Laboratoires d'Évaluation d'algorithmes pour l'identification de visages statiques

SYS828 – Systèmes biométriques

Responsable et enseignant: Eric GRANGER
Auxiliaire de laboratoire: George Ekladious

Session: A2019

Laboratoire 3 – Méthodologie

Afin d'observer l'impact d'une méthode d'extraction de caractéristiques et la réduction de dimensionalité sur les performances de classificateurs, nous devons définir une méthodologie pour l'entraînement de ces derniers. Les objectifs de ce laboratoire sont de se familiariser avec les étapes d'un apprentissage supervisé et les moyens utilisés pour évaluer les performances de classificateurs. Les expérimentations seront effectuées à l'aide du classificateur kNN.

En première partie, nous décrirons le classificateur kNN ainsi que le protocole utilisé pour l'entraînement supervisé d'un classificateur. Ensuite, nous nous attarderons sur les éléments entourant l'évaluation de performance de classificateurs : les indicateurs de performances, les estimateurs et les techniques de validation. Finalement, nous expliquerons la partie expérimentale pour ce laboratoire.

3.1 Entraînement supervisé du classificateur kNN

3.1.1 Classificateur kNN

Le classificateur des k plus proche voisins (kNN pour k *Nearest Neighbor*) est un est algorithme qui classe les objets selon leurs proximité au données utilisées pour l'entraînement dans l'espace de caractéristiques $\mathbb{R}^I$. C'est une approche par modélisation simple qui approxime la frontière de décision localement. Il n'y a pas d'entraînement proprement dit et les calculs sont seulement effectués lors de la classification.

Comme la Figure 1 le démontre, la classification est effectué par un vote majoritaire de sont voisinage et l'objet est assigné à la classe la plus présente parmi les k plus proche voisins. Bien que plusieurs distances peut être utilisé pour définir le voisinage, la mesure la plus utilisée est normalement la distance euclidienne, définie par

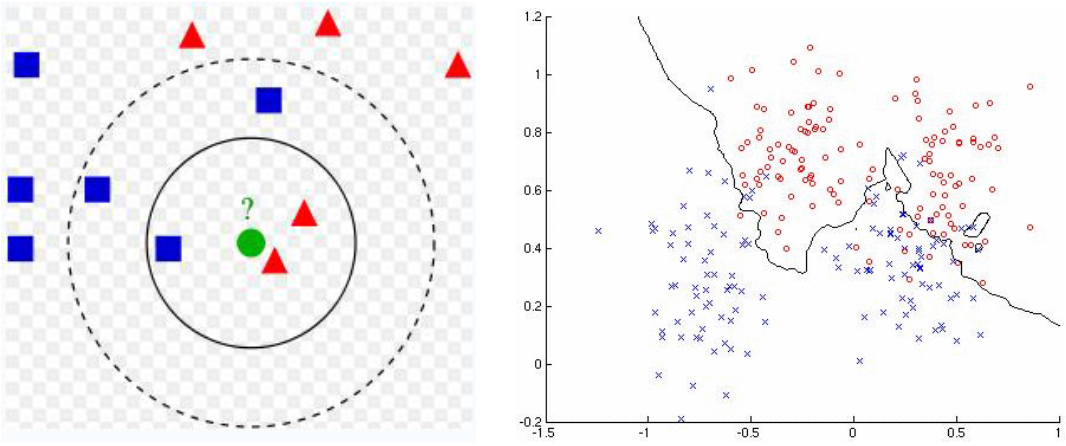
$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_1 - q_1)^2 + ... + (p_I - q_I)^2} = \sqrt{\sum_{i=1}^{I} (p_i - q_i)^2} \tag{1}$$

où p et q sont deux points dans un espace $\mathbb{R}^I$.

Le classificateur kNN est une méthode non-paramétrique qui ne nécessite pas d'établir une hypothèse au préalable sur la nature des distributions de données (contrairement à une régression linéaire, par exemple). Le seul paramètre à déterminer est la taille du voisinage (k) et est défini à partir des données. Une grande valeur de k réduit l'effet du bruit sur les données, mais définie des frontières de décisions sans tenir compte de particularités locales.

