Apprentissage non supervisé

IFT-17587
Concepts avancés pour systèmes intelligents
Luc Lamontagne

Plan de la présentation

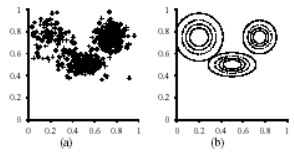
- Apprentissage non supervisé - *Clustering*
  - Approche hiérarchique
  - *k-means*
  - *Expectation maximization*

Apprentissage non supervisé

- Une forme d'apprentissage effectuée à partir uniquement des données brutes.
- Aucune rétroaction sur :
  - La classification d'une instance d'entraînement;
  - Le résultat de l'apprentissage.
- Très utile pour des applications comme la recherche d'informations
  - Impossible de classifier manuellement tout le contenu du web.
  - Le grand nombre de documents permet de faire de l'analyse exploratoire de leur contenu (*exploratory data analysis*).
    - Tenter de comprendre les caractéristiques de base d'un phénomène.
  - Tenter de généraliser certains aspects du contenu.

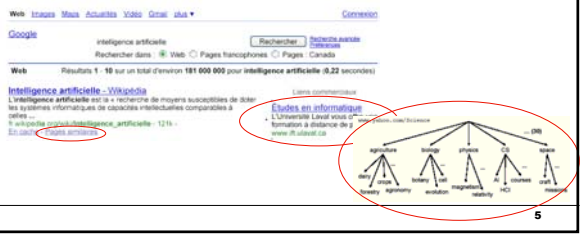
Clustering

- La forme la plus répandue d'apprentissage non supervisé.
- Regroupement d'instances ayant des traits communs
  - **Utile pour identifier des tendances dans les données.**
  - Pour dégager des thèmes communs dans des documents.



Clustering

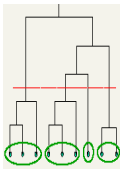
- La forme la plus répandue d'apprentissage non supervisé
- Regroupement d'instances ayant des traits communs
  - Utile pour identifier des tendances dans les données.
  - **Pour dégager des thèmes communs dans des documents**.
    - Par exemple, les suggestions de Google ou la hiérarchie de Yahoo!



Clustering :

Approche hiérarchique

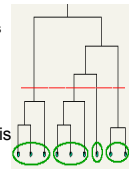
- Approche vers le haut (*bottom-up*)
  - On démarre avec chaque instance assigné à un groupe individuel.
  - On regroupe progressivement les groupes 2 par 2.
    - On choisi toujours les 2 éléments les plus proches.
      - On a besoin d'une fonction de distance ou de similarité.
      - Similarité → maximise, distance → minimise.
    - On peut regrouper
      - Deux instances;
      - Une instance et un sous-groupe;
      - Deux sous-groupes.
  - On arrête lorsqu'on n'a plus qu'un seul groupe.
  - Les données sont structurées sous forme d'arbre (dendogramme).
  - On coupe l'arbre à la hauteur qui nous permet d'obtenir les nombre de regroupements désirés.



Clustering :

Approche hiérarchique

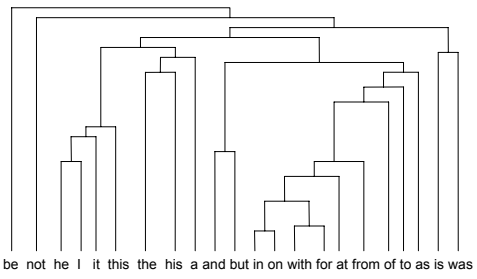
- Il existe une version vers le bas (*top-down)*
  - On partitionne nos instances en 2 sous-groupes.
    - Il faut trouver la partition qui maximise la distance entre les deux groupes.
  - On répète jusqu'à ce que chaque sous-groupe soit un singleton.
- Difficile à réaliser en pratique – problème calculatoire.
  - L'approche *bottom-up* est habituellement préférée.
- Cependant la signification des sous-groupes est parfois plus facile à interpréter.



Clustering :

Approche hiérarchique

- Exemple de *clustering* de mots fréquents en anglais



Clustering :

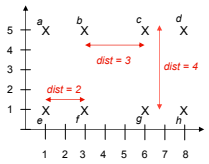
Approche hiérarchique

- Différentes fonctions de similarité possibles pour le *clustering* hiérarchiques de sous-groupes.
- On fait le calcule pour chaque paire de membres des deux sous-groupes.
- Options
  - *Single link* : similarité des deux membres les plus similaires.
  - *Complete link* : similarité des deux membres les moins similaires.
  - Moyenne : la moyenne des similarités entre les membres.

