MErr$	Err_1	$\hat{\mu}_1$	$\hat{\sigma}1$	$\hat{b}_1$	Err_2	$\hat{\mu}_2$	$\hat{\sigma}2$	$\hat{b}_2$
SF-EM	2	2	21.3%	18.2%	0.26	1.02	-	22.6%	2.17	0.86	-
SF-EM	2	3	8.9%	11.8%	0.14	1.03	-	7.7%	3.04	0.94	-
SF-EM	3	2	29.8%	16.4%	0.92	1.08	-	31.6%	2.3	0.83	-
SF-EM	3	3	10.6%	13%	0.74	1.16	-	10.2%	3.08	0.89	-
EM	2	2	19.3%	21.7%	0.2	0.95	-	18.1%	2.1	0.88	-
EM	2	3	8.5%	12%	0.06	0.93	-	7%	3	0.92	-
EM	3	2	28%	15.7%	0.85	1.07	-	29.6%	2.2	0.8	-
EM	3	3	10.2%	11.8%	0.7	1.12	-	9.8%	3.05	0.88	-
Geo-EM-A2	2	2	22.1%	7.8%	0.25	0.84	0.08	27%	2.4	0.7	0.06
Geo-EM-A2	2	3	13%	1.4%	0.45	1.07	0.09	17%	3.3	0.74	0.06
Geo-EM-A2	3	2	34.3%	4.2%	0.72	0.8	0.08	38%	2.6	0.63	0.06
Geo-EM-A2	3	3	26.3%	0.9%	1.24	1.1	0.13	29.6%	3.45	0.68	0.06
Geo-EM-B1	2	2	**16.2%**	21.6%	-0.1	0.78	0.06	14.2%	2.1	0.72	0.07
Geo-EM-B1	2	3	**5.4%**	8.5%	0	0.76	0.07	4.2%	3	0.83	0.11
Geo-EM-B1	3	2	**22.3%**	27.5%	0.17	0.88	0.1	21%	2.2	0.75	0.07
Geo-EM-B1	3	3	**6.7%**	16.4%	0.06	0.85	0.13	5.5%	3	0.87	0.09

Sur ces simulations, il est clair que Geo-EM-B1 donne les meilleurs résultat à chaque fois. Il semble être l'algorithme le plus robuste. Il est suivi par EM puis les autres. EM est donc robuste lui aussi, les autres le semblent moins.

On peut observer que Geo-EM-B1 estime une variance plus faible que les autres algorithmes et surtout que Geo-EM-A1, cela tend à vérifier qu'une estimation de la variance trop grande donne de mauvais résultats.

Enfin, on peut voir que Geo-EM-A2 commet une petite erreur pour la classe 1 et une grande pour la classe 2, cela veut dire qu'il surestime la proportion de la classe 1 par rapport à la classe 2 ce qui donne une erreur finale importante. On peut dire cela grâce à la définition de Err_k. Au contraire, Geo-EM-B1 ne fait pas cette erreur, les erreurs des classes sont relativement plus proches, cela grâce à une sous-estimation de la variance.

Vérifions maintenant sur 3 classes les résultats de nos algorithmes.

3.3.3 Comparaisons des algorithmes sur des simulations complexes avec 3 classes

Ici, toutes les simulations sont issus d'un modèle géostatistique avec :
$\mu_1 = 0$, $\mu_2 = 2$, $\mu_3 = 5$, $V_1 = V_2 = V_3 = 1$, $b_1 = b_2 = b_3 = 0.1$, 150 itérations, 100 simulations.

lorsque β=0 :
$a_1 = 1$, $a_2 = 1$ et $a_3 = 1$ donne une proportion de 0.33 pour la classe 1, de 0.33 pour la classe 2 et de 0.33 pour la classe 3.
$a_1 = 1$, $a_2 = 2$ et $a_3 = 2$ donne une proportion de 0.16 pour la classe 1, de 0.42 pour la classe 2

et de 0.42 pour la classe 3.
$a_1 = 1$, $a_2 = 2$ et $a_3 = 3$ donne une proportion de 0.1 pour la classe 1, de 0.24 pour la classe 2 et de 0.66 pour la classe 3.
$a_1 = 2$, $a_2 = 1$ et $a_3 = 2$ donne une proportion de 0.42 pour la classe 1, de 0.16 pour la classe 2 et de 0.42 pour la classe 3.

