UNIVERSITÉ TOULOUSE III PAUL SABATIER — Université de Toulouse

TP4 : Classification bayésienne

M1 IAA

L'objectif est de réaliser un classifieur bayésien permettant de classer les images de trois espèces de fleurs (ne recopiez pas les images, afin de préserver votre quota). Lancez le script `Donnees.py`, qui affiche des images de pensées, d'œillets et de chrysanthèmes. Vous constatez que ces images n'ont pas toutes la même taille.

Base de données : Pensées, Oeillets et Chrysanthèmes

1 Prétraitement : calcul de la couleur moyenne d'une image

Dans un premier temps, vous allez classer les images selon la couleur moyenne de chaque espèce de fleurs. En chaque pixel de chaque image, les trois niveaux de couleur $(R, V, B) \in [0, 255]^3$ sont d'abord transformés en *niveaux de couleur normalisés* (r, v, b), qui sont définis de la manière suivante :

$$(r, v, b) = \frac{1}{\max\{1, R + V + B\}} (R, V, B)$$

Le principal intérêt des niveaux de couleur normalisés est que deux valeurs parmi (r, v, b) permettent de déduire la troisième, puisque $r+v+b = 1$, sauf dans le cas exceptionnel où $(r, v, b) = (0, 0, 0)$. Une image est donc caractérisée par les moyennes $(\bar{r}, \bar{v}, \bar{b})$, ou plus simplement par $(\bar{r}, \bar{v})$, puisque $\bar{r} + \bar{v} + \bar{b} = 1$, c'est-à-dire par un vecteur $\mathbf{x} = [\bar{r}, \bar{v}] \in \mathbb{R}^2$ qu'on appelle sa *couleur moyenne*. Compte tenu des différences de couleurs moyennes entre les trois espèces de fleurs, on postule que ce vecteur suffira à les distinguer.

Écrivez la fonction `Pretraitement` dans le script `Donnees.py` qui calcule la couleur moyenne d'une image. La fonction `Pretraitement` est censée normaliser les couleurs d'une image et calculer les couleurs moyennes $[\bar{r}, \bar{v}]$ dans $\mathbb{R}^2$ de l'image.

En affichant les couleurs moyennes de l'ensemble des images de fleurs sous la forme de trois nuages de points de $\mathbb{R}^2$, la couleur moyenne vous semble-t-elle une caractéristique suffisamment discriminante de ces trois espèces de fleurs ?

2 Estimation de la vraisemblance de chaque espèce de fleurs

Les trois nuages de points précédents peuvent être modélisés par des lois normales bidimensionnelles. Il est rappelé que la densité de probabilité d'une loi normale s'écrit, en dimension d :

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\,(\det \Sigma)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu)^{\top}\right\} \tag{1}$$

où :

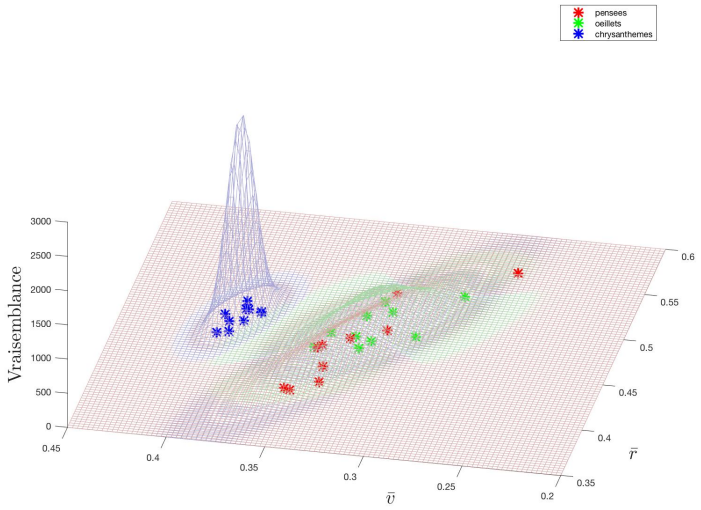
— μ désigne l'espérance (la moyenne) des vecteurs $\mathbf{x} \in \mathbb{R}^d$: $\mu = E[\mathbf{x}] = \int \mathbf{x}\, p(\mathbf{x})\, d\mathbf{x}$.

— Σ désigne la matrice de variance/covariance : $\Sigma = E[(\mathbf{x}-\mu)^T(\mathbf{x}-\mu)]$.

Dans le cadre bayésien, la vraisemblance de la classe ω_i, qui est caractérisée par la moyenne μ_i et la matrice de variance/covariance Σ_i, peut être modélisée par une loi normale analogue à (1) :

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{d/2}\,(\det \Sigma_i)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_i){\Sigma_i}^{-1}(\mathbf{x}-\mu_i)^{\top}\right\}, \quad i \in [1,3]$$

Il faut donc estimer les paramètres μ_i et Σ_i des trois classes correspondant aux trois espèces de fleurs.