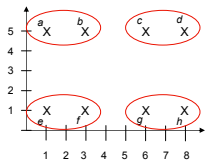
Clustering :

Approche hiérarchique

- Exemple -- *Single link* vs. *Complete link*



1. Instances initiales

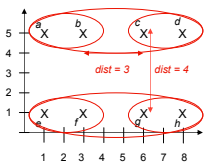


2. Regroupements intermédiaires

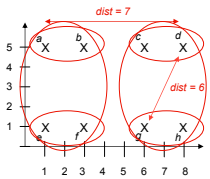
Clustering :

Approche hiérarchique

- *Single link* vs. *Complete link*



3. Regroupement *single link*



4. Regroupement *complete link*

Clustering :

Approche k-Means

- Nous avons des instances à regrouper
  - k est le nombre de groupe que l'on veut créer (*clusters*)
  - On adopte une structure linéaire → pas d'arbre.
  - Idée générale : on défini un groupe par son centre de masse.
- Approche :
  1. Assigner arbitrairement chaque instance à un des k groupes
     - Peut-être fait aléatoirement.
  2. Calculer la moyenne de chaque groupe (centroïde)
     - Pourrait être aussi la médiane…

$$\mu = \frac{1}{M}\sum_{x \in C} x \text{ où } M = |C|$$

  3. Déplacer chaque instance dans le groupe dont la moyenne est la plus près.

$$c_j = \{x_i \mid \forall \mu_l \,,\, dist(x_i, \mu_j) \le dist(x_i, \mu_l)\}$$

  4. Recommencer les étapes 2 et 3 quelques fois.
     - On arrête lorsque les déplacements d'instances se stabilisent.

Clustering :

Exemple de *k-Means*

1. Initialisation et calcul de la moyenne

2. Déplacement vers le groupe le plus près

3. Mise à jour de la moyenne

4. Déplacement en fonction de la moyenne…

Clustering :

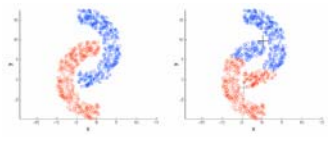
Approche k-Means

- Pour mettre en pratique, on a besoin de :
  - Déterminer le nombre de groupe *k* pour notre application,
  - Une fonction pour calculer la similarité (ou la distance) entre une instance et la moyenne.
    - Similaire à ce qu'on fait en *kNN*
  - Une fonction de calcul de la moyenne.
- Complexité
  - *O(# instances . # groupes . # attributs . # itérations)*
  - Le nombre de groupes et d'attributs sont constants.
  - L'algorithme converge habituellement assez rapidement.
  - Donc *O(N)*.

Clustering :

Approche k-Means

- Limitations :
  - Lorsqu'on ne peut pas déterminer le nombre de groupe au préalable.
  - Difficilement applicable lorsque l'espace ne peut pas facilement être partitionné par des formes circulaires.



Original Points K-means (2 Clusters)

Clustering :

Expectation Maximization (EM)

- Dans *k-Means*, les instances appartiennent à un seul sous-groupe.
  - *Hard clustering* (~ regroupement stricte)
- Dans EM, elles appartiennent à plusieurs d'entre eux (*Soft clustering*)
  - Cependant le degré d'appartenance varie d'un groupe à l'autre.
  - Peut être interprété comme la probabilité que l'instance x_i appartiennent au groupe C_j, c.-à-d. $P(C_j|x_i)$
- Comme pour *k-Means*, on itère sur deux étapes :
  - *Étape E* : Déterminer le lien entre chaque instance et chaque groupe.
    - On met à jour l'appartenance h_{ij} de chaque instance en fonction de la moyenne précédente du groupe.

