

IFT 615 – Intelligence Artificielle

Formes d'apprentissage

Algorithme des K plus proches voisins

Professeur: Froduald Kabanza

Assistants: D'Jeff Nkashama

Sujets couverts

- Formes d'apprentissage
- Classification avec l'algorithme des K plus proches voisins

Types de problèmes d'apprentissage

Apprentissage supervisé



Algorithme d'apprentissage



Classifieur

- Chat
- Canard
- Chien

Apprentissage non supervisé

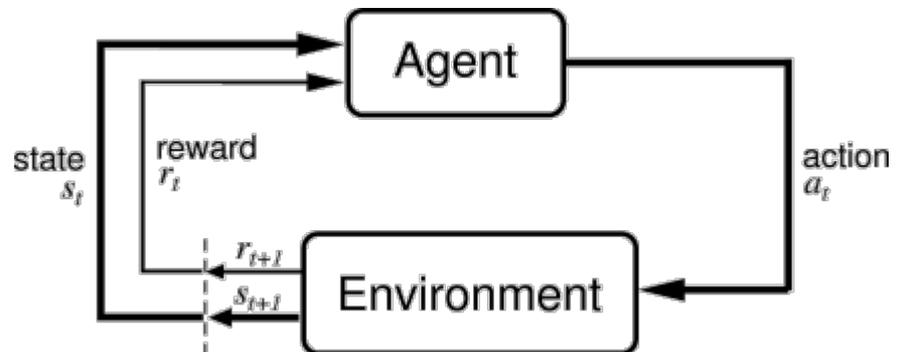


Algorithme d'apprentissage

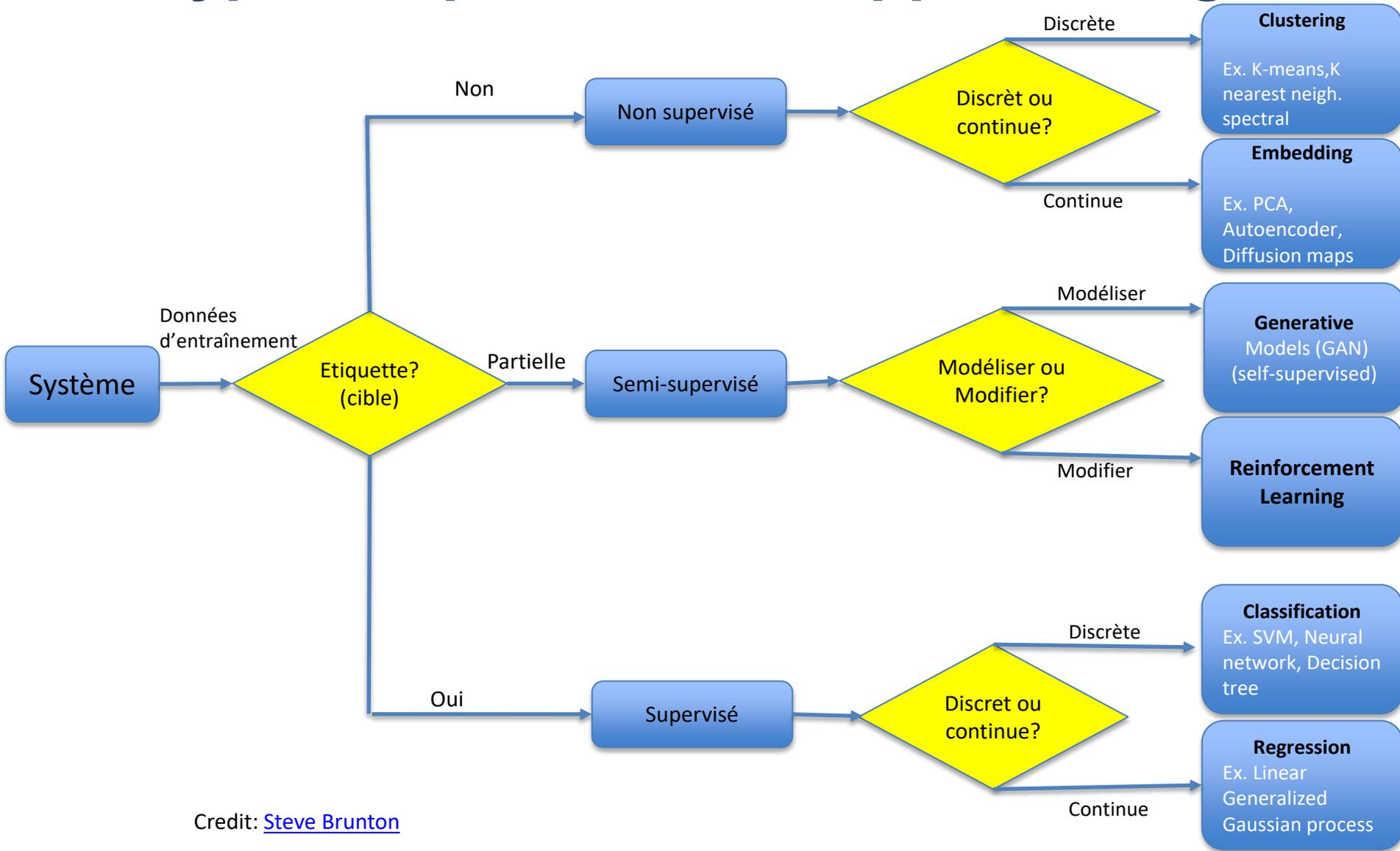


- Groupe 1 (chats)
- Groupe 2 (canards)
- Groupe 3 (chiens)

Apprentissage par renforcement



Types de problèmes d'apprentissage



Credit: [Steve Brunton](#)

APPRENTISSAGE SUPERVISÉE

Motivation – Classification d'images

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 81 68
