k Nearest Neibors (k-NN)
k-plus proches voisins (k-PPV)

Master AIDN: Applications Interactives et Données Numériques

Sylvie Gibet

1

1

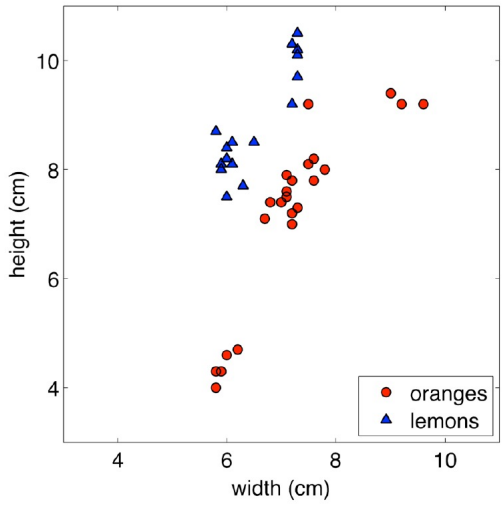
Plus proches voisins

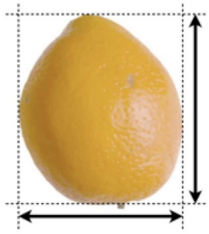
- Modèle non paramétrique
  - À base de distance
  - Frontières de décision non linéaires

2

2

Oranges et citrons

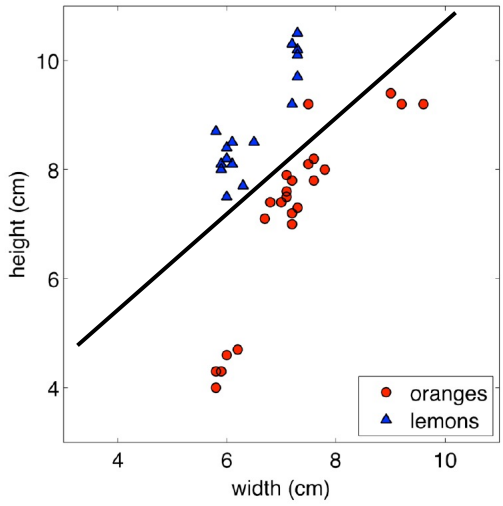




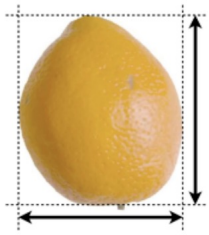
3

3

Oranges et citrons



On peut construire une frontière de décision linéaire :
$y = \text{sign}(w_0 + w_1x_1 + w_2x_2)$



4

4

Que veut dire classification linéaire

- Classification : intrinsèquement non linéaire
  - Regroupe des objets différents dans la même classe, par conséquence une différence dans le vecteur de features produit zéro changement dans la réponse
- Classification linéaire
  - La fonction *h* qui prédit est linéaire :

$$h(\boldsymbol{x}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0$$

  - La fonction *f* qui prend la décision est non-linéaire :

$$y(\mathbf{x}) = f(h(\mathbf{x}))$$

  - Méthode paramétrique (paramètres $w_0, w_1, \ldots, w_p$)

5

5

Plus proches voisins

- Méthode non paramétrique
  - Training : enregistrement des données d'apprentissage
  - Les exemples de test sont classifiés à partir d'exemples similaires de l'espace d'entraînement
  - La similarité est exprimée par une distance

6

6

Plus proches voisins

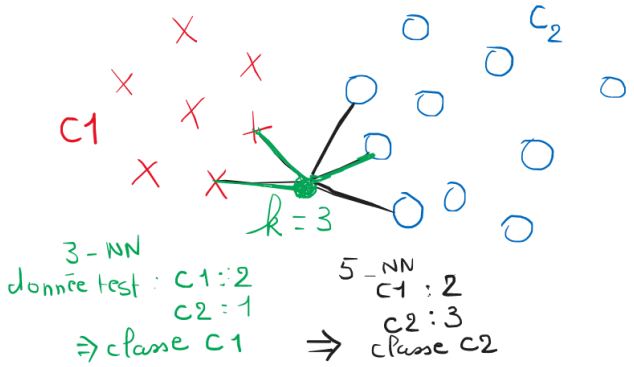
- Ensemble dans l'espace Euclidien : $\mathbf{x} \in \Re^d$
- Pour un exemple de test, on évalue sa classe à partir de celles des exemples les plus proches de l'espace d'entraînement
- Distance classiquement utilisée : distance Euclidienne :

$$||\mathbf{x}^{(a)} - \mathbf{x}^{(b)}||_2 = \sqrt{\sum_{j=1}^{d} (x_j^{(a)} - x_j^{(b)})^2}$$

