Sujet de Travaux Dirigés / Pratiques
TP MRF - IMA203

Introduction aux champs de Markov
pour le traitement de l'image

Objectifs de la séance :

Le but de cette séance est de programmer un échantillonneur de Gibbs et de l'étudier dans le cas d'un champ binaire. On utilise ensuite ce modèle pour faire de la classification d'image dans un cadre bayésien.

Les programmes sont écrits sous Matlab ou Python. L'ossature des programmes est donnée et ils doivent être complétés.

Le compte-rendu est à rendre pour le 20 décembre (format papier) lors du cours. Il peut être rédigé en binôme. Les données et programme sont accessibles sur le site : `http://perso.telecom-paristech.fr/~tupin/cours/IMA203/TPMARKOV`

Créez un répertoire pour ce TP dans lequel vous recopierez les programmes donnés. Le TP peut s'effectuer sous Matlab (lancer `Matlab`) ou avec Spyder (lancer `/cal/softs/anaconda/anaconda3/bin/spyder/`). Sous Spyder, pour que les figures s'affichent dans un graphique séparé, cliquer sur la clé à molette du menu, choisir Console IPython, puis Graphiques et sélectionner Sortie Automatique. Ensuite dans le Menu Consoles, exécuter "Redémarrer le noyau".

N.B : Les programmes `tp_part1` *et* `tp_part2` *vous donnent l'ossature pour chaque partie du TP. Mais ils sont INCOMPLETS. Vous ne pouvez pas les lancer directement, mais vous pouvez vous en inspirer pour réaliser le TP.*

1 Etude du modèle d'Ising

Pour cette partie, vous devez récupérer les programmes `tp_part1` et `echan` dans votre répertoire.

Dans cette section on considère un champ markovien binaire (valeurs dans $E = \{0, 1\}$). Le voisinage est défini par la 4-connexité et le potentiel d'une clique d'ordre 2 est défini par $U_c(0, 1) = U_c(1, 0) = +\beta$ et $U_c(1, 1) = U_c(0, 0) = 0$ (le potentiel des cliques singleton est nul).

1. Ecrivez la forme de l'énergie globale puis calculez la pour les deux images suivantes en fonction de β :

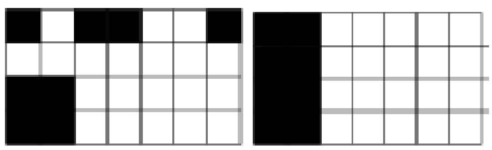

FIGURE 1 – Image 1 (à gauche) et image 2 (à droite)

2. Ecrivez la forme de l'énergie conditionnelle locale d'un site puis calculez la pour les deux configurations suivantes (en 4-connexité), ainsi que les probabilités conditionnelles locales pour les étiquettes 0 et 1. Donnez la valeur la plus probable pour le pixel central.

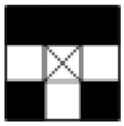

FIGURE 2 – Voisinage local 1. Le pixel à considérer est le pixel central indiqué par une croix.

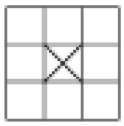

FIGURE 3 – Voisinage local 2. Le pixel à considérer est le pixel central indiqué par une croix.

3. Programmer l'échantillonneur de Gibbs pour ce modèle en modifiant le programme `echan.m` ou `echan.py` qui est appelé par `tp_part1.m` ou `tp_part1.py`. Pour cela, recopiez ces programmes dans votre répertoire et vérifiez que vous êtes dans le bon répertoire sous Spyder ou Matlab. Sous Spyder, il faut cliquer dans l'image pour qu'il exécute l'itération suivante.
4. Faites tourner le programme plusieurs fois. Obtenez vous toujours la même image ? Commentez.

5. Faites varier β de 0.5 à 20. Commentez les résultats.

6. Quelle est l'image qui minimise globalement l'énergie pour ce modèle ?

7. Changez β et donnez lui une valeur négative. Décrivez le résultat et justifiez le.

8. On travaille maintenant en 8 connexité, mais toujours avec des cliques d'ordre 2 (non isotropes cette fois). Pour chacune des images suivantes, proposez les potentiels des cliques qui permettent d'obtenir ces réalisations. Au départ tous les potentiels de clique sont nuls.

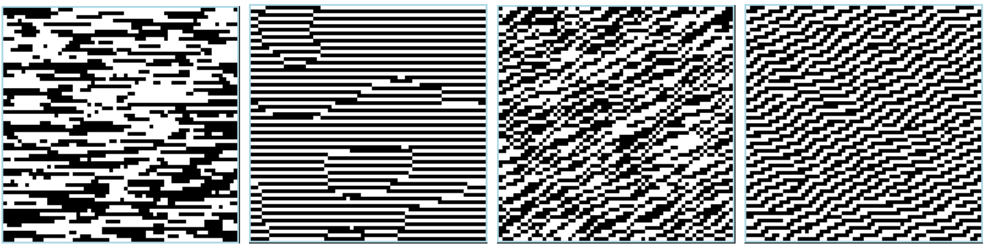

FIGURE 4 – Image A, B, C, D (de gauche à droite)

— Image A : il y a un seul potentiel de clique d'ordre 2 qui vaut -1. Indiquez lequel.

— Image B : en plus du précédent, il y a un potentiel de clique d'ordre 2 qui vaut 1. Indiquez lequel.

— Image C : en plus des 2 précédents, il y a un potentiel de clique d'ordre 2 qui vaut -1. Indiquez lequel.

— Image D : en plus des 3 précédents, il y a un potentiel de clique d'ordre 2 qui vaut +1. Indiquez lequel.

Bonus : programmez un modèle d'Ising avec un potentiel attractif en diagonale uniquement. Commentez le résultat.

2 Classification binaire d'une image

Pour cette partie vous devez récupérer les programmes `tp_part2` et les images `IoriginaleBW.png` (image des classes, non observée en pratique, réalisation x du champ X des classes que vous allez chercher à estimer)) et `Iobservee.png` (image des observations, en niveau de gris, réalisation y du champ Y que vous utilisez pour estimer x).

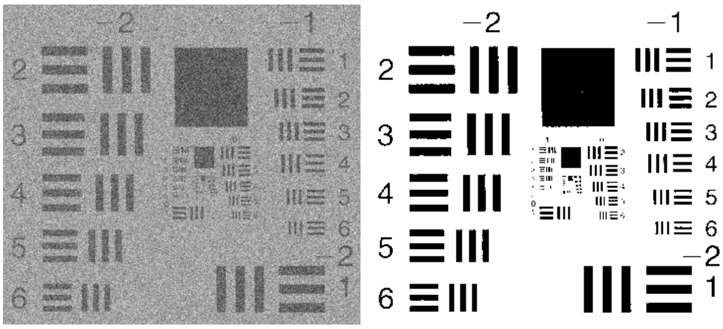

FIGURE 5 – Image observée y à gauche (en niveaux de gris) et image binaire "idéale" x à droite (image originale de classes qu'on cherche à retrouver)

Vous disposez d'une image idéale des classes `IoriginaleBW.png` et de sa version observée `Iobservee.png`. L'objectif est de réaliser une classification en deux classes de cette image observée (idéalement on souhaite retrouver l'image originale idéale). On note x_s la classe du pixel s (que l'on cherche), et y_s le niveau de gris observé. L'objectif est d'utiliser un modèle global sur le champ aléatoire X pour classer l'image. Comme nous l'avons vu en cours cela revient à minimiser l'énergie suivante :

$$U(x|y) = \sum_s -ln(P(Y_s = y_s|X_s = x_s)) + \sum_c V_c(x_s, s \in c)$$

2.1 Analyse des distributions des niveaux de gris

Dans cette partie, on fait l'apprentissage des probabilités $P(Y_s = y_s|X_s)$, c'est à dire de $P(Y_s = y_s|X_s = 0)$ et $P(Y_s = y_s|X_s = 1)$. Cela revient à étudier l'histogramme des niveaux de gris de pixels qui sont dans la classe 0 et de pixels qui sont dans la classe 1.

