

Systemes de recommandation

2e partie

Raphaël Fournier-S'niehotta

CNAM Paris, fournier@cnam.fr

HTT-FOD
RCP217
2020-2021

le **cnam**

Plan

3 | Recommandation par le contenu

4 | Systèmes de recommandation hybrides

5 | Recommandation sociale

6 | Évaluation des systèmes de recommandation

7 | Thématiques avancées

Recommandation par le contenu

Recommandation Content-based

- Situations où l'on dispose de caractéristiques descriptives pour les éléments
- Exemple : genre de films, artistes d'une chanson (parolier, arrangeur, mélodiste, etc.), etc.
- Les seules notes de l'utilisateur suffisent!
- Particulièrement utile pour les éléments nouveaux (peu de notes)
- similarité avec ce que les utilisateurs ont déjà apprécié

Deