

Segmentation par champ de Markov (2/2)

Rappels du cours précédent

La segmentation par classification d'une image \mathbf{x} , qui est une réalisation d'un champ aléatoire \mathbf{X} , peut être effectuée par le maximum de vraisemblance, mais les résultats sont relativement décevants, même pour une image de synthèse facile à segmenter « à la main ». Une piste d'amélioration est celle de la classification par le maximum a posteriori (MAP), à condition d'introduire un a priori *global* $p(\mathbf{K} = \mathbf{k})$ sur le champ aléatoire \mathbf{K} , sous la forme d'un *couplage* entre sites voisins. La configuration recherchée est celle qui maximise le MAP :

$$\hat{\mathbf{k}} = \arg \max_{\mathbf{k} \in \Omega} \{p(\mathbf{X} = \mathbf{x} | \mathbf{K} = \mathbf{k}) p(\mathbf{K} = \mathbf{k})\} \quad (1)$$

Le champ aléatoire discret \mathbf{K} doit vérifier la *propriété markovienne*, caractéristique des champs de Markov.

Propriété markovienne

Un champ aléatoire \mathbf{K} est un champ de Markov si la probabilité conditionnelle locale en un site s n'est fonction que de la configuration locale sur le voisinage $\mathcal{V}(s)$ de s , c'est-à-dire si :

$$p(K_s = k_s | K_t = k_t, t \in \mathcal{S} \setminus \{s\}) = p(K_s = k_s | K_t = k_t, t \in \mathcal{V}(s)) \quad (2)$$

Cette *propriété markovienne* nécessite bien sûr de définir le *système de voisinage* entre sites, qui doit être symétrique pour que le graphe modélisant les interactions entre variables aléatoires soit non orienté. Pour les pixels d'une image (cf. figure 1), nous choisissons le système de voisinage des « 8 plus proches voisins » (8 ppv).

Bien sûr, en définissant le voisinage d'un pixel s par $\mathcal{V}(s) = \mathcal{S} \setminus \{s\}$, c'est-à-dire en considérant que deux pixels quelconques de l'image sont voisins, il est clair que tout champ aléatoire vérifie la propriété markovienne, mais cette propriété n'a d'intérêt que si le voisinage est *à courte portée*, comme cela est le cas des 8 ppv. Nous verrons en effet qu'un voisinage à longue portée risque de beaucoup trop ralentir les calculs.

Système de cliques

Du système de voisinage choisi, nous déduisons un *système de cliques*. On appelle *clique* un sous-ensemble de \mathcal{S} dont tous les éléments sont voisins entre eux. L'ensemble des cliques de cardinal i est noté \mathcal{C}_i . Pour le système de voisinage des 8 ppv, il existe quatre ensembles de cliques : \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 et \mathcal{C}_4 (cf. figure 1).

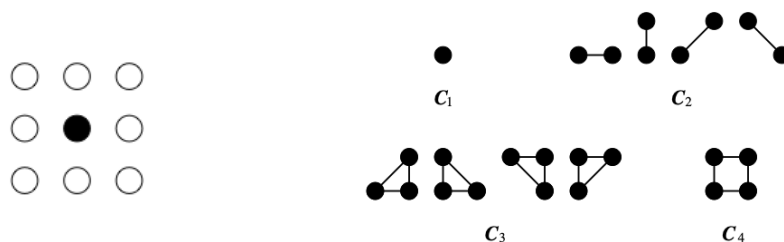


FIGURE 1 – À gauche : système de voisinage des 8 ppv. À droite : ensembles de cliques qui en découlent.

Champ de Gibbs

Le champ aléatoire \mathbf{K} est un *champ de Gibbs* si la probabilité d'une configuration \mathbf{k} s'écrit :

$$p(\mathbf{K} = \mathbf{k}) = \frac{1}{Z} \exp \{-U(\mathbf{k})\} = \frac{1}{Z} \exp \left\{ - \sum_{c \in \mathcal{U}_i \mathcal{C}_i} U_c(\mathbf{k}_c) \right\} \quad (3)$$

où $U(\mathbf{k})$ est l'énergie de la configuration \mathbf{k} , et $U_c(\mathbf{k}_c)$ est l'énergie locale de la clique c , qui ne dépend que de la configuration locale \mathbf{k}_c . Une configuration \mathbf{k} est donc d'autant plus probable que son énergie est plus faible.

