Apprentissage à base d'exemples

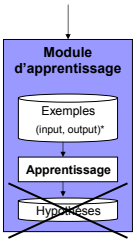
IFT-17587

Concepts avancés pour systèmes intelligents

Luc Lamontagne

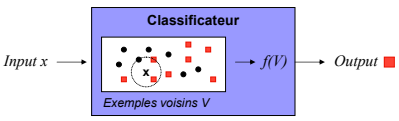
Apprentissage à base d'exemples

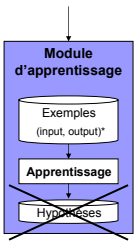
- Approche habituelle en apprentissage
  - On généralise une hypothèse à partir d'une banque d'exemples.
- *kNN* – les *k* plus proches voisins
  - On ne construit pas d'hypothèse.
  - On emmagasine les *N* exemples.
  - Lorsqu'on a une nouvelle instance à classifier
    - On prend la décision à partir de *k* exemples similaires.
- Plusieurs noms dans la littérature :
  - *Instance-based learning*
  - *Memory-based learning*
  - *Case-based reasoning* (plus générale)



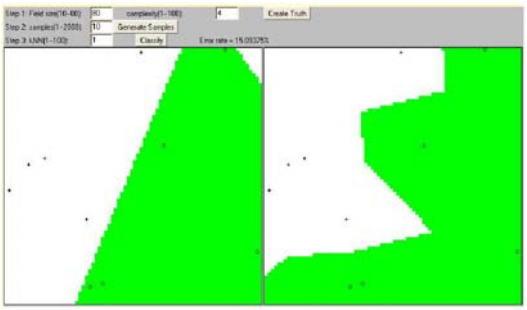
Apprentissage à base d'exemples

- Approche *kNN* - Étant donné une nouvelle instance à classifier :
  - Identifier les *k* exemples les plus près de l'instance
    - Métrique de distance ou de similarité.
  - Déterminer la catégorie ou la valeur à partir de ces exemples.
- Avantage:
  - Estimation locale pour chaque instance à classer.
- Désavantage:
  - Coût de classification élevé.





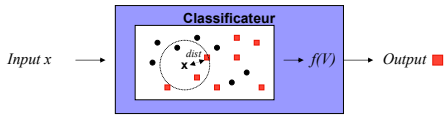
Apprentissage à base d'exemples :
Partitionnement de l'espace de classification



Apprentissage à base d'exemples :

Identifier les *k* plus proches voisins

- Les instances sont des points dans un espace à d-dimensions
  - d est le nombre d'attributs.
- Une instance x_i est définie par son vecteur d'attributs
  $$<a_1(x_i), a_2(x_i), \ldots, a_d(x_i)>$$
  - Chaque instance à également une catégorie v_i.
- Identifier les voisins les plus proches de x_i
  - Trouver les k instances ayant la plus petite distance $dist(x_i, x_j)$
  - Similarité → une fonction inverse de la distance



Apprentissage à base d'exemples :

Distance des *k* plus proches voisins

- Mesures souvent utilisées pour la distance $dist(x_i, x_j)$
  - La distance euclidienne (valeurs continues)
    $$dist(x_i, x_j) = \sqrt{\sum_{r=1}^{d} (a_r(x_i) - a_r(x_j))^2}$$
  - La distance de *Manhattan* (valeurs continues)
    $$dist(x_i, x_j) = \sum_{r=1}^{d} |a_r(x_i) - a_r(x_j)|$$
  - La distance de *Hamming* (valeurs discrètes)
    $$dist(x_i, x_j) = \#\{r \in d : a_r(x_i) \neq a_r(x_j)\}$$

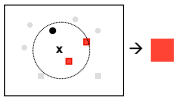
Apprentissage à base d'exemples :