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 43 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 57 08 30 03 49 13 36 65
92 70 95 23 04 60 11 42 69 81 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 03 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 31 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 32 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
09 36 68 87 57 62 20 72 03 46 33 67 46 55 12 52 63 93 53 69
04 42 16 73 36 65 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 63 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 84 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 21 67 58

What the computer sees

image classification →

- 82% cat
- 15% dog
- 2% hat
- 1% mug

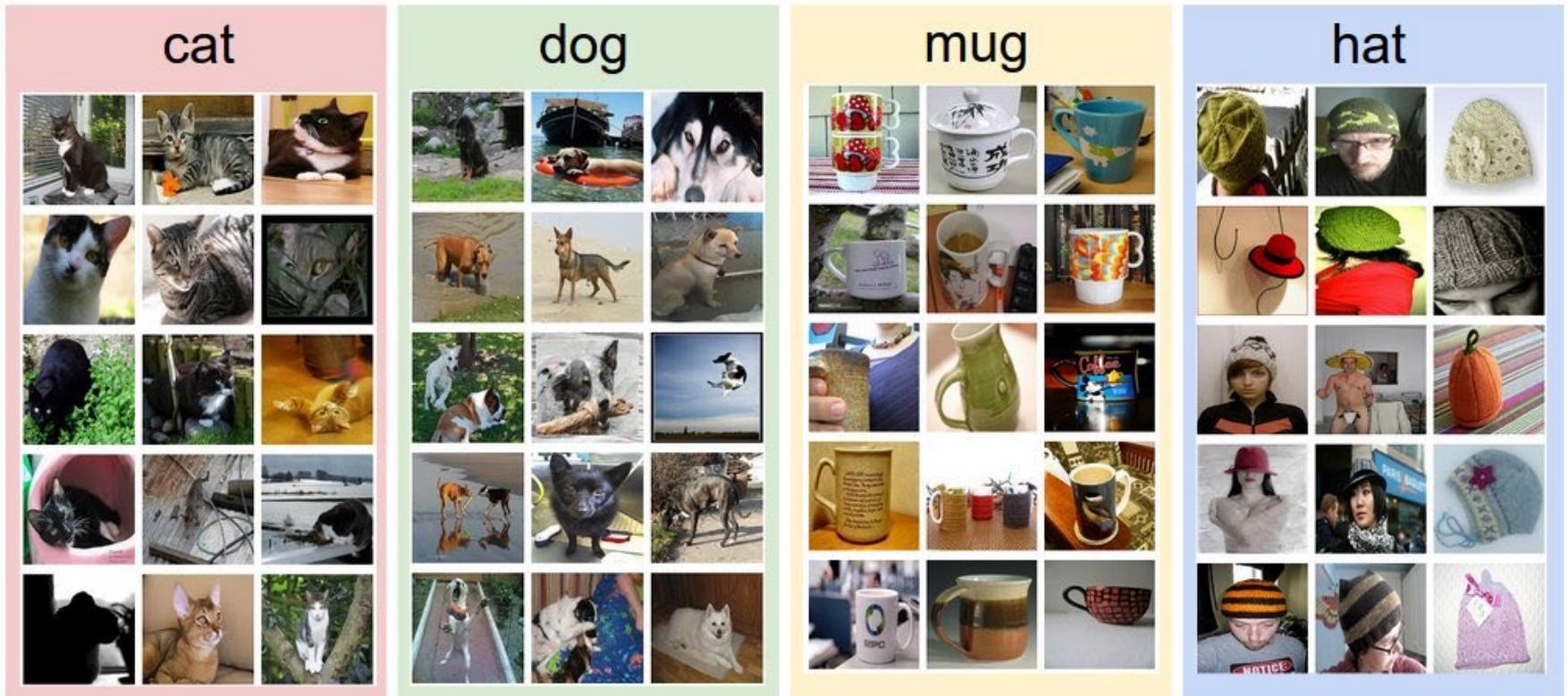
<http://cs231n.github.io/assets/classify.png>

Motivation – Classification d'images

Principales étapes

- **Éntrée** : N images, chacune étiquetée par l'une des K classes. C'est l'ensemble de **données d'entraînement**.
- **Apprentissage** : On utilise les données d'entraînement pour apprendre à quoi ressemble chacune des classes. C.à-d., on entraîne un classifieur, autrement dit, on **apprend le modèle**.
- **Évaluation**: Finalement, on évalue la qualité du classifieur en lui demandant de prédire les étiquettes pour un ensemble d'images vues pour la première fois.

Exemple 1 – Classification d'images



<http://cs231n.github.io/assets/trainset.jpg>

Exemple 2 – Reconnaissance de caractères

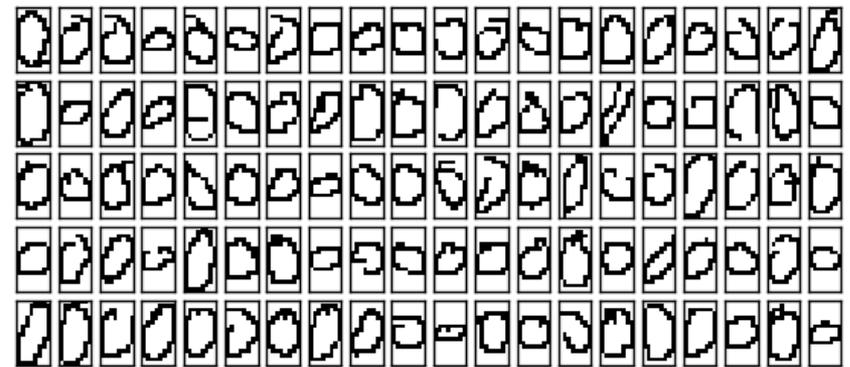
- Reconnaissance de caractères manuscrits

Ensemble d'entraînement

(100 exemples d'apprentissage par classe)



Classe 'e'



Classe 'o'

APPRENTISSAGE SUPERVISÉ : FORMULATION DU PROBLÈME

Apprentissage supervisé

Un problème d'apprentissage supervisé est formulé comme suit.

Étant donné un ensemble d'entraînement de N exemples:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \} D$$

où chaque y_j a été généré par une fonction inconnue $y = f(\mathbf{x})$,

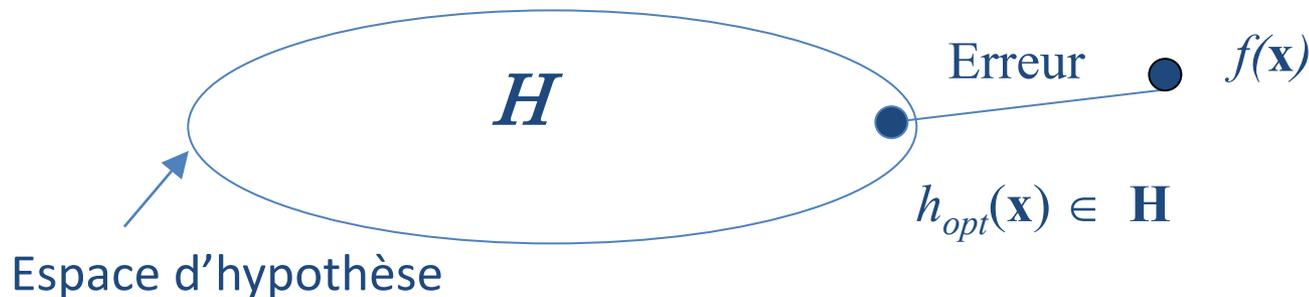
Trouve une nouvelle fonction h (modèle ou hypothèse)

qui sera une bonne approximation de f (c'est à dire $f(\mathbf{x}) \approx h(\mathbf{x})$) »



Apprentissage supervisé

- Étant donné un ensemble de données d'entraînement, l'apprentissage est un problème de **recherche** de l'hypothèse h **dans un espace d'hypothèses** H , tel que h minimise la **distance** à $f(x)$