7

7

Exemple



8

8

Plus proches voisins

- **Algorithme**
  - 1. Trouver l'exemple (**x***,*c**) de l'espace d'entraînement le plus proche de l'exemple **x**. C'est-à-dire :

$$\mathbf{x}^* = \underset{\mathbf{x}^{(i)} \in \text{train. set}}{\text{argmin}} \ \text{distance}(\mathbf{x}^{(i)}, \mathbf{x})$$

  - 2. Sortie de classification : $y = c^*$

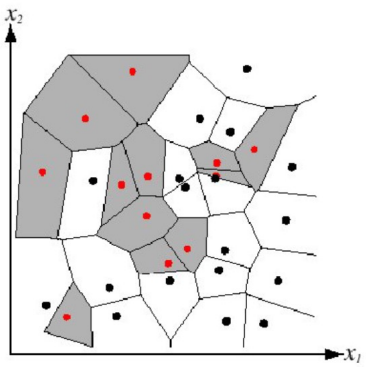
Remarque : on n'a pas vraiment besoin de garder la racine carrée !

9

9

Plus proches voisins : frontières de décision

- L'algorithme du plus proche voisin ne calcule pas explicitement les frontières de décision, mais celles-ci peuvent être inférées
- Frontières de décision : visualisées par un diagramme de Voronoi
  - Montre comment l'espace est subdivisé en classes
  - Chaque segment est équisistant entre deux points de classes opposées

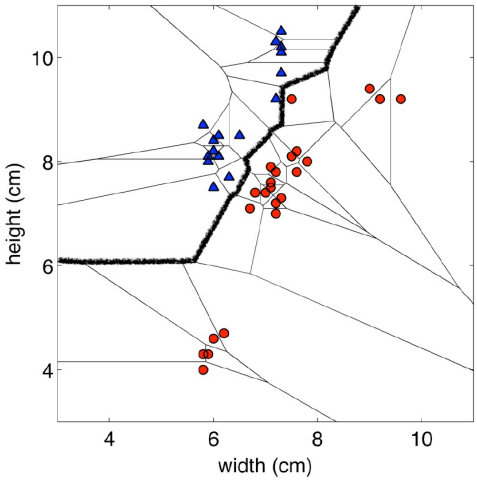


10

10

Plus proches voisins : frontières de décision

- Exemple en 2D de frontières de décision

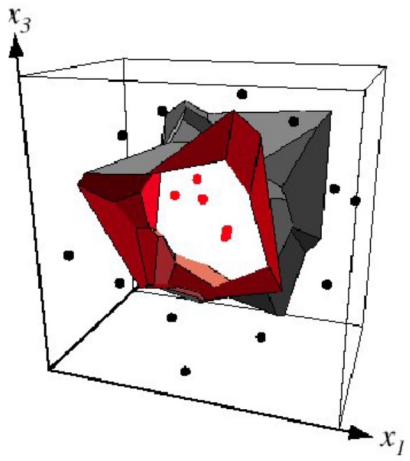


11

11

Plus proches voisins : frontières de décision

- Exemple en 3D de frontières de décision

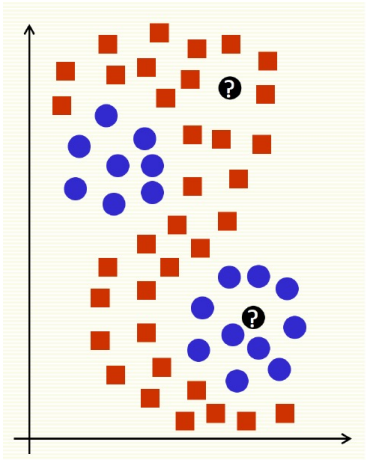


12

12

Plus proches voisins : influence de k

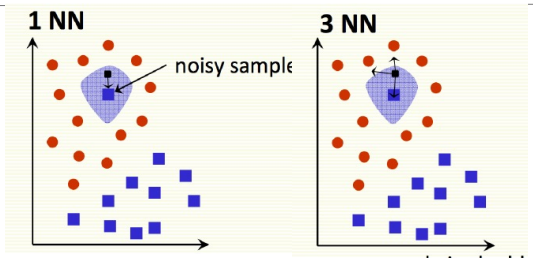
- L'accuracy (precision) des méthodes k-NN peuvent dépendre de la valeur k