Algo	a_1,a_2,a_3	$MErr$	Err_1	$\hat{\mu}_1$	$\hat{\sigma}_1$	$\hat{b}_1$	Err_2	$\hat{\mu}_2$	$\hat{\sigma}_2$	$\hat{b}_2$	Err_3	$\hat{\mu}_3$	$\hat{\sigma}_3$	$\hat{b}_3$
SF-EM	1, 1, 1	21.7%	24%	0	0.9	-	31.8%	1.95	0.9	-	8.4%	4.9	0.98	-
SF-EM	1, 2, 2	27%	15.1%	0.5	1.04	-	42.3%	2.6	0.8	-	15.7%	5.12	0.87	-
SF-EM	1, 2, 3	53%	2.6%	1.5	1.4	-	78%	4.3	0.76	-	50%	5.57	0.7	-
SF-EM	2, 1, 2	19.5%	19.8%	-0.13	0.86	-	37%	2.14	0.98	-	11.5%	5.06	0.89	-
EM	1, 1, 1	19.3%	19.7%	-0.1	0.9	-	27.8%	1.97	0.9	-	9%	5	0.94	-
EM	1, 2, 2	25.9%	13.7%	0.6	1.05	-	43.5%	2.4	0.8	-	12.3%	5.1	0.89	-
EM	1, 2, 3	55.5%	0.3%	1.6	1.4	-	90%	4.5	0.6	-	50%	5.5	0.7	-
EM	2, 1, 2	19%	17.9%	-0.1	0.87	-	40%	2.17	0.94	-	11.7%	5.1	0.9	-
Geo-EM-A2	1, 1, 1	**15.4%**	15.3%	-0.2	0.76	0.08	21.9%	2.1	0.7	0.05	7%	5	0.87	0.07
Geo-EM-A2	1, 2, 2	26.4%	4.9%	0.5	0.9	0.12	42%	2.8	0.66	0.07	17.8%	5.3	0.7	0.08
Geo-EM-A2	1, 2, 3	54.2%	0.1%	1.2	1.16	0.05	85%	4.2	0.5	0.05	50.4%	5.8	0.55	0.05
Geo-EM-A2	2, 1, 2	**16.6%**	17.5%	-0.27	0.76	0.07	20.8%	2.2	0.8	0.08	12.5%	5.2	0.8	0.08
Geo-EM-B1	1, 1, 1	16%	18.3%	-0.2	0.66	0.08	18.8%	2.1	0.6	0.05	9.5%	5.1	0.75	0.08
Geo-EM-B1	1, 2, 2	**19%**	19%	0	0.7	0.11	24%	2.3	0.6	0.05	12.8%	5.2	0.7	0.1
Geo-EM-B1	1, 2, 3	**26.5%**	10.2%	0.25	0.8	0.15	40%	3	0.6	0.07	23%	5.4	0.67	0.07
Geo-EM-B1	2, 1, 2	17.8%	22.2%	-0.35	0.7	0.07	24%	2.1	0.6	0.08	11.4%	5.15	0.8	0.09

Sur ces simulations, il est clair que Geo-EM-B1 donne les meilleurs résultats. Cela confirme les conclusions données aux résultats précédents.

On remarque ici aussi que l'estimation de l'écart type des classes est plus faible chez Geo-EM-B1 que pour les autres algorithmes.

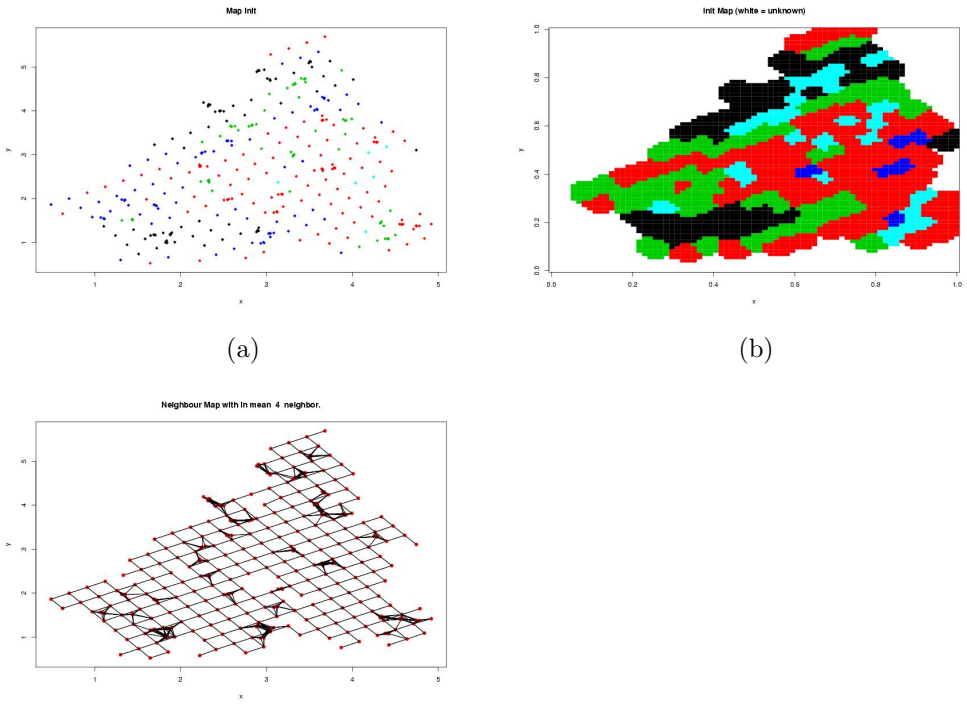
Enfin, on peut voir ici aussi que Geo-EM-A2 commet des erreurs différentes pour chaque classe, il surestime donc encore un classe par rapport aux autres, ce que ne fait pas Geo-EM-B1.

Toutes ces simulations ont donc permis de voir quelles étaient les différences entres les algorithmes, mais aussi de mieux comprendre leur fonctionnement.

3.4 Les résultats sur cas réels

Un des problèmes dans le domaine des géostatistiques est que les données terrain sont chères et donc peu nombreuses.