Dans l'expression (3), le rôle du dénominateur Z , que l'on appelle la *fonction de partition*, est de rendre $p(\mathbf{K} = \mathbf{k})$ sommable à 1. De (3), nous tirons facilement l'expression de Z :

$$Z = \sum_{\mathbf{k} \in \Omega} \exp \{-U(\mathbf{k})\} \quad (4)$$

Il est généralement inutile de calculer Z . Heureusement, car cela serait quasi-impossible, en pratique, à cause de la taille de l'ensemble Ω . À titre d'exemple, pour une image de taille 100×100 comportant $N = 4$ segments, le nombre de configurations est égal à 4^{100^2} .

Un résultat essentiel, qui permet de simplifier la manipulation des champs de Markov, est le *théorème de Hammersley-Clifford* (1951), qui s'énonce ainsi :

Si $\forall \mathbf{k} \in \Omega$, $p(\mathbf{K} = \mathbf{k}) > 0$, c'est-à-dire si la probabilité de toute configuration \mathbf{k} est non nulle, alors un champ de Markov \mathbf{K} peut s'écrire sous la forme d'un champ de Gibbs (et réciproquement).

La condition $p(\mathbf{K} = \mathbf{k}) > 0$ de ce théorème vient de ce que l'expression $\exp \{-U(\mathbf{k})\}$ n'est jamais nulle.

Modèle de Potts

Le *modèle de Potts* (1952), qui constitue une généralisation du modèle d'Ising (1924), vise à pénaliser les changements de classe entre paires de sites voisins (seules les cliques de \mathcal{C}_2 sont impliquées). Ce modèle consiste donc à réécrire l'a priori du problème (1) sous la forme suivante :

$$p(\mathbf{K} = \mathbf{k}) \propto \exp \left\{ -\beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \right\} \quad (5)$$

où $\beta > 0$ est un paramètre réel, et où le *symbole de Kronecker* $\delta(k_s, k_t)$ est égal à 1 si $k_s = k_t$, et à 0 sinon.

Par ailleurs, nous avons déjà vu que grâce à l'hypothèse d'indépendance des variables aléatoires, la vraisemblance du problème (1) s'écrivait sous la forme d'un produit :

$$p(\mathbf{X} = \mathbf{x} | \mathbf{K} = \mathbf{k}) = \prod_{s \in \mathcal{S}} \frac{1}{\sigma_{k_s} \sqrt{2\pi}} \exp \left\{ -\frac{(x_s - \mu_{k_s})^2}{2\sigma_{k_s}^2} \right\} \quad (6)$$

Vu que $1/\sigma_{k_s} = \exp\{-\ln \sigma_{k_s}\}$, nous tirons de (1), (5) et (6) l'expression suivante de la probabilité a posteriori :

$$p(\mathbf{K} = \mathbf{k} | \mathbf{X} = \mathbf{x}) \propto \exp \left\{ - \sum_{s \in \mathcal{S}} \left[\ln \sigma_{k_s} + \frac{(x_s - \mu_{k_s})^2}{2\sigma_{k_s}^2} \right] - \beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \right\} \quad (7)$$

La probabilité a posteriori du champ aléatoire \mathbf{K} s'écrit donc sous la forme (3), avec les énergies locales suivantes :

$$\begin{cases} U_{c=\{s\}}(k_s) = \ln \sigma_{k_s} + \frac{(x_s - \mu_{k_s})^2}{2\sigma_{k_s}^2} \\ U_{c=\{s,t\}}(k_s, k_t) = \beta [1 - \delta(k_s, k_t)] \end{cases} \quad (8)$$

ce qui montre que \mathbf{K} est un champ de Gibbs, donc un champ de Markov.