3.1.2 Apprentissage avec validation

Une méthode généralement utilisé pour déterminer les paramètres de divers classificateur est l'*apprentissage avec validation*. Cette technique consiste à séparer les données utilisées pour l'apprentissage en deux groupes – un premier dédié à l'entraînement et l'autre à la validation. Nous faisons alors évoluer les paramètres d'un algorithme d'apprentissage à l'aide d'entraînements successifs avec les mêmes données, et chaque présentation des données d'entraînement est alors appelée *époque* d'entraînement. Comme le montre la Figure 2, les performances du classificateur sont évaluées avec les données de validation après chaque époques d'entraînement et l'apprentissage est poursuivie jusqu'à ce que le classificateur cesse de s'améliorer.



(a) Classification (b) Frontière de décision

Figure 1: Exemple de classification d'un exemple avec kNN (1a) et de frontière de décision pour $k = 8$ (1b). Dans la Figure 1a, si $k = 2$, l'exemple est un triangle, alors que si $k = 3$, l'exemple est un carré.

Cette technique est couramment utilisée pour faire évoluer les poids synaptiques des réseaux de neurones. Pour le classificateur kNN, le seul paramètre à déterminer est la valeur de k. Dans ce cas spécifique, nous entraînerons kNN en utilisant les même données, mais différentes valeurs de k, et la valeur de k donnant les meilleures performances avec les données de validation sera conservée.

3.2 Évaluation de classificateurs

Afin d'évaluer les performances de classificateurs, une base de données de test, indépendante à la base de données d'apprentissage est utilisée. Chaque algorithme d'apprentissage est évalué en fonction de ses capacités de généralisation, des ressources en mémoire utilisées et du coût de calcul. Les principaux indicateurs de performance utilisés sont :

A) **Le taux de classification :** ratio de bonnes classifications obtenues par rapport à l'ensemble des données de la base de test. Pour les problèmes à plusieurs classes, il est également possible d'utiliser une *matrice de confusion* qui comptabilise les taux de classifications pour chaque classe.

B) **La grandeur du classificateur :** le nombre d'éléments utilisés pour modéliser les données d'*entraînement*. Dans le cas des réseaux de neurones, par exemple, cette valeur serait le nombre de neurones utilisés. La *compression* peut également être utilisée. Plutôt que d'indiquer directement les ressources utilisées, la compression indique le nombre moyen d'exemples modélisés par chaque de neurones[1].

[1]Dans le cas de kNN, comme l'ensemble des données d'entraînement est directement utilisé pour modéliser le problème, la grandeur du classificateur est le nombre données d'entraînement, et la compression est de 1.

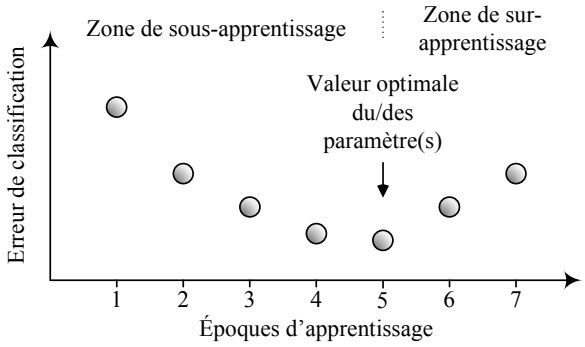


Figure 2: Variation de l'erreur de classification en fonction du nombre d'époque d'entraînement. Les valeurs optimales des paramètres sont déterminées lorsque l'erreur de classification est au minimum. Avant et après ce moment, l'algorithme est respectivement en sous-apprentissage et en sur-apprentissage. En sous-apprentissage, le classificateur n'est pas optimal alors qu'en sur-apprentissage, les données sont apprises par coeur et le classificateur perd ses capacités de généralisation.

C) **Le temps de convergence :** Le nombre d'itérations nécessaires pour arrêter l'algorithme d'apprentissage. Pour les réseaux de neurones et kNN, ceci revient à utiliser le nombre d'époques d'apprentissage.

3.2.1 Estimation des indicateurs

Bien que nous ayons défini des indicateurs de performance, il est impossible d'avoir la valeur exacte de ces indicateurs. En effet, l'utilisation d'une quantité finie de données pour l'évaluation des performances implique que nous avons une connaissance incomplète sur la nature des données. C'est pourquoi nous devons utiliser des *estimateurs* qui seront calculés en effectuant plusieurs réplications d'une même expérience.