Ces données ont étés récoltées dans le Jura Suisse sur une région de 14.5 km^2 où on étudie la concentration de sept métaux potentiellement toxiques nommés Cd, Co, Cr, Cu, Ni, Pb et Zn. Les scientifiques ayant effectués ces relevées ont classifié le terrain avec une séparation géologique, argovien et non argovien. Le but de la classification automatique sur les métaux avec nos algorithmes est de trouver une corrélation entre la classification géologique et les métaux.



(a) (b)

(c)

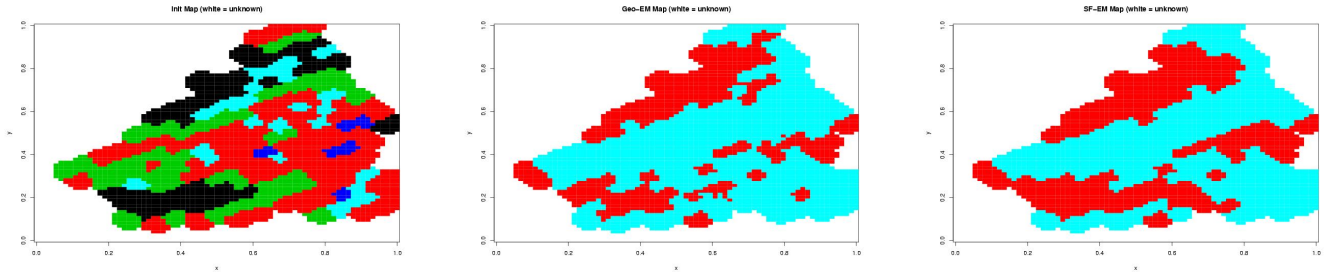
Figure 3: Graphique des données du Jura. (a) indique où se trouve chaque point de mesure et comment les scientifiques l'ont classifié. (l'argovien est en noir) (b) est la même carte que (a) mais avec une coloration des zones sans mesures. Ainsi, en chaque point de la carte où il n'y a aucune mesure, on met la couleur du point de mesure le plus proche, à condition qu'il ne soit pas trop loin. S'il n'y a aucune mesure à proximité, le point reste blanc. (c) est un graphique de voisinage de la carte (il y a une arrête entre deux points s'ils sont voisins).

On peut voir sur la figure 3 la carte Jura avec un graphique de sinage et les 5 classes données par les spécialistes sur le terrain.

Nous avons ensuite essayé de classer les points en fonction de différentes mesures (Co, Cr,...) séparément. On pourra ensuite conclure sur les corrélations entre la classification géologique du sol et les métaux mesurés ici.

La figure 4 est un exemple de mesures avec lesquelles la corrélation entre la classification terrain et celle des algorithmes est forte.

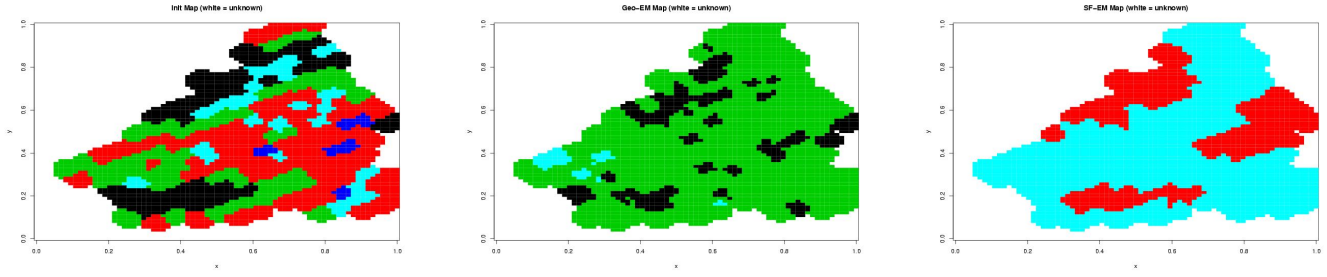
La figure 5 est un exemple de mesures avec lesquelles la corrélation entre la classification terrain et celle des algorithmes est faible.



(a) (b) (c)

Figure 4: Classification automatiques du métal Co. (a) classification géologique, (b) classification par algorithme Geo-EM (c) classification par algorithme SF-EM.

Carte initiale Cr, Geo-EM Cr, SF-EM



(a) (b) (c)

Figure 5: Classification automatiques du métal Cr. (a) classification géologique, (b) classification par algorithme Geo-EM (c) classification par algorithme SF-EM.

On voit donc qu'ici avec l'algorithme Geo-EM, les résultats ne sont pas lisses car il n'y a pas d'estimation de β alors que SF-EM classifie les données de manière plus lisse. Cela montre la nécessité de réaliser l'algorithme CP-EM issu du modèle complet.

3.5 Commentaires sur l'implémentation

Durant tout mon stage, l'implémentation du code s'est principalement faite sous C et R. La principale partie a été l'implémentation des nouveaux algorithmes de simulation et de classification dans le programme existant "nem".

Dans "nem", j'ai programmé :
- La gestion de données irrégulièrement réparties.
- Des tests de la robustesse de SF-EM quand les données ne sont pas sur une grille régulière et sont issues d'un modèle HMRF (ce qui a donné de bons résultats).
- La simulation d'un modèle Géostatistique.
- La simulation d'un modèle Complet.
- La classification par l'algorithme Geo-EM.
- Les 7 versions de Geo-EM.
- De petites modifications nécessaires pour garder un fonctionnement simple du programme.

La simulation d'un modèle géostatistique n'était pas possible avant, il a donc fallu que je modifie le code afin d'avoir le choix de modèle à simuler ainsi que celui de l'algorithme pour estimer la classification. Ensuite, il a fallu rajouter à la fois les fonctions de création de $\mathbf{\Sigma}$ (je l'ai codé) et à la foi les fonctions permettant de donner des valeurs aux points en fonction de $\mathbf{\Sigma}$ (récupération et adaptation de fonctions existantes.).

Pour réaliser cela, je me suis basé sur un code R appelé "rmultnorm" faisant appel à des fonctions que je ne possédais pas sous C. Il a donc fallu que j'assimile l'algorithme de "rmultnorm" et que je retrouve des algorithmes C pour inverser ou diagonaliser une matrice afin de réaliser des simulations de modèles géostatistique.

Une fois le modèle géostatistique codé aussi, le modèle complet étant un mélange des modèles géostatistique et HMRF, il m'a suffit de faire en sorte que dans ce cas là, on passe par les deux étapes pour simuler les classes et les données. Ce ne fut donc pas très complexe.

Pour réaliser toutes ces modifications, il m'a fallu assimiler le code et principalement les 5 fichiers principaux que j'ai du modifier à plusieurs reprises. Ce travail fut plus fastidieux que compliqué du fait que je connaissais déjà les modèles.

Afin de charger des fichiers de données pour que nem puisse les lire et travailler dessus, j'ai programmé "fileCharge" qui, à partir d'un fichier de données comprenant le nombre de points, les coordonnées des points et les valeurs en chaque point crée :
- toto.radius contenant la taille du rayon maximum d'influence du voisinage.
- toto.nei contenant la liste des points et de leurs voisins.
- toto.coord contenant les coordonnées des points.
- toto.dat contenant les données en chaque point.
- toto.str contenant le nombre de points et de données par point.

Pour simuler des données, nem à besoin d'une carte avec un certain nombre de points et leur coordonnées. Pour cela j'ai créé "strauss" (reprit d'un code Fortran) qui est capable de simuler n points dans un espace [0,1]*[0,1] avec un espace minimum entre deux points r.

Un petit récapitulatif du fonctionnement de l'algorithme nem :

- Classification de données réelles :
  - Chargement des données présentes dans un fichier afin de créer plusieurs fichiers nécessaires à nem grâce au programme fileCharge.
  - Choix du modèle le plus approprié aux données et donc de l'algorithme correspondant.
  - Choix des différents paramètres de l'algorithme (critère d'arrêt, nombre de classes...).
  - En sortie plusieurs fichiers indiquant les résultats de l'algorithme (les paramètres de toutes les classes, la classification...).
- Classification de données simulées :
  - Simulation de points dans l'espace grâce à un algorithme Strauss qui permet de mettre une distance minimale entre deux points lorsqu'il simule leurs emplacements.
  - Choix du modèle pour la simulation des données et de la classification initiale.
  - Choix de l'algorithme de classification.
    - Répétition de : Simulation (des données et des classes) puis Estimation. Afin d'avoir des résultats moyens sur de nombreux tests.
  - En sortie plusieurs fichiers indiquant les résultats de l'algorithme (les paramètres de toutes les classes, la classification...).

Il était nécessaire de créer quelques programmes permettant de visionner clairement les résultats des algorithmes. Mon choix c'est tout naturellement porté sur le langage R qui est un langage interprété spécialisé en mathématiques et statistiques (proche de Matlab dans son fonctionnement) et qui permet des analyses rapides et complètes.

De plus, du fait que l'année dernière j'ai entièrement fait mon stage sous R, je connaissais relativement bien les avantages et les inconvénients de ce langage. Cela m'a permis de faire rapidement des petits programmes permettant :

- De tracer un graphe de voisinage pour chaque graphe. Les points étant les sommets et une arrête n'apparaît entre deux points que si leur distance est inférieure au rayon de voisinage (voir figure 6, image a).
- D'afficher les classifications initiales puis estimées par l'algorithme ainsi que de calculer l'erreur (voir figure 6, image b).
- De faire des affichages des distributions des données.
- D'afficher les résultats de nombreuses simulations dans un tableau.
- De calculer le Small world effect d'un graphe (distance moyenne entre deux sommets en termes d'arcs) afin de comparer des graphes de voisinage.
- De dessiner le variogramme théorique à partir de paramètres estimés par l'algorithme de classification et celui estimé à partir des données connues (voir figure 6, image c).
- D'afficher les évolutions des paramètres estimés par un algorithme au fil des itérations (voir figure 6, image d).
- ...

J'ai écrit ces fonctions au moment où j'en avais besoin, certaines ne m'ont pas été très utiles mais la plus part le seront encore après mon stage. Cela m'a aidé à comprendre certaines réactions d'un algorithme ou alors de comparer les réactions de deux algorithmes face à un même problème.