- Les données sont souvent bruitées et disponibles en quantité limitée. Il y a donc une variation dans les données et dans les modèles (représentations).
- L'erreur dépend de la qualité des données d'entraînements et de la méthode utilisée pour sélectionner/chercher la bonne hypothèse

Apprentissage supervisé

- Données : **ensemble d'entraînement** de N exemples:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \} D \quad \text{avec} \quad y = f(\mathbf{x})$$

- Problème : trouver $h(x)$ tel que $f(\mathbf{x}) \approx h(\mathbf{x})$
- Un algorithme d'apprentissage peut donc être vu comme étant une fonction A à laquelle on donne un ensemble d'entraînement et qui donne en retour la fonction h

$$A(D) = h$$

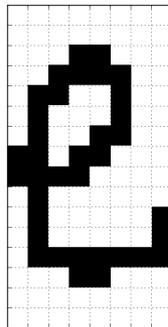
Modèles et approche pour chercher la fonction h

- Dépendamment des approches d'apprentissage, la fonction h peut être représentée de différente manière.
- Dans ce cours, nous voyons:
 - ◆ K plus proches voisins
 - ◆ Perceptron
 - ◆ Régression logistique
 - ◆ Réseau de neurones
 - ◆ Arbre de décision
- Le livre couvre différentes autres approches

REPRÉSENTATION DES DONNÉES

Représentation des données

- L'**entrée** \mathbf{X} est représentée par un vecteur de valeurs d'attributs réels (représentation factorisée)
 - ◆ ex.: une image est représentée par un vecteur contenant la valeur de chacun des pixels



```
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  
        0.,  0.,  0.,  0.,  0.,  0.,  1.,  1.,  0.,  0.,  0.,  0.,  0.,  
        1.,  1.,  1.,  1.,  0.,  0.,  0.,  1.,  1.,  0.,  0.,  1.,  0.,  
        0.,  0.,  1.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  1.,  0.,  0.,  
        1.,  1.,  0.,  0.,  1.,  1.,  0.,  1.,  1.,  0.,  0.,  0.,  1.,  
        1.,  1.,  1.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  
        0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  1.,  0.,  
        0.,  0.,  0.,  0.,  1.,  0.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  
        0.,  0.,  0.,  1.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.]
```

- La **sortie désirée** ou **cible** y aura une représentation différente selon le problème à résoudre:
 - ◆ problème de classification en C classes: valeur discrète (index de 0 à $C-1$)
 - ◆ problème de régression: valeur réelle ou continue

K PLUS PROCHES VOISINS

Illustration: 3 plus proches voisins

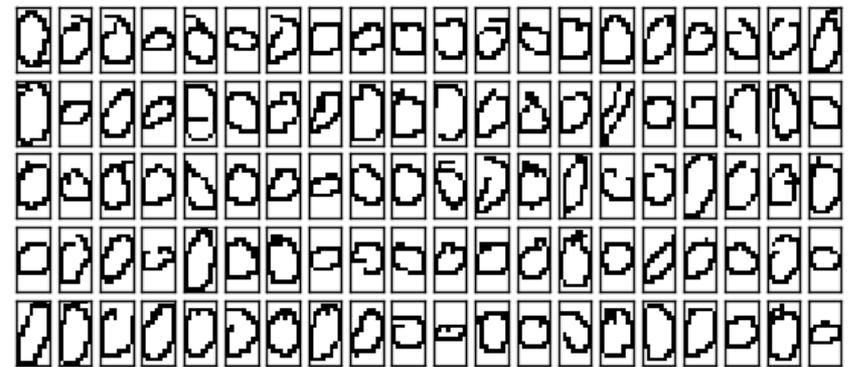
- Reconnaissance de caractère: est-ce un 'e' ou un 'o'?

Ensemble d'entraînement

(100 exemples d'apprentissage par classe)



Classe 'e'



Classe 'o'

Illustration: 3 plus proches voisins

- Reconnaissance de caractère: est-ce un 'e' ou un 'o'?