L'estimateur généralement utilisé est la moyen, définie par

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \tag{2}$$

et les intervalles de confiances sont définis à l'aide des distributions normale ou de Student. Ainsi il sera possible d'affirmer, à l'aide de tests statistiques s'il y a des différences significatives entre les différents systèmes de classification. Notez bien qu'il est également possible d'utiliser la médiane avec laquelle il existe plusieurs tests statistiques intéressants.[2].

Afin d'effectuer plusieurs réplications du processus d'apprentissage avec validation, il existe deux méthodes de validation.

A) **Validation "hold-out" :** utilisée lorsque beaucoup de données sont disponible, la validation "hold-out" effectue plusieurs réplications à l'aide de groupes de données distincts. Par exemple, pour 5 réplications, 5 base de données d'entraînement, de validation et de test seront utilisées.

[2]Voir des livres de statistiques pour plus d'information sur les estimateurs, intervalles de confiance et tests statistiques

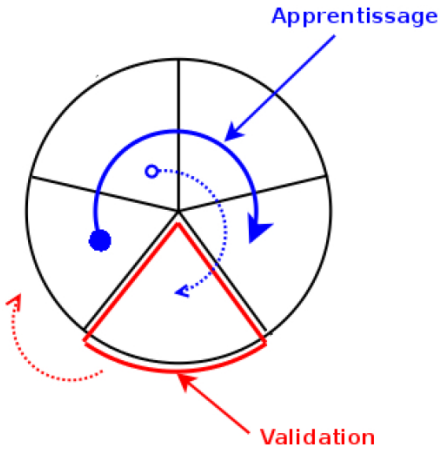


Figure 3: Illustration de la validation croisée à 5 blocs.

B) **Validation croisée :** quand la base de données consacrée à l'entraînement est petite on utilise l'algorithme de validation croisée à K blocs (*K-fold cross-validation*). Comme il est illustré ans la Figure 3, pour 5 réplications ($K = 5$), on divise la base d'entraînement en cinq parties – quatre pour l'apprentissage et une pour la validation – et on répète l'apprentissage cinq fois avec rotations des parties.

3.3 Expérimentations

3.3.1 Exploration avec les données synthétiques

A) Utiliser le script $exp3\_methodologie\_a.m$ pour déterminer le nombre de caractéristiques. Vous devez utiliser le classificateur 1NN (kNN avec $k = 1$) et un entraînement avec validation "hold-out". Pour chaque valeur du nombre de caractéristiques, vous devez entraîner 1NN et déterminer le taux d'erreur avec la base de validation

B) Utiliser le script $exp3\_methodologie\_b.m$ pour entraîner le classificateur kNN en utilisant un apprentissage avec validation "hold-out" et la base de données synthétiques vues dans le laboratoire précédent. Faire 5 réplications et calculez les indicateurs de performance pour kNN.

C) Refaire la même procédure avec validation croisée.

Fonction utiles (PRTools) : `knnc`, `plotc`, `testc`, `classc`.

3.3.2 Exploration avec les données réelles

Écrire et exécuter un script (exp3_faces.m) pour entraîner le classificateur k-NN en utilisant un apprentissage avec validation "cross-validation" et avec les algorithmes d'extraction de caractéristiques:

A) Séparer la base de données d'AT&T en bases d'apprentissage et de test. Les 5 premiers visages de chaque individu sont assignés à la base d'apprentissage et les 5 derniers, à la base de test (voir laboratoire 2).

B) Séparer la base d'apprentissage de manière à faire une validation croisé en 5 blocs. Vous devriez avoir 5 base d'entraînement (contenant 4/5 des données) et 5 bases de validation (le dernier 1/5 des données).

C) Déterminer le nombre de caractéristiques par utiliser le classificateur 1NN (kNN avec $k = 1$) et un entraînement avec validation croisée. Pour chaque valeur du nombre de caractéristiques, vous devez entraîner 1NN et déterminer le taux d'erreur avec la base de validation.

D) En utilisant le meilleur nombre de caractéristiques, entraîner les classificateurs à l'aide d'une validation croisée afin de déterminer le paramètre k.

E) Évaluer la performance sur la base de test.

F) Refaire les mêmes étapes avec des bases de données traitées avec les algorithmes d'extraction de caractéristiques.