L'avantage de R est aussi que l'on n'est pas obligé d'écrire des programmes pour faire des analyses, il n'y a pas de compilation, il suffit d'entrer une ligne de commandes pour qu'il nous donne une réponse. De plus, ce programme est gratuit et très réparti, cela permet de disposer d'une très grande quantité de fonctions dans tous les domaines imaginables et donc d'avancer relativement rapidement dans son travail.

Ces résultats m'ont à la fois permis de comprendre et de comparer les algorithmes, mais aussi de repérer les défauts d'un algorithme afin de pouvoir le modifier et l'améliorer. C'est cela qui nous a permis avec mes tuteurs de créer toutes les versions de Geo-EM.

3.6 Conclusions et perspectives

De toutes ces expériences, on peut donner un jugement sur tous ces algorithmes :

- EM
  - Rapide, efficace et relativement stable.

- SF-EM
  - Rapide, efficace sur des cas simples, assez stable.

- Geo-EM-V
  - Lent, peu efficace, donne des variances trop élevées.

- Geo-EM-A1
  - Assez rapide, relativement proche de SF-EM, mais moins efficace.
  - Estime relativement bien les paramètres (variance k, moyenne k, b_k).

- Geo-EM-A2
  - Assez rapide, très efficace sur des tests simples.
  - Estime relativement bien les paramètres (variance k, moyenne k, b_k)
  - Ne se montre pas très stable sur des tests difficiles.

- Geo-EM-B1
  - Assez rapide, très efficace.
  - Estime relativement bien les paramètres moyenne k et b_k mais sous estime la variance.
  - Stable et robuste, il résiste relativement bien aux cas difficiles.

Parmi les Geo-EM, on choisira donc le plus fréquemment B1, même s'il n'est pas toujours le plus efficace, il se révèle le plus stable et dans 80% des cas étudiés, il reste tout de même le meilleur.

L'une des raisons de son efficacité est le fait qu'il réduise la variance des classes de façon 'intelligente', ainsi il évite de tomber dans des pièges ou l'algorithme classifie tous les points dans une même classe alors qu'il existe deux classes de points. De plus, dans les cas difficiles (proportions différentes, moyennes proches,...), réduire la variance permet aux classes proches de ne pas trop se mélanger.

Ces algorithmes ont donc donné satisfaction, Geo-EM arrive aussi à de bons résultats, nous pouvons dire que cet algorithme fonctionne bien.