— Pour réaliser cet apprentissage, il faut sélectionner des pixels appartenant à la classe 0 d'une part (zone sombre de l'image observée), et des pixels appartenant à la classe 1 d'autre part (zone claire de l'image observée). Quelles sont les distributions suivies par les niveaux de gris dans ces deux classes ? Donnez les moyennes et variances des deux classes. *On pourra sélectionner des pixels manuellement, en utilisant la commande `v0=I(i1:i2,j1:j2)` sous Matlab et `v0=I[i1:i2,j1:j2]` sous Python qui met dans un vecteur toutes les valeurs des pixels de l'image I compris entre les indices $i1$ et $i2$ (lignes), et $j1$ et $j2$ (colonnes).*

Dans la suite, on supposera les variances égales.

— Supposons qu'on n'utilise pas de modèle markovien sur X et qu'on classe un pixel seulement en fonction de son niveau de gris en comparant $P(Y_s = y_s|X_s = 0)$ et $P(Y_s = y_s|X_s = 1)$. Montrez que cela revient à seuiller l'image et donnez la valeur du seuil optimal en fonction des paramètres trouvés précédemment (on dit qu'on fait une classification par maximum de vraisemblance ponctuel).

— A partir des résultats trouvés pour $P(Y_s = y_s|X_s)$, écrivez l'énergie d'attache aux données $U_{attdo} = \sum_s -ln(P(Y_s = y_s|X_s = x_s))$.

2.2 Modèle d'Ising pour la régularisation

Pour améliorer les résultats du seuillage, il est nécessaire d'introduire une régularisation (modèle a priori global).

— Soit la fonction $1_{x_s \neq x_t} = 0$ si $x_s = x_t$, et $1_{x_s \neq x_t} = 1$ sinon. Ecrire le potentiel des cliques d'ordre deux pour ce modèle d'Ising en fonction de $\Delta(x_s, x_t)$ où x_s et x_t sont les classes des pixels s et t voisins en 4-connexité et du paramètre de régularisation β. Ce modèle vaudra 0 quand les deux pixels voisins sont égaux et $+\beta$ sinon.

— Ecrire l'*énergie globale* de tout le champ et l'*énergie conditionnelle locale* pour un site s en utilisant les résultats établis précédemment pour l'énergie d'attache aux données et pour l'énergie de régularisation.

— Ecrire les énergies conditionnelles locales pour les classes 0 et 1 du pixel central, en utilisant la configuration du voisinage local 1 précédent (p.2, voisins dans les états 0, 1, 1, 1) en supposant que le niveau de gris du pixel est $y_s = 105$, et en utilisant les valeurs de moyennes et variance trouvées précédemment.

— Dans quelle classe sera mis ce pixel si on lui attribue la classe qui minimise localement l'énergie ?

— Si on considère l'énergie globale du champ, quelle est la solution x quand β vaut 0 ?

Si on considère l'énergie globale du champ, quelle est la solution x quand β vaut $+\infty$?

— Comment varie la solution quand β augmente ? Commentez sur l'intérêt de ce modèle markovien.

2.3 Optimisation par algorithme ICM

On va réaliser l'optimisation de l'énergie globale précédemment définie, en utilisant l'algorithme ICM (Iterated Conditional Mode) qui consiste à partir d'une bonne initialisation des classes, à minimiser l'énergie conditionnelle locale des pixels les uns à la suite des autres. Cet algorithme converge vers un minimum local mais il est très rapide. Modifiez la fonction `echan.m` ou `echan.py` pour programmer l'ICM, en prenant en compte le terme d'attache aux données que vous avez appris (NB il faut cette fois-ci tenir compte de l'image observée en niveau de gris et rajouter le terme d'attache aux données dans l'énergie).

— Que proposez vous pour avoir une bonne initialisation de la solution ? Justifiez votre réponse.

— Avec quelle valeur de β obtenez vous une bonne solution ? (c'est à dire la plus proche de l'image "idéale" IoriginaleBW.png donnée). Comparez ce résultat avec le résultat du seuillage optimal.

— Essayez avec d'autres initialisations (avec une image constante, avec une image aléatoire). Commentez leur influence.

— Programmez le recuit simulé et comparez avec les solutions précédentes.