

## TP3 – Classification bayésienne

### Rappels de cours

La segmentation d'une image en niveaux de gris  $\mathbf{x} = (x_s)_{s \in \mathcal{S}}$  peut être effectuée par *classification*. En choisissant un nombre  $N$  de classes, supposées gaussiennes, et en supposant connues les moyennes  $\mu_1, \dots, \mu_N$  et les écarts-types  $\sigma_1, \dots, \sigma_N$  des différentes classes, le résultat est la configuration  $\hat{\mathbf{k}} = (\hat{k}_s)_{s \in \mathcal{S}}$  qui maximise la *probabilité a posteriori* de la configuration  $\mathbf{k} = (k_s)_{s \in \mathcal{S}}$ , sachant  $\mathbf{x}$ . Or, d'après le théorème de Bayes :

$$p(\mathbf{k}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{k}) p(\mathbf{k})}{p(\mathbf{x})} \propto p(\mathbf{x}|\mathbf{k}) p(\mathbf{k}) \quad (1)$$

L'hypothèse d'indépendance des données permet d'écrire la *vraisemblance* sous la forme d'un produit :

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

Quant à la *probabilité a priori* de la configuration  $\mathbf{k}$ , elle est donnée par le *modèle de Potts* :

$$p(\mathbf{k}) \propto \exp \left\{ -\beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \right\} \quad (3)$$

où  $\mathcal{C}_2$  désigne l'ensemble des paires  $\{s, t\}$  de pixels voisins, défini par le « système de voisinage des 8 plus proches voisins ». De (1), (2) et (3), il vient :

$$p(\mathbf{k}|\mathbf{x}) \propto \exp \left\{ -\sum_{s \in \mathcal{S}} \frac{1}{2} \left[ \ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right] - \beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \right\} \quad (4)$$

Chercher le maximum de  $p(\mathbf{k}|\mathbf{x})$  équivaut donc à chercher le minimum de l'*énergie*  $U(\mathbf{k})$  :

$$U(\mathbf{k}) = \sum_{s \in \mathcal{S}} \frac{1}{2} \left[ \ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right] + \beta \sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)] \quad (5)$$

Étant donné qu'il est impossible, en pratique, de tester  $N^{\text{card}(\mathcal{S})}$  configurations  $\mathbf{k}$ , il faut recourir à une *méta-heuristique*, en l'occurrence le *recuit simulé*, qui fait décroître un paramètre  $T$  appelé *température*, à chaque itération. L'algorithme complet s'écrit :

1. **Initialisations** :  $T \leftarrow T_0$  ;  $\mathbf{k} \leftarrow$  Configuration 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 valeur  $k'_s \in \{1, \dots, N\} \setminus \{k_s\}$ , et comparer les deux énergies locales suivantes :

$$\begin{cases} U_s = \frac{1}{2} \left[ \ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right] + \beta \sum_{t \in \mathcal{V}(s)} [1 - \delta(k_s, k_t)] \\ U'_s = \frac{1}{2} \left[ \ln \sigma_{k'_s}^2 + \frac{(x_s - \mu_{k'_s})^2}{\sigma_{k'_s}^2} \right] + \beta \sum_{t \in \mathcal{V}(s)} [1 - \delta(k'_s, k_t)] \end{cases} \quad (6)$$

où  $\mathcal{V}(s)$  désigne l'ensemble des pixels  $t$  voisins de  $s$ .

- Si  $U'_s < U_s$ , remplacer  $k_s$  par  $k'_s$ . Sinon, faire de même, 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.

## Exercice 1 : segmentation par classification supervisée

Écrivez les fonctions `estimation`, `attache_donnees`, `regularisation` et `recuit`, appelées par `exercice_1` :

- Les paramètres de chaque classe (moyenne et écart-type) sont estimés par la fonction `estimation`, à partir d'un échantillon sélectionné par l'utilisateur, d'où le caractère *supervisé* de la classification.
- La fonction `attache_donnees` doit retourner une matrice tridimensionnelle contenant, pour chaque pixel  $s$ , la valeur de l'attache aux données  $\frac{1}{2} \left[ \ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right]$ , relativement à chacune des  $N$  classes.
- La fonction `regularisation` doit retourner le terme de régularisation  $\sum_{\{s,t\} \in \mathcal{C}_2} [1 - \delta(k_s, k_t)]$  de l'énergie  $U(\mathbf{k})$ , pour laquelle il est conseillé d'utiliser l'opérateur « différent de », qui s'écrit  $\sim$  en Matlab.