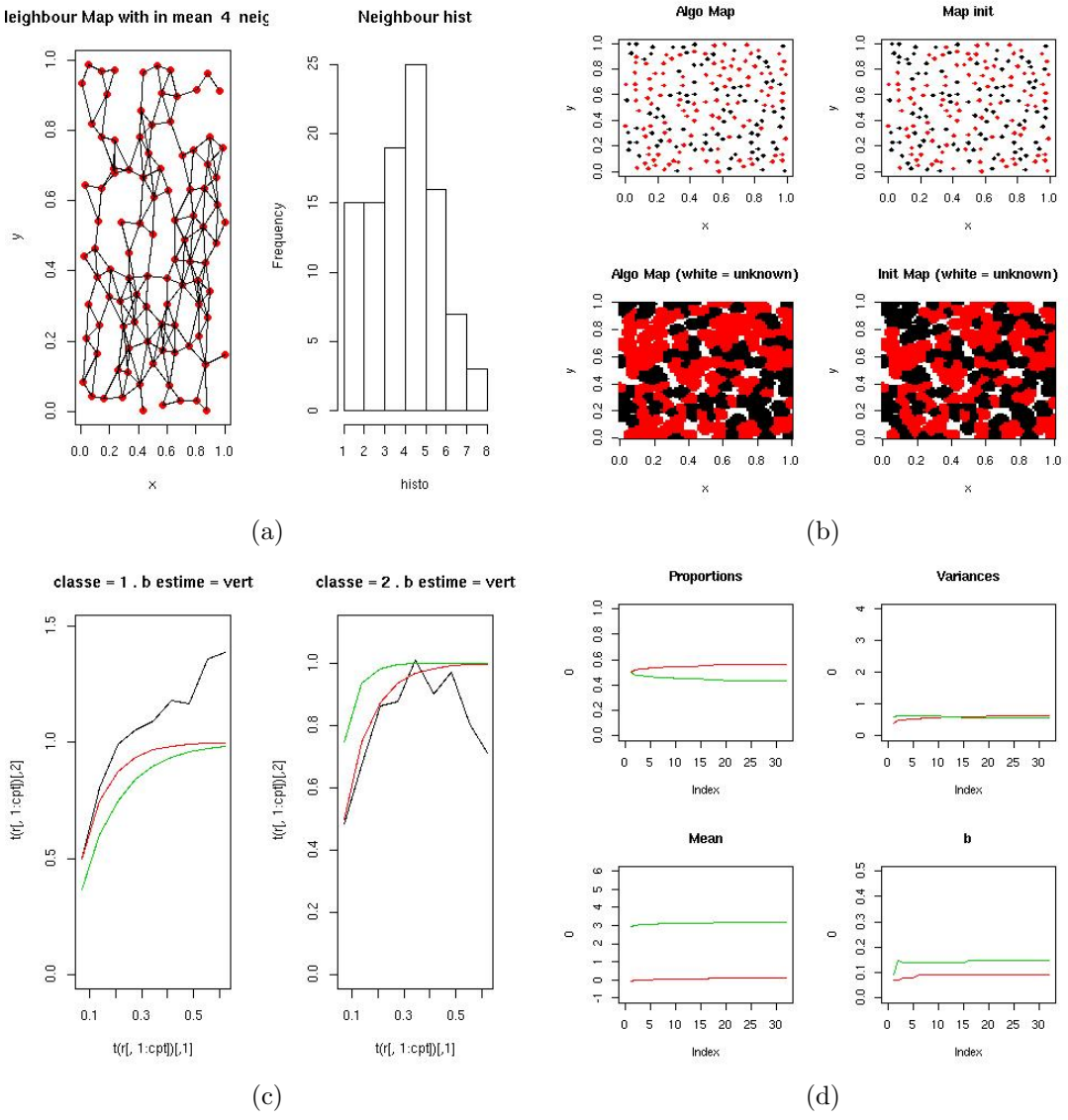


Figure 6: (a) graphe de voisinage et histogramme du nombre de voisins par point, (b) représentation des classifications, (c) représentation des variogrammes, (d) graphe des évolutions des paramètres

Cependant, après analyses complètes des algorithmes, et tout particulièrement avec les tests sur données réelles, il nous est apparu la nécessité de trouver comment classifier les données avec un modèle complet qui combinera les bonnes performances du modèle géostatistique en terme de pourcentage d'erreur et l'aspect plus lisse des classifications issu d'un modèle HMRF.

J'ai commencé à travailler dans cette direction durant la fin de mon stage, un 'algorithme test' a été réalisé et a donné des résultats mais il n'est pas validé et permet juste de préparer le travail à suivre. Toutefois j'ai eu l'occasion de le tester et de remarquer des résultats intéressant qui renforcent l'idée qu'il faut travailler sur l'algorithme CP-EM.

4 Bilan et Perspectives

4.1 Difficultés / facilités d'adaptation

4.1.1 Les mathématiques

Cette partie du travail était la plus importante car elle concernait directement la recherche, le reste ce n'était que des modifications. C'est à ce moment là que j'ai pu effectuer un travail proche de la recherche et que j'ai pu étudier les techniques de mes tuteurs.

Comme on peut le constater, la compréhension des modèles, de leurs algorithmes de simulations et de leurs algorithmes de classification demande certaines connaissances mathématiques. Cette partie de mon stage fut certainement la plus complexe, malgré des connaissances en statistiques et quelques cours en rapport avec ce stage, il fallut que j'acquière d'autre bases. Pour cela, N. Peyrard et D. Allard m'ont bien aidé, ils ont répondu à toutes mes questions de façon claire (il faut dire qu'ils enseignent tous les deux à l'IUP) et m'ont fait quelques "présentations" utiles pour ma compréhension.

Etant donné que j'avais beaucoup moins de connaissances qu'eux, mon défaut fut de facilement demander de l'aide avant d'avancer. Or, petit à petit, j'ai compris que j'etais capable d'assimiler et de réaliser certaines choses tout seul même si cela prenait plus de temps. J'ai ainsi pu apprendre une certaine autonomie que je n'avais pas au début du stage.

On peut donc voir que cette partie de la compréhension du sujet fut fastidieuse et complexe mais nécessaire pour avancer dans le sujet. Malgré tout, je n'ai pas tout assimilé en une semaine, j'ai commencé par les bases, le reste est venu petit à petit.

En ce qui concerne le modèle géostatistique et l'algorithme Geo-EM, j'ai eu beaucoup de travail dans la compréhension mathématique, au début je me suis contenté de suivre les instructions et de coder l'algorithme. Au fur et à mesure des codages et tests, j'ai bien assimilé l'algorithme ce qui m'a permis de proposer des améliorations à l'algorithme.

Cette partie fut donc complète pour moi car elle m'apporta tout d'abord une vision d'apprentissage au métier de la recherche puis une intégration en travaillant moi-même sur le problème.

Après avoir donc acquis toutes les bases mathématiques et algorithmiques, je me suis mis à la lecture du code et je suis donc passé à la phase d'assimilation informatique.

4.1.2 L'informatique

Une fois les bases de l'algorithmique acquises, j'ai pu commencer mon apprentissage informatique, celui là s'est découpé en plusieurs phases :
- Assimilation du code existant (en C)
- Apprentissage de Latex pour faire des documents avec des formules mathématiques.
- Utilisation de R pour les analyses.

Durant sa thèse, N. Peyrard avait déjà reprit un code existant, après elle un étudiant l'a aussi reprit. Je suis donc arrivé comme quatrième scribe sur ce programme qui comporte un dizaine de fichiers. J'ai surtout travaillé sur 4-5 fichiers que j'ai du partiellement voir quasiment entièrement assimiler. Cela ne fut pas excessivement dur car le code était très bien commenté et en haut de chaque fichiers, les utilisateurs avaient fait un listing de leur modification, je repris moi aussi le système afin que le prochain utilisateur puisse aisément reprendre le code existant. Le code était déjà d'une taille importante (il l'est encore plus aujourd'hui), cette partie m'a donc pris du temps mais fut relativement aisée du fait des commentaires réguliers mais en anglais! De plus, le code reprend l'algorithmique que je connaissais déjà, les deux réunis m'ont permis de bien assimiler les méthodes en cours.

