

TP4 – Segmentation par classification

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 μ_1, \dots, μ_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 :

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

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

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

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

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

où \mathcal{C}_2 désigne l'ensemble des cliques de cardinal 2, c'est-à-dire l'ensemble des paires $\{s, t\}$ de pixels voisins (c'est le système de voisinage des « 8 plus proches voisins » qui est utilisé). Nous déduisons de (1), (2) et (3) :

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

Chercher le maximum de $p(\mathbf{K} = \mathbf{k} | \mathbf{X} = \mathbf{x})$ équivaut donc à chercher le minimum de l'*énergie* $U(\mathbf{k})$. Pour ce faire, il ne suffit pas d'optimiser l'énergie localement, en chaque pixel $s \in \mathcal{S}$, ce qui s'écrirait :

$$\hat{k}_s = \arg \min_{k_s \in \{1, \dots, N\}} \left\{ \frac{1}{2} \left[\ln \sigma_{k_s}^2 + \frac{(x_s - \mu_{k_s})^2}{\sigma_{k_s}^2} \right] + \beta \sum_{t \in \mathcal{V}(s)} [1 - \delta(k_s, k_t)] \right\} \quad (5)$$

Pour trouver le minimum global de $U(\mathbf{k})$, il est impensable de tester les $N^{\text{card}(\mathcal{S})}$ configurations possibles. Nous pouvons en revanche utiliser le *recuit simulé*. Cette méta-heuristique 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 :

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

- 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 q_{\max} n'est pas atteint.

Exercice 1 : segmentation par classification supervisée

Écrivez les fonctions `attache_aux_donnees`, `regularisation` et `recuit_simule`, qui sont appelées par le script `exercice_1` :

- La fonction `attache_aux_donnees` doit retourner une matrice tridimensionnelle contenant, pour chaque pixel, la valeur du terme d'attache aux données de (5) relativement à chacune des N classes.
- La fonction `regularisation` doit retourner la valeur du terme de régularisation de (5).

Les paramètres de chaque classe (moyenne et écart-type) sont estimés à partir d'un échantillon sélectionné par l'utilisateur, d'où le caractère *supervisé* de la classification.

Ajustez les paramètres T_0 , α , q_{\max} et β de façon à maximiser le pourcentage de bonnes classifications. Observez ensuite ce qui se passe dans les cas suivants (liste non exhaustive) :

- 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 (voir cours) :

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

où μ_i , σ_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 le problème en moindres carrés suivant :

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

qui est linéaire vis-à-vis de l'inconnue p_i , mais non linéaire vis-à-vis de μ_i et σ_i . La méthode la plus adaptée à la résolution de ce problème est l'algorithme EM, qui a déjà été vu dans le TP3. Nous proposons ci-après une méthode moins performante, mais plus facile à mettre en œuvre.

Exercice 2 : segmentation par classification non supervisée

Faites une copie du script `exercice_1`, de nom `exercice_2`, dans lequel vous remplacerez l'appel à la fonction `estimation` par un appel à la fonction `estimation_non_super`, que vous devez écrire.

L'histogramme normalisé $f(x)$ d'une image en niveaux de gris se calcule à l'aide de la fonction `ksdensity`. L'estimation des paramètres μ_i et σ_i peut être effectuée en minimisant l'argument du problème (8) par tirages aléatoires. Les moyennes μ_i doivent être recherchées dans l'intervalle $[0, 255]$, mais vous pourrez limiter la recherche des écarts-types σ_i à l'intervalle $[10, 25]$ pour 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 (il est inutile de retourner les paramètres p_i , qui ne sont pas nécessaires à la méthode de classification par MAP). Pour estimer ces poids, à chaque tirage aléatoire de $2N$ valeurs réelles (μ_i, σ_i) , $i \in \{1, \dots, N\}$, vous devez 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 la couleur (exercice facultatif)

Il serait dommage de ne pas utiliser l'information de couleur, qui constitue un indice majeur pour segmenter une image. Faites une copie des fonctions `estimation` et `attache_donnees`, de noms `estimation_RVB` et `attache_donnees_RVB`, qui sont appelées par le script `exercice_3`. Modifiez ces fonctions de manière à segmenter l'image `cellules.jpg` par classification supervisée.

Rappel – La loi normale en dimension d s'écrit, si $\mathbf{x} \in \mathbb{R}^d$:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \sqrt{\det \Sigma}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu) \right\} \quad (9)$$

où $\mu \in \mathbb{R}^d$ et $\Sigma \in \mathbb{R}^{d \times d}$ désignent, respectivement, la moyenne et la matrice de variance/covariance.