Observez ce qui se passe dans les cas suivants :

- Si le nombre  $N$  de classes est différent de 4.
- Lorsque les échantillons sont mal sélectionnés.
- Si  $T_0 = 0$ , ce qui élimine tout changement de configuration faisant croître l'énergie.

## Classification non supervisée

Pour éviter à l'utilisateur de sélectionner à la main un échantillon de chaque classe, il est envisageable d'estimer les paramètres des  $N$  classes, en cherchant un mélange de  $N$  gaussiennes coïncidant avec l'histogramme  $f(x)$  de l'image en niveaux de gris :

$$f(x) = \sum_{i=1}^N \frac{p_i}{\sigma_i \sqrt{2\pi}} \exp \left\{ -\frac{(x - \mu_i)^2}{2\sigma_i^2} \right\}, \quad x \in \{1, \dots, 255\} \quad (7)$$

où  $\mu_i$ ,  $\sigma_i$  et  $p_i$  désignent, respectivement, la moyenne, l'écart-type et le poids de la  $i^{\text{ème}}$  gaussienne. L'estimation des paramètres de ce modèle revient donc à résoudre un problème en moindres carrés linéaire vis-à-vis de  $p_i$ , mais non linéaire vis-à-vis de  $\mu_i$  et  $\sigma_i$  :

$$(\hat{\mu}_i, \hat{\sigma}_i, \hat{p}_i)_{i \in \{1, \dots, N\}} = \arg \min_{(\mu_i, \sigma_i, p_i)_{i \in \{1, \dots, N\}}} \sum_{x=0}^{255} \left[ f(x) - \sum_{i=1}^N \frac{p_i}{\sigma_i \sqrt{2\pi}} \exp \left\{ -\frac{(x - \mu_i)^2}{2\sigma_i^2} \right\} \right]^2 \quad (8)$$

## Exercice 2 : segmentation par classification non supervisée

Écrivez la fonction `argument_eq_8`, appelée par le script `exercice_2`, permettant de calculer l'argument du problème d'optimisation (8). L'estimation des paramètres  $\mu_i$  et  $\sigma_i$  peut être effectuée en minimisant l'argument de (8) par tirages aléatoires. Les moyennes  $\mu_i$  sont recherchées dans l'intervalle  $[0, 255]$ , mais les écarts-types  $\sigma_i$  peuvent être recherchés dans l'intervalle  $[10, 25]$ , afin d'accélérer la résolution. Quant à l'estimation des poids  $p_i$ , elle est facilitée par le fait que le problème en moindres carrés (8) est linéaire en  $p_i$ . Pour estimer les poids, à chaque tirage aléatoire de  $2N$  valeurs réelles  $(\mu_i, \sigma_i)$ ,  $i \in \{1, \dots, N\}$ , il faut donc résoudre un système linéaire du type  $\mathbf{A} \mathbf{P} = \mathbf{F}$ , où  $\mathbf{P} = [p_1, \dots, p_N]^T$  et où  $\mathbf{F}$  contient les 256 valeurs de l'histogramme.

Bien que beaucoup plus lente, à cause de l'estimation des paramètres par tirages aléatoires, cette méthode doit vous permettre d'atteindre un pourcentage de bonnes classifications comparable à celui de l'exercice 1, et ce de manière entièrement automatique !

## Exercice 3 : utilisation de l'algorithme EM (facultatif)

La méthode la plus adaptée à la résolution d'un problème tel que (8) est l'algorithme EM, qui a déjà été vu dans le TP2. Libre à vous d'écrire une nouvelle version du script précédent, de nom `exercice_3`.