A l'INRA et dans tous les laboratoires à tendance mathématiques, tous les rapports, articles... sont faits en Latex car ce logiciel a l'avantage de permettre aisément de faire des formules mathématiques complexes. En arrivant, je ne connaissais Latex que de nom et je donc du tout apprendre à partir de fichier Latex existants et d'une documentation simple mais complète permettant de bien assimiler le programme. De plus, tout au long de mon stage j'ai rédigé des rapports intermédiaires sur les résultats, les choix, les algorithmes... qui m'ont permis de bien connaître Latex avant de commencer mon rapport. Je pense donc que même si cela m'a prit un certain temps, l'apprentissage de Latex m'a été utile et me le serra pour le futur.

L'année dernière, j'ai aussi réalisé mon stage à l'INRA et j'ai entièrement travaillé sous R qui est un langage interprété spécialisé dans les statistiques et graphiques, très utilisé à l'INRA. Afin d'analyser mes résultats ou de représenter graphiquement des cartes, rien n'était fait. J'ai donc fait quelques programmes sous R tout au long du stage, cela fut très utile et servira je le pense aux prochains utilisateurs.

4.2 Apports à la Biométrie

Les chercheurs ont souvent un travail de recherche principal et de nombreuses missions annexes pour lesquelles il leur est difficile de consacrer beaucoup de temps. Parmi ces missions, de nombreuses sont donnés à des stagiaires pour différentes raisons :

- Un stagiaire qualifié est capable de faire un travail d'ingénieur et donc de réaliser des taches relativement complexes.
- Un stage de plus de quatre mois permet de finir une mission assez importante.
- Avoir un stagiaire permet au chercheur d'avancer rapidement sur un sujet.

Ainsi, ma venue au sein du laboratoire Biométrie a demandé du temps de disponibilité à D.Allard et N. Peyrard mais leur a permis de gagner 2-3 mois de travail d'ingénieur (comptons 1 mois de formation par rapport à un ingénieur connaissant le travail). Cela sans avoir la prétention de travailler déjà comme un ingénieur confirmé mais en pensant que c'est le but de ma formation et donc que je tends à m'en rapprocher.

De plus, ma spécialité m'a permis d'acquérir des connaissances en statistiques qui m'ont été utiles durant ce stage, cela m'a donc aidé à me mettre relativement vite au travail. étant donné que la mission fixée a quasiment été remplie et que cela ouvre la porte à une dernière phase : la mise en place de la classification par modèle complet, je peux dire que ce stage a aussi permis de relancer l'intérêt pour un tel sujet. Les résultats de l'algorithme Geo-EM ont donc montré l'intérêt du sujet et aussi celui de continuer dans cette voie là.

Je pense qu'au-delà de la satisfaction de mes maîtres de stage et du laboratoire Biométrie, cet algorithme pourra être utilisé dans le futur pour une meilleur compréhension du problème de classification de données géostatistiques mais aussi servir d'outil algorithmique facile à réutiliser par la suite.

Regroupant tout cela, je pense que j'ai apporté quelque chose à la recherche, même si c'est infime, c'est déjà une satisfaction personnelle. Malgré tout, je reste réaliste et je sais que j'ai travaillé avec des chercheurs mais je n'ai pas fait un réel travail de chercheur du fait que je ne pouvais pas être entièrement autonome et que je n'en ai pas les qualifications (je ne me souviens pas avoir fait de thèse, des rédactions peut-être mais c'est tout!).

4.3 Apports personnels

Ce stage m'a beaucoup apporté, surtout sur mes connaissances mathématiques. Travailler avec des chercheurs en mathématiques m'a obligé à un certain formalisme auquel je n'étais pas habitué à l'IUP où on apprend plutôt le formalisme informatique.

Tant au niveau informatique que mathématiques, un stage de 4 mois permet de réaliser une mission d'une relative importance et de s'immerger pleinement dans un sujet. De plus, cela m'a permis d'apprendre à gérer mon temps ou encore à préparer des réunions de travail ce qui me sera nécessaire dans ma vie professionnelle.

étant donné que durant mon stage j'ai réalisé 2 oraux, cela m'a permis de m'exercer et de recevoir des critiques différentes pour toujours plus progresser. Dans ce sens là, les rapports (intermédiaires et finaux) que j'ai rendu ont toujours étés corrigés ce qui m'a permis de voir mes erreur et de les corriger.

Enfin, j'ai aussi appris à me servir de Latex ce qui me sera certainement utile pour le futur.

Un des autres atouts de ce laboratoire est de réaliser tous les lundi un 'café des sciences' ou soit des gens de Biométrie, soit des extérieurs présentent leurs travaux ce qui permet à tout le monde de s'instruire ainsi que de communiquer l'actualité du laboratoire. J'ai ainsi eu l'occasion de réaliser un café des sciences sur l'algorithme par colonies de fourmis (cours d'initiation à la recherche) ainsi qu'une préparation à ma soutenance de stage fin août.

Le dernier apport est assez hors sujet mais important. Durant un mois, un chercheur tchèque est venu travailler au laboratoire et c'est trouvé dans le même bureau que moi. Cela m'a permis de communiquer en anglais tant au niveau technique que classique et même si je ne peux pas considérer avoir progressé, cela m'a montré l'importance de la langue anglaise. L'anglais est réellement très utilisé et il est très souvent plus facile de trouver un article anglais d'un chercheur français plutôt qu'un article français. L'anglais est le seul langage oral permettant une communication mondiale et je m'en suis encore plus rendu compte.

4.4 Transferts du savoir

Durant ce stage, je peux dire qu'il y a eu un transfert du savoir permanent. J'étais souvent le receveur, du moins au début, ensuite cela s'est équilibré car je communiquais de façon relativement régulière mes résultats à mes encadrant.

Le transfert du savoir s'est donc effectué grâce à plusieurs supports, le premier est bien entendu ce rapport dont un exemplaire sera conservé par mes encadrant. Ensuite les codes sources ont étaient abondamment commentés et une documentation succincte réalisée, cela pour permettre une maintenance et une évolutivité aisée. De plus, la communication permanente et les nombreux rapports intermédiaires ont permis un transfert du savoir aisé et complet.

Enfin, un deuxième rapport a été rédigé. Ce rapport est beaucoup plus technique et servira de trace complète des analyses effectuées pendant le stage. Ce rapport est beaucoup moins travaillé sur la forme mais reste complet et regroupe tous les rapports intermédiaires en plus d'un guide utilisateur, d'un guide programmeur et d'une liste de mes travaux durant mon stage.

Pour finir, durant le dernière semaine de mon stage, N.Peyrard et D.Allard ont essayé le programme eux même afin de comprendre les modifications et de pouvoir aisément utiliser dans le futur ce programme en mon absence. Cela a été l'occasion de dernières mises au point même si toutes les modifications apportées allaient dans le sens de l'existant du programme.

4.5 Perspectives

Suite à ce stage, on peut dégager deux principaux travaux à effectuer.

Le travail principal et entrant le plus dans le domaine de la recherche est de réaliser, valider puis tester l'algorithme CP-EM. On a vu que cet algorithme semble nécessaire, de plus une piste a déjà était exploré et nous a montré qu'il ne fallait pas s'arrêter là. Ce sera donc la suite logique à mon stage.

Un autre travail à effectuer est la modification de l'algorithme Geo-EM afin qu'il puisse classifier des points sur lesquels plusieurs données sont relevées. IL faut savoir que lors de données réelles, il y a souvent plusieurs données sur un même point et ces données réunies donnent une information relativement plus complète pour la classification. Il serait donc intéressant de se pencher sur ce problème.

Un dernier travail concernerait plus la forme que le fond du programme, il s'agie de modifier le programme afin qu'il fonctionne par menus et non plus par arguments comme c'est le cas actuellement. Cela le rendrait beaucoup plus conviviale et permettrait sa diffusion.

Annexes

Semainier

Semaines 1 et 2 :
- Présentation des locaux, du personnel et du matériel.
- Introduction aux géostatistiques et à la segmentation d'images
- Présentation de l'algorithme de segmentation markovienne d'images.
- Assimilation du code.

Semaines 3 et 4 :
- Assimilation du code et de l'algorithme.
- Codage de la gestion de données irrégulièrement réparties.

Semaines 5 et 6 :
- Assimilation des trois modèles probabilistes(voir partie 2.1).
- Codage de la gestion de la simulation des trois modèles.
- Codage de routines sous R afin de faire quelques vérifications sur les simulations.

Semaines 7 à 10 :
- Création de l'algorithme Geo-EM.
- Implémentation de l'algorithme Geo-EM.
- Débuggage.
- Mise en place de modifications afin d'améliorer l'algorithme.

Semaines 11 et 12 :
- Mise en place de batteries de tests.
- Débuggage.
- Mise en place de simulations afin de comparer les algorithmes.
- Comparaison des algorithmes sur cas réels.
- Analyses des résultats grâce à la réalisation de petits programmes R.

Semaines 13 et 14 :
- Commentaire du code.
- Rédaction du rapport technique ainsi que du guide utilisateur et d'un guide programmeur.
- Rédaction du rapport de stage.

Semaines 15 :
- Préparation à l'oral (27/08)
- relecture et retouches du rapport de stage avec les maîtres de stages.

Semaines 16 à 18 :
- Préparations finales à l'oral.
- Recherches concernant l'algorithme complet.
- Finissions des rapports et envoi du rapport de stage.

Bibliographie

• T. Oetiker, H. Partl, I. Hyna et E. Schlegl (1999). Une courte introduction à Latex.

• H. Garreta (1992). Le Langage C.

• N. Peyrard (2001). Approximations de type champ moyen des modèles de champ de Markov pour la segmentation de données spatiales.

• N. Cressie (1993). Statistics for Spatial Data.

• D. Allard et G. Guillot (1999). Clustering Geostatistical Data.

• O. Atteia, J.-P. dubois et R. Webster (1993). Geostatistical Analysis of soil contamination in the Swiss Jura.

• G. Govaert et C. Ambroise (2004). Analyses de Données et Data Mining.