Vraisemblance des fleurs

En s'inspirant des fichiers python du TP3, écrivez la fonction `fit`, dans un fichier `Bayes.py`, permettant d'effectuer l'estimation empirique des paramètres d'une loi normale bidimensionnelle ($d = 2$) à partir des vecteurs $\mathbf{x} = [\bar{r}, \bar{v}]$ stockés dans la matrice de données `X`. La fonction `fit` est censée estimer les paramètres μ_i et Σ_i des trois classes ω_i correspondant aux trois espèces de fleurs, à partir des matrices `X_pensees`, `X_oeillets` et `X_chrysanthemes`, puis superposer la vraisemblance de chaque classe (en perspective) au nuage de points à partir de laquelle elle a été estimée.

Remarque : Vous pourrez vous inspirer de la fonction `PlotSurface.py` pour afficher les vraisemblances des trois espèces de fleurs.

3 Classification d'images de fleurs

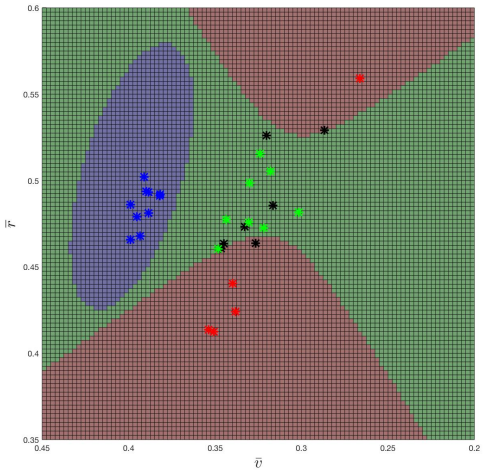
Nous souhaitons maintenant prédire à quelle espèce de fleurs une image requête **x** doit être associée. Comme nous avons utilisé des données étiquetées (chacune des images étant associée à une espèce de fleurs), il s'agit

de **classification supervisée**. Un premier type de classification consiste à affecter à $\mathbf{x}$ la classe ω_i qui maximise la vraisemblance $p(\mathbf{x}|\omega_i)$. Il s'agit alors d'un classifieur par « maximum de vraisemblance ».

Par ailleurs, la règle de Bayes donne l'expression suivante de la **probabilité a posteriori** $p(\omega_i|\mathbf{x})$, c'est-à-dire de la probabilité pour que la classe ω_i contienne $\mathbf{x}$:

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)\, p(\omega_i)}{p(\mathbf{x})} \tag{2}$$

Il semble naturel d'affecter à $\mathbf{x}$ la classe ω_i qui maximise $p(\omega_i|\mathbf{x})$. Une telle classification est dite par « maximum a posteriori » (MAP). Sachant que le dénominateur $p(\mathbf{x})$ de (2) est indépendant de ω_i, il n'est pas nécessaire de le connaître pour trouver le maximum des $p(\omega_i|\mathbf{x})$. En revanche, il est nécessaire de connaître la « probabilité a priori » $p(\omega_i)$ de chaque classe ω_i, faute de quoi on fait généralement l'hypothèse que les classes sont équiprobables (l'estimateur par maximum a posteriori revient alors à un estimateur par maximum de vraisemblance).



Maximum a posteriori sur les fleurs

Écrivez une fonction `predict` dans le fichier `Bayes.py` où, en jouant sur les probabilités a priori des trois classes, vous essaierez de maximiser le pourcentage d'images correctement classées en s'inspirant des codes du TP3.

Remarque : On pourrait utiliser la fonction `log` pour éliminer l'exponentielle et éviter ainsi des problèmes numériques potentiels (comme vu en cours/TD)

4 Amélioration du classifieur

Même en jouant sur les probabilités a priori, le classifieur obtenu reste décevant. Or, l'observation attentive des images de pensées et d'œillets, dont les couleurs moyennes sont similaires, montre que ces deux espèces de fleurs ne sont pas structurées de la même façon : les pensées sont plus sombres au centre, c'est-à-dire au niveau du pistil. Cela suggère de ne pas seulement calculer la couleur moyenne des images, mais de scinder chaque image en deux parties complémentaires : le centre C (notion à préciser) et le pourtour P (complémentaire de C).

Écrivez un fichier `BayesBetter.py` reprenant le principe du classifieur MAP que précédemment, mais utilisant trois caractéristiques pour décrire une image, à savoir le couple de valeurs $(\bar{r}, \bar{v})$ calculées sur le pourtour P et la valeur $\bar{r}$ calculée sur le centre C.