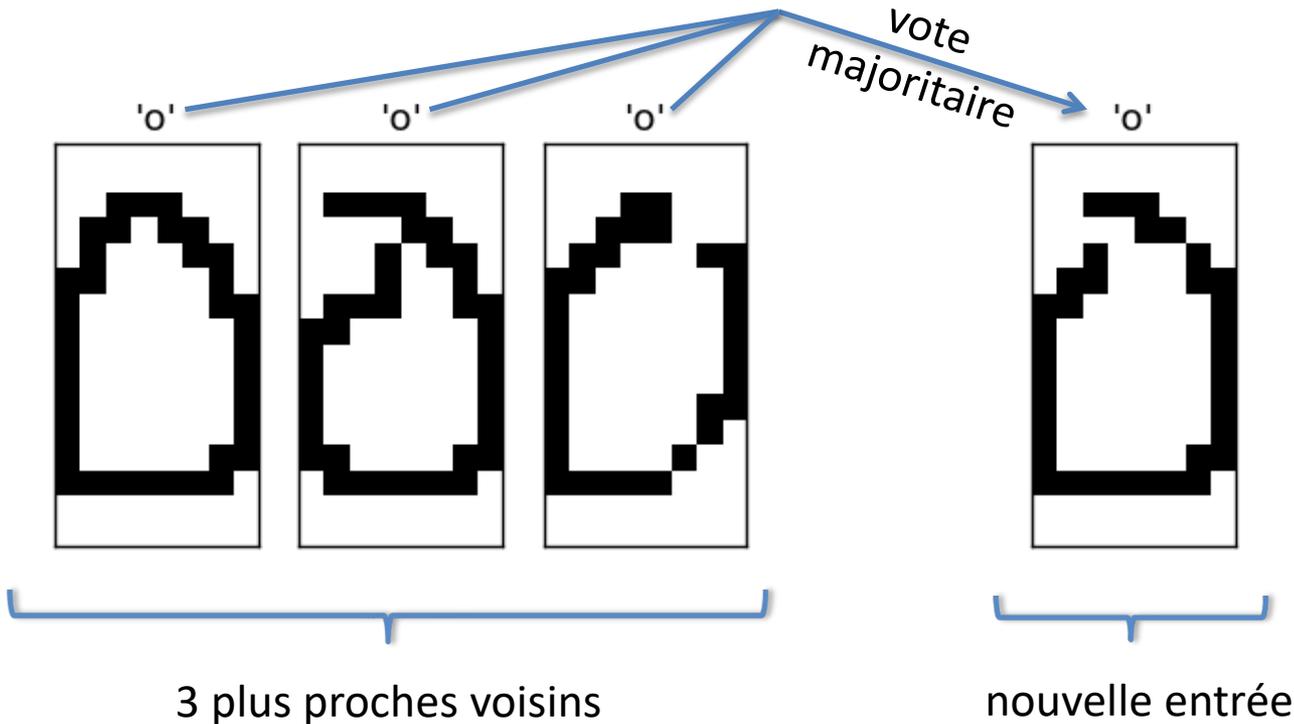


Illustration: 3 plus proches voisins

- Reconnaissance de caractère: est-ce un 'e' ou un 'o'?

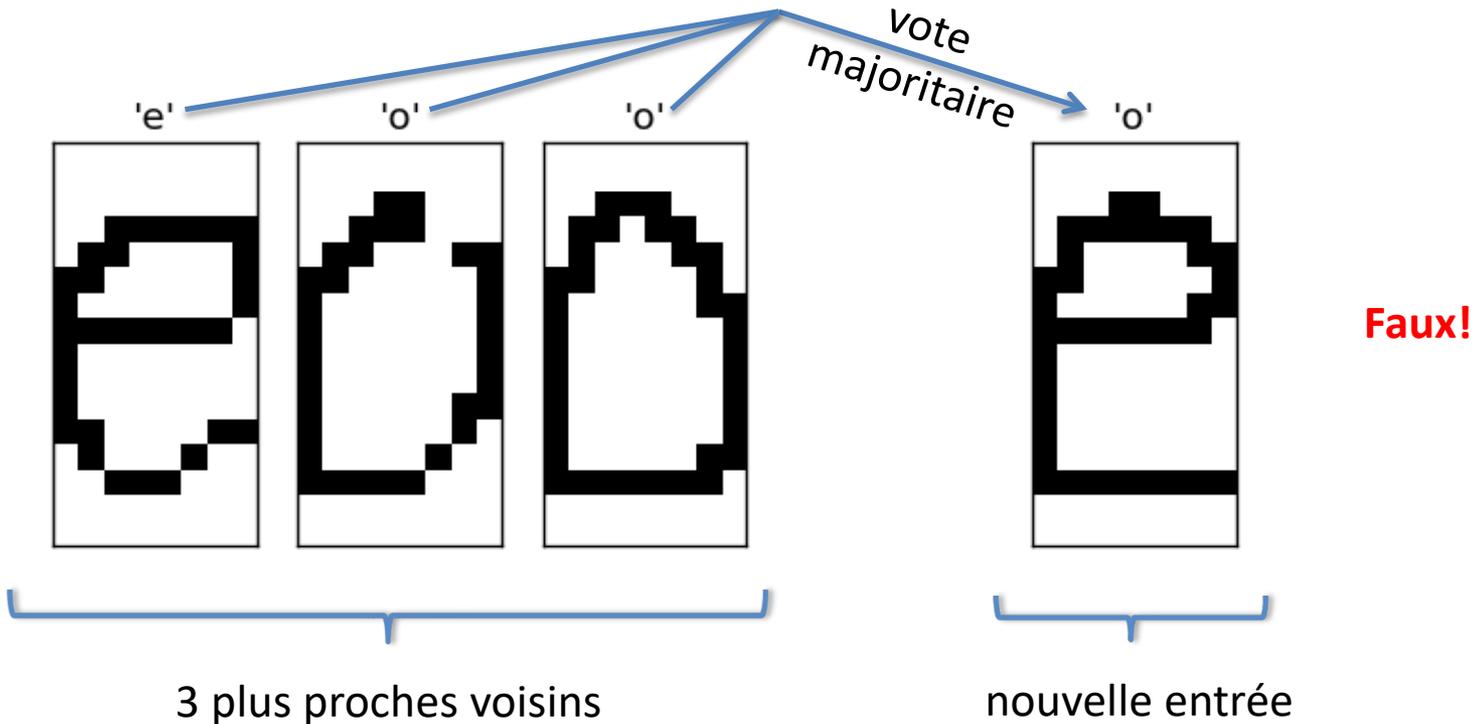
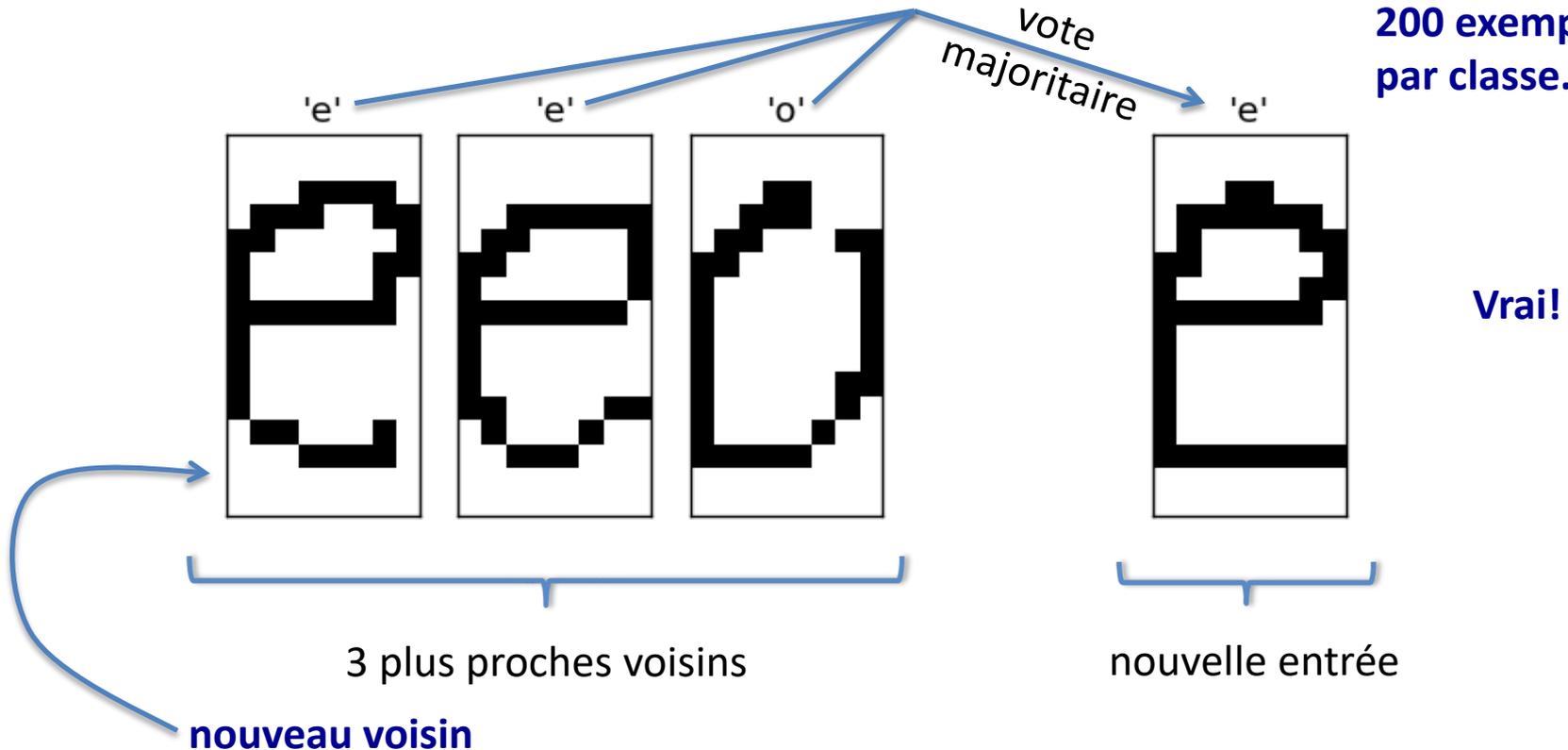


Illustration: 3 plus proches voisins

- Reconnaissance de caractère: est-ce un 'e' ou un 'o'?

Si on ajoute
200 exemples
par classe...



Exemple: classifieur k plus proches voisins

- Possiblement l'algorithme d'apprentissage de classification le plus simple
- **Idée:** étant donnée une entrée \mathbf{x}
 1. trouver les k entrées \mathbf{x}_t parmi les exemples d'apprentissage qui sont les plus « proches » de \mathbf{x}
 2. faire voter chacune de ces entrées pour leur classe associée y_t
 3. retourner la classe majoritaire
- Le succès de cet algorithme va dépendre de deux facteurs
 - ◆ la quantité de données d'entraînement (plus il y en a, meilleure sera la performance)
 - ◆ la qualité de la mesure de distance (est-ce que deux entrées jugées similaires sont de la même classe?)
 - » en pratique, on utilise souvent la distance Euclidienne:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_k (x_{1,k} - x_{2,k})^2}$$

\mathbf{X} = vecteur
 x = scalaire

Rappel - Apprentissage supervisé

- Un problème d'apprentissage supervisé est formulé comme suit:
« Étant donné un **ensemble d'entraînement** de N exemples:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \} D$$

où chaque y_j a été généré par une **fonction inconnue** $y = f(\mathbf{x})$,
découvrir une nouvelle fonction h (**modèle** ou **hypothèse**)
qui sera une bonne approximation de f (c'est à dire $f(\mathbf{x}) \approx h(\mathbf{x})$) »

- Un algorithme d'apprentissage peut donc être vu comme étant une fonction A à laquelle on donne un ensemble d'entraînement et qui donne en retour cette fonction h

$$A(D) = h$$

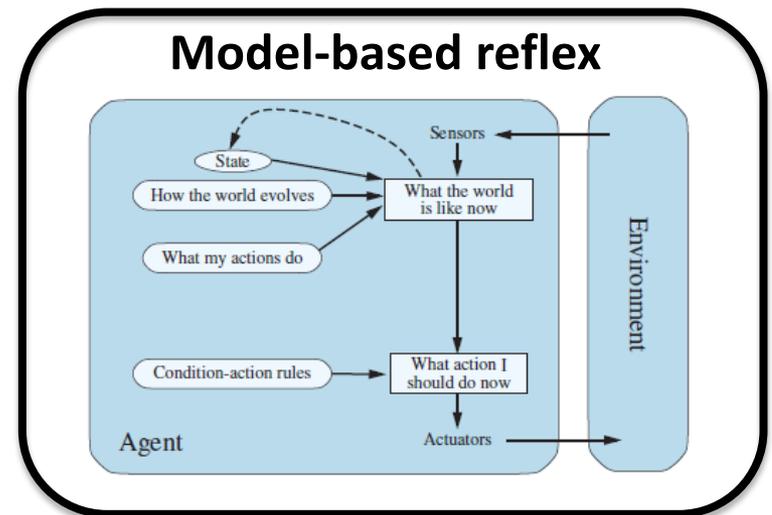
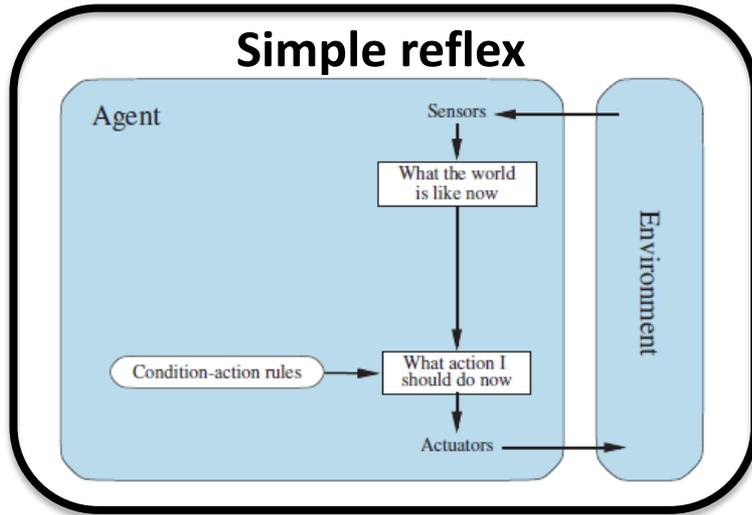
Retour sur classifieur k plus proches voisins

- Dans le cas de l'algorithme k plus proches voisins:
 - ◆ A est un programme qui produit lui-même un programme, soit celui qui fait une prédiction à l'aide de la procédure k plus proches voisins
 - ◆ $h = A(D)$ est le programme qui fait voter les k plus proches voisins dans D d'une entrée donnée
 - ◆ $h(\mathbf{x})$ est la sortie du programme pour l'entrée \mathbf{x} , c'est à dire une prédiction de la classe de \mathbf{x}
 - ◆ f est la « fonction » qui a généré nos données d'entraînement
 - ex.: l'être humain qui a étiqueté les images de caractères
- On peut démontrer que plus D est grand, plus h sera une bonne approximation de f
 - ◆ intuition: en augmentant la taille de l'ensemble d'entraînement, les k plus proches voisins ne peuvent changer qu'en étant encore plus proches (plus similaires) à l'entrée

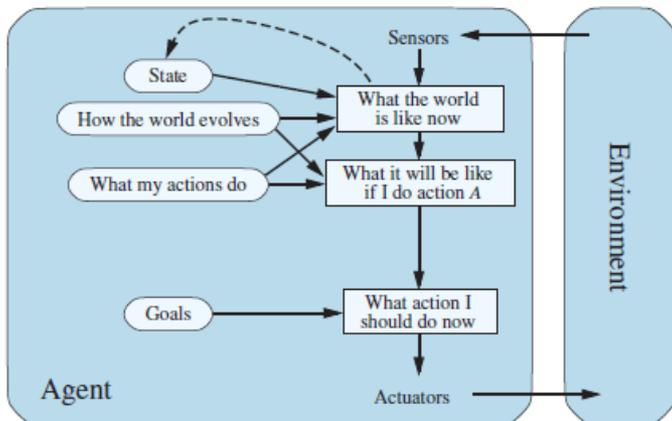
Mesure de la performance d'un algorithme d'apprentissage

- Comment évaluer le succès d'un algorithme?
 - ◆ on pourrait regarder l'erreur moyenne commise sur les exemples d'entraînement, mais cette erreur sera nécessairement optimiste
 - » h a déjà vu la bonne réponse pour ces exemples!
 - » on mesure donc seulement la capacité de l'algorithme à **mémoriser**
- Ce qui nous intéresse vraiment, c'est la capacité de l'algorithme à **généraliser** sur de **nouveaux exemples**
 - ◆ ça reflète mieux le contexte dans lequel on va utiliser h
- Pour mesurer la généralisation, on met de côté des exemples étiquetés, qui seront utilisés seulement à la toute fin, pour calculer l'erreur
 - ◆ on l'appelle l'**ensemble de test**
 - ◆ on y reviendra de façon plus approfondie plus tard

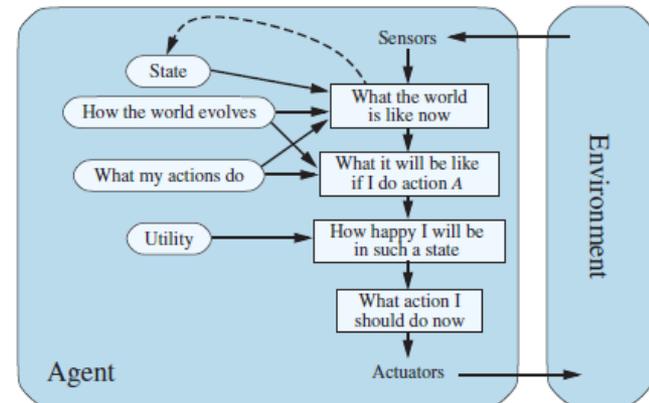
Algorithme des K plus proches voisins pour quel type d'agent?



Goal-based



Utility-based



Vous devriez être capable de...

- Nommer les trois formes d'apprentissage: supervisé, non supervisé, par renforcement
- Expliquer et simuler l'algorithme des K plus proches voisins