Déterminer la catégorie d'une instance

- Pour les fonctions à valeurs discrètes
  - L'ensemble V contient les catégories possibles.
  - Par exemple $V = \{v, f\}$ ou $V = \{faible, moyen, élevé\}$
- L'estimation de la fonction $f(x_i)$
  - Vote sur la valeur qui revient le plus souvent parmi les k voisins.

$$f(x_i) = \max_{v \in V} \sum_{j=1}^{k} \delta(v, f(x_j))$$

$$\text{où } \delta(a,b) = \begin{cases} 1 \text{ si } a = b \\ 0 \text{ sinon} \end{cases}$$

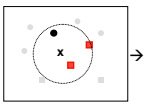


Apprentissage à base d'exemples :

Déterminer la valeur d'une instance

- Fonction à valeurs continues – régression
  - L'ensemble V est défini sur un intervalle de valeurs.
  - Par exemple le prix d'une maison, la priorité d'une tâche, l'orientation d'un objet...
  - On doit estimer le résultat sur une échelle de valeur continue.
- Retourne la moyenne des valeurs des k plus proches voisins.

$$f(x_i) = \frac{\sum_{j=1}^{k} f(x_j)}{k}$$



Apprentissage à base d'exemples :

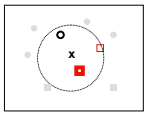
kNN avec distances pondérées

- Amélioration possible
  - Pondérer la contribution de chacun des *k* voisins
    - Dépend de la distance par rapport à l'instance à classer.
- Pour les valeurs discrètes, on obtient:

$$f(x_i) = \max_{v \in V} \sum_{j=1}^{k} w_j \delta(v, f(x_j)) \text{ où } w_j = \frac{1}{dist(x_i, x_j)^2}$$

- Pour les valeurs continues, on obtient:

$$f(x_i) = \frac{\sum_{j=1}^{k} w_j f(x_j)}{\sum_{j=1}^{k} w_j} \text{ où } w_j = \frac{1}{dist(x_i, x_j)^2}$$



Apprentissage à base d'exemples :

Importance des attributs

- Certains attributs plus important que d'autres.
  - Par exemple, pour la détection de pourriels (*spams)*
    - Les instances sont représentées par un ensemble un mots.
    - Les catégories sont $V = \{v, f\}$.
  - Certains mots jouent un rôle plus important dans le concept de spam.
    - Ex. : *Free, Cash, Amazing, Stuff, $, Stock, Pick, Drugs, Viagra*…
- Solution : donner un poids à chacun des attributs.
  - On donne plus de poids w_r aux attributs plus importants pour la classification.
  - Par exemple, pour la distance euclidienne :

$$dist(x_i, x_j) = \sqrt{\sum_{r}^{d} w_r \left(a_r(x_i) - a_r(x_j)\right)}$$

Apprentissage à base d'exemples :

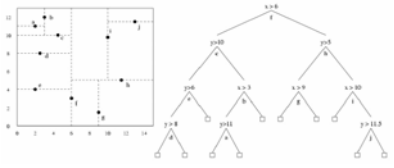
En pratique

- Approche simple et efficace.
  - Complexité = $O(NA)$ où N est le # d'instances et A le # d'attributs.
- *kNN* est robuste aux erreurs dans les exemples d'entraînement.
- Quelques points à considérer :
  - Définition de la distance pour un attribut.
    - Propre au domaine d'application.
      - Quel est la distance entre 23ºC et 31ºC ?
      - Quel est la distance entre les couleurs bleu et vert ?
      - Quel est la distance entre les mots *money* et *cash* ?
  - Comment déterminer les poids des attributs ?
    - Peut être fait en optimisant sur une banque d'instances d'entraînement
    - Par exemple, faire une validation croisée avec un algorithme génétique.
  - Maintenance de la base d'exemples
    - Combien d'exemples garder en mémoire ?
- Forme particulière du raisonnement à base de cas.