13

13

Plus proches voisins : influence de k



Toutes les données dans la région bleue seront mal-classifiées : classe bleue

Toutes les données dans la région bleue seront classifiées correctement : classe rouge

- Les méthodes k-NN sont sensibles aux données mal-étiquetées (classe "bruit")
- Solution : on filtre en prenant les k plus proches voisins (ici k = 3), et en votant

14

14

Plus proches voisins

- **Algorithme**
  - 1. Trouver les k exemples $\{\mathbf{x}^{(i)}, c^{(i)}\}$ les plus proches de l'exemple **x**.
  - 2. Sortie de classification : classe majoritaire

  Pour les k-PPV (k-NN) on compte le nombre de classes et on sélectionne la classe la plus représentée

15

15

k-NN

- Comment sélectionner *k* ?
  - *k* grand peut conduire à de meilleures performances
  - Mais si *k* est trop grand on peut choisir des exemples qui sont très éloignées (ce ne sont plus des voisins)
  - Utiliser de **la cross-validation** pour trouver *k*
  - Règle implicite : $k < sqrt(n)$, *n* étant le nombre d'exemples d'apprentissage

16

16

k-NN : problèmes et solutions

- Si certains *features* (attributs, ou coordonnées de **x**) ont des plages de variations plus grandes, alors ils risquent de prendre plus d'importance :
  - Échelle normalisée
    - Simple option: mettre à l'échelle linéairement chaque feature en le forçant à rester dans la plage [0,1] : x_j/max_{xj}
    - mettre à l'échelle linéairement chaque feature pour avoir une moyenne de 0 et une variance de 1 : normaliser chaque feature x_j : $(x_j - m)/\sigma$
  - Attention : parfois l'échelle a une importance !

17

17

k-NN : problèmes et solutions

- Des attributs non signifiants ou corrélés rajoutent du bruit aux mesures de distance
  - Éliminer certains *features*
    - Exemple : les données de position de chaque articulation d'une chaîne articulée sont dépendantes entre elles
  - Mettre des poids aux *features*
    - Exemple : mettre du poids sur certaines parties de l'image par rapport à d'autres

18

18

k-NN : problèmes et solutions

- Features non métriques (symboles)
  - Trouver d'autres distances :
    - Exemple de la distance de hamming (exemple ci-dessous) -> associe le nombre de positions où les deux suites diffèrent.

```
def hamming_distance(s1, s2) -> int:
    """Return the Hamming distance between equal-length sequences."""
    if len(s1) != len(s2):
        raise ValueError("Undefined for sequences of unequal length.")
    return sum(el1 != el2 for el1, el2 in zip(s1, s2))
```

    - (zip : prend des iterables, les agrege et les met dans un tuple)

Exemple : S1 = ['A','C','T','G'] et S2 = ['A','C','G','T']

Return : ?

19

19

k-NN : problèmes et solutions

- Features non métriques (symboles)
  - Trouver d'autres distances :
    - Distance cosinus : métrique employée en fouille de textes

$$\cos\theta = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}.$$

mot1 = ['M','A','I','S','O','N'] ; mot2 = ['M','A','T','I','N','S']

print(len(mot1) - hamming_distance(mot1,mot2)) = ?

20

20

k-NN : problèmes et solutions

- Complexité en temps lors des tests : pour trouver un PPV d'un point en entrée **x**, on doit calculter les distances de tous les *N* exemples d'entraînement : complexité en $O(k.d.N)$
  - Utiliser des sous-ensembles de dimensions
  - Pré-trier les exemples dans des structures de données rapides (exemple le kd-trees)
  - Calculer uniquement une distance approchée (e.g, LSH : locality sensitive hashing) : adaptée à la *distance* de Hamming dans un espace binaire de dimension d
  - Enlever les données redondantes (e.g., condenser)
  - Trouver des données représentatives de cluster (centroïdes)

21

21

k-NN : problèmes et solutions

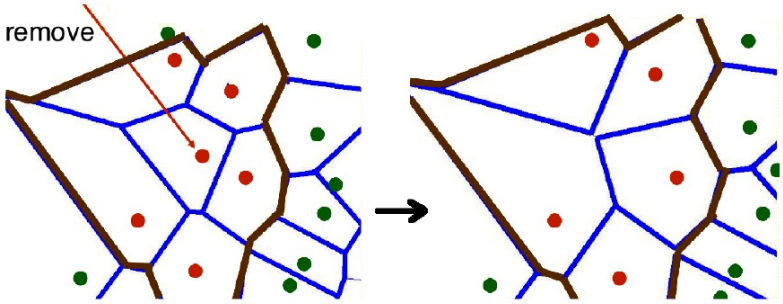
- Besoins en stockage : on doit stocker toutes les données d'entraînement
  - Enlever les données redondantes (e.g., condenser)
  - Pré-trier augmente souvent les besoins de stockage
- Données hautement multi-dimensionnelles
  - La quantité de données d'entraînement requise augmente avec la dimension
  - Le coût de calcul augmente également

22

22

k-NN : enlever les redondances

- Si les voisins de Voronoi ont la même classe, un exemple n'est pas utile, l'enlever



23

23

Exemple : classification des chiffres

- Bonnes performances si beaucoup de données

- Yann LeCunn – MNIST Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: $d = 784$
  - 60,000 training samples
  - 10,000 test samples
- Nearest neighbour is competitive

	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

24

24

Exemple : où a été prise cette photo ?

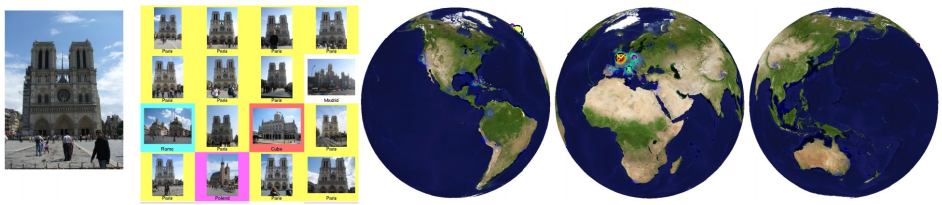
- Problème : Où (quel pays, coordonnées GPS)

[Paper: James Hays, Alexei A. Efros. im2gps: estimating geographic information from a single image. CVPR'08. Project page: http://graphics.cs.cmu.edu/projects/im2gps/]

25

Exemple : où a été prise cette photo ?

- Problème : Où (quel pays, coordonnées GPS)
  - Récupérer 6M images à partir de Flickr avec infos GPS (échantillonnage dense)
  - Représenter chaque image par des features significatifs
  - Faire du k-NN !

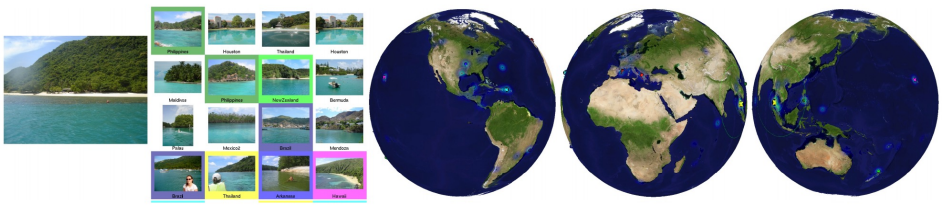


26

26

Exemple : où a été prise cette photo ?

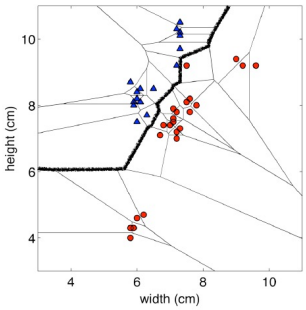
- Problème : Où (quel pays, coordonnées GPS)
  - Récupérer 6M images à partir de Flickr avec infos GPS (échantillonnage dense)
  - Représenter chaque image par des features significatifs
  - Faire du k-NN (k grand donne de meilleurs résultats : k = 120)



27

27

k-NN - Récapitulatif



- Frontières de décision de formes complexes
- Si beaucoup d'exemples, k-NN marche plutôt bien
- Problèmes
  - Complexité croit avec le nombre d'exemples
  - Sensible à la classe "bruit"
  - Sensible aux échelles des features
  - Les distances ont moins de sens en haute dimension
- Biais inductif : quelle frontière de séparation s'attend-on à avoir ?

28

28