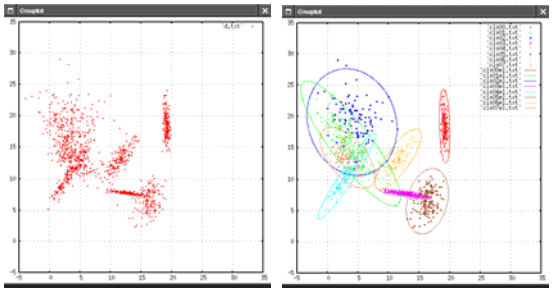
$$h_{ij} = \frac{sim(x_i, \mu_j)}{\sum_{l=1}^{k} sim(x_i, \mu_l)}$$

  - *Étape M* : Calculer la moyenne du groupe
    - Mise à jour de l'estimation à partir des nouvelles appartenances au groupe.

$$\mu_j = \frac{\sum_{i=1}^{n} h_{ij} x_i}{\sum_{i=1}^{n} h_{ij}}$$

Clustering :

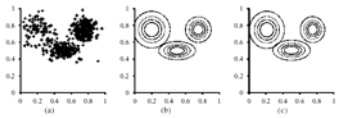
Expectation Maximization



Clustering :

Expectation Maximization (EM)

- On rencontre cette approche dans plusieurs applications.
  - Traitement de la langue → traduction automatique
  - Astronomie → catégorisation d'étoiles.
- Recette générale qui est utile pour estimer les paramètres d'une variable cachée
  - Des variables du modèle de données ne peuvent pas être observées.
- Idée générale :
  - Nous avons une distribution de données à partir de laquelle des exemples sont générés.
  - Cette distribution a k composantes (une mixture).
  - Les paramètres de chacune des composantes C_j ne sont pas connue à priori.
  - On alterne les étapes pour estimer cette.
    - *Étape E* : Estimer la probabilité $p_{ij} = P(C_j|x_i)$, i.e que l'instance x_i soit générée par la composante C_j
    - *Étape M* : Calculer les paramètres de la distribution de la composante C_j qui maximisent la vraisemblance des données (*loglikelihood*)



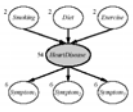
Clustering – Expectation maximization :

EM et réseaux bayesiens

- On peut utiliser EM pour apprendre les probabilités conditionnelles d'un réseau bayesien.
- Par exemple
  - Nous avons une banque de N exemples qui nous permet d'observer :
    - Les évidences : *smoking, diet, exercise*
    - Les symptômes : $symptom_1 \ldots$
  - Cependant il est impossible de connaître la valeur de la variable *HeartDisease* (cachée) pour ces exemples.
- Approche *EM* :
  - Au début, on suppose qu'on connaît les probabilités du réseau.
    - On détermine arbitrairement $P(HeartDisease|e)$.
  - Pour tous les exemples :
    - Calculer la probabilité de chacune des valeurs de *HeartDisease*

$$\theta_i = \sum_{j=1}^{N} P\left(HeartDisease = val_i \middle| exemple_j\right) \Big/ N$$

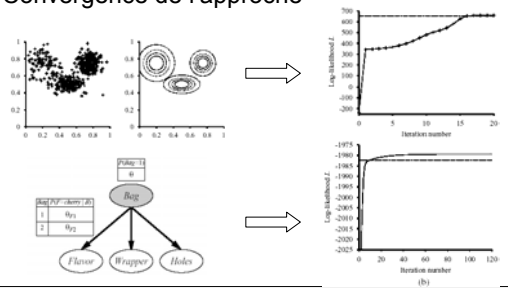
    - Mettre à jour les probabilités conditionnelles du réseau.
    - Utilisation du réseau pour faire ces calculs.
  - Répéter quelques itérations jusqu'à convergence.
    - Mais il faut limiter le nombre d'itérations pour éviter un surentraînement.



Clustering – Expectation maximization :

EM et réseaux bayesiens

- Convergence de l'approche



Clustering:

Conclusion

- L'approche hiérarchique
  - Est préférable pour une analyse détaillée de données.
  - Donne plus d'information que l'approche non hiérarchique.
  - A un niveau de complexité plus élevé que l'approche non hiérarchique
    - *Bottom-up* → $O(N^2)$
    - *Top-down* → $O(N^3)$
- L'approche non hiérarchique
  - Est préférable si :
    - l'efficacité est une considération
    - On a un très grand volume de données.
  - *k-Means* est la méthode la plus simple.
    - Bon de l'utiliser en premier (parfois suffisant).
  - *k-Means* suppose que les données proviennent d'un espace euclidien.
    - Pas toujours adéquat pour plusieurs jeux de données (ex. couleurs).
  - *EM* est plus flexible
    - Peut prendre en compte différentes distributions de probabilités.
  - *EM* est utilisée en pratique pour apprendre les paramètres des réseaux bayesiens, des modèles markoviens, des processus temporels…