Classification par le maximum a posteriori

Maximiser la probabilité a posteriori (7) revient à minimiser l'énergie suivante :

$$U(\mathbf{k}) = \sum_{s \in \mathcal{S}} \left[\ln \sigma_{k_s} + \frac{(x_s - \mu_{k_s})^2}{2 \sigma_{k_s}^2} \right] + \beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \quad (9)$$

Dans cette expression, la première somme est appelée *attache aux données*, la deuxième somme est appelée *régularisation*, et $\beta > 0$ permet de pondérer l'une par rapport à l'autre.

Contrairement à ce que nous avons fait en classant les pixels indépendamment les uns des autres par le maximum de vraisemblance, il est impossible de faire une recherche exhaustive du minimum de $U(\mathbf{k})$. En effet, pour effectuer la classification d'une image de taille 100×100 comportant $N = 4$ classes, le nombre de calculs à effectuer passe de $4 \cdot 10^4$ à 4^{10^4} . Il nous faut donc suivre une autre démarche.

Minimisation de $U(\mathbf{k})$ par recuit simulé

Le *recuit simulé* est une « méta-heuristique » qui permet de trouver l'optimum d'une fonction (minimum ou maximum) de façon stochastique (non déterministe). Il s'agit d'un algorithme itératif. D'autres méta-heuristiques existent, qui sont souvent inspirées de phénomènes naturels (le recuit simulé s'inspire de techniques de métallurgie) : colonies de fourmis, essaims particulaires, algorithmes génétiques, etc.

Le recuit simulé fait décroître un paramètre T , appelé *température*, en le multipliant par $\alpha < 1$ à chaque itération. L'algorithme complet s'écrit :

1. **Initialisations** : $T \leftarrow T_0$; $\mathbf{K} \leftarrow$ Configuration \mathbf{k} obtenue par maximisation de la vraisemblance.
2. **Parcours de tous les pixels s de l'image, visitée ligne par ligne et colonne par colonne** :

- Tirer une nouvelle réalisation $k'_s \in \{1, \dots, N\} \setminus \{k_s\}$ de K_s , et comparer les deux énergies locales :

$$\begin{cases} U_s = \ln \sigma_{k_s} + \frac{(x_s - \mu_{k_s})^2}{2 \sigma_{k_s}^2} + \beta \sum_{t \in \mathcal{V}(s)} [1 - \delta(k_s, k_t)] \\ U'_s = \ln \sigma_{k'_s} + \frac{(x_s - \mu_{k'_s})^2}{2 \sigma_{k'_s}^2} + \beta \sum_{t \in \mathcal{V}(s)} [1 - \delta(k'_s, k_t)] \end{cases} \quad (10)$$

- Si $U'_s < U_s$, alors $K_s \leftarrow k'_s$. Sinon, la nouvelle réalisation k'_s de K_s peut quand même être acceptée, mais avec une probabilité $\exp \left\{ -\frac{U'_s - U_s}{T} \right\}$ qui décroît avec la température T . Une particularité du recuit simulé est donc de ne pas systématiquement éliminer les changements de configuration qui font croître l'énergie.

3. **Mises à jour** : $T \leftarrow \alpha T$, puis retour en 2, tant que le nombre maximal d'itérations n'est pas atteint.

La probabilité $\exp \left\{ -\frac{U'_s - U_s}{T} \right\}$ montre que tout changement de configuration est accepté à haute température. Au contraire, lorsque $T \rightarrow 0$, seuls les changements de configuration faisant décroître l'énergie sont acceptés. L'idée de cette méta-heuristique, qui s'inspire de techniques de refroidissement utilisées en métallurgie pour éviter la formation de bulles d'air, est de commencer par explorer la totalité de l'espace des configurations Ω , puis de progressivement affiner la recherche d'un minimum global.

L'algorithme du recuit simulé montre l'intérêt qu'il y a à utiliser un couplage à courte portée entre pixels voisins. En effet, plus le voisinage est grand, plus le temps de calcul des deux énergies locales (10) est élevé.

Cet algorithme est facile à mettre en œuvre, mais il nécessite d'ajuster plusieurs paramètres : la température initiale T_0 , la raison α de la suite de valeurs de T , et le nombre maximal d'itérations q_{\max} , sans oublier le paramètre β du modèle, qui doit lui aussi être ajusté, de manière à obtenir le meilleur pourcentage de bonnes classifications (cf. TP3).