Apprentissage à base d'exemples :

Réduction de la complexité

- Un balayage séquentiel des exemples est parfois trop coûteux.
- Partitionner la banque d'exemples à l'aide d'un arbre
  - On fragmente la banque en plus petites partitions.
    - *kd-trees* : construire un arbre binaire dont les partitions ont moins de b instances.
    - *ID3* : induction d'un arbre de décision comme au chapitre 18.
  - Les exemples sont dans les feuilles de l'arbre.
  - On peut consulter plus d'une feuille pour trouver les exemples les plus similaires.



Apprentissage à base d'exemples :

Maintenance des exemples

- Combien d'exemples garder en mémoire ?
  - Trop d'exemples → le temps de calcul augmente.
  - Peu d'exemples → la qualité de la solution diminue.
- Il faut donc contrôler le nombre d'exemples du système.
- Quelques approches possibles :
  - Exploiter les exemples pendant une période de temps et éliminer ceux qui :
    - ont été les moins utilisés;
    - n'ont pas été utilisés récemment;
    - sont les moins utiles (mais il faut déterminer l'utilité…).
  - Comprimer la base d'exemples pour éliminer les cas qui n'apportent pas de contribution significative.
    - Idée : éliminer les exemples redondants ou très similaires.

Maintenance des exemples :

Comprimer la banque d'exemples

function Construct_Compact_CB (*Cases*), **returns** a case base
inputs: *Cases*, all the examples originally used of the system.
locals: *New_CB*, the result of this function, a subset of the original cases, initially ← {}.
Threshold, a cutoff threshold for forgetting cases.
Changes ← true;

while *Changes* **do**
Changes ← false;
for each case *C* **in** *Cases* // *On détermine la pertinence de garder chacun des exemples.*
if Max-Similarity (*C*, New_CB) < *Threshold* **then** // s'il n'y a pas d'exemples similaires à C.
Changes ← true;
Add *C* to *New_CB*; // on l'ajoute dans la banque.
Remove *C* from *Cases*;
return *New_CB*

function Max-Similarity (*C*, *Cases*) **returns** a similarity value
Sim = 0.0;
for each case *C'* **in** *Cases* do
if *C*.V = *C'*.V; // Si la valeur est la même pour les deux cas -- catégorie discrète
Sim ← Max(*Sim*,Get-Similarity(*C*, *C'*));
return *Sim*;

Raisonnement à base de cas (CBR)

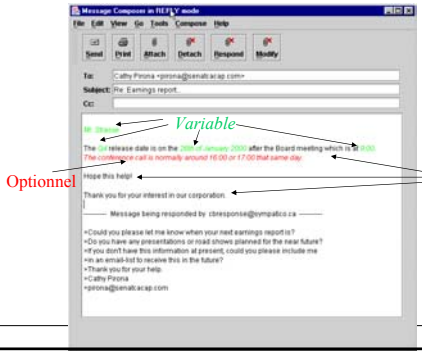
- Approche de résolution de problèmes.
  - Extension de l'apprentissage à base d'exemples.
- Utilisation d'expériences passées pour résoudre de nouveaux problèmes.
  - Cas = problème + solution + résultat*
  - La solution n'est pas nécessairement une catégorie.
    - Ex 1 : la solution est un plan d'actions → schedule de cours.
    - Ex 2 : la solution est un texte → document de jurisprudence en droit.
- Raisonnement de type *search and adapt*
  - *Search* → rechercher des cas similaires dans une base d'exemples.
  - *Adapt* → modifier la solution pour l'adapter au contexte du nouveau problème.
- 3 grandes familles de systèmes CBR
  - CBR structurel : un cas est une instance structurée tel un objet ou enregistrement BD.
  - CBR conversationnel : un cas est un ensemble de questions-réponses.
  - CBR textuel : exploite le contenu textuel des cas.

Raisonnement à base de cas :

Exemple – Réponse au courriel

Raisonnement à base de cas :

Exemple – Réponse au courriel



Raisonnement à base de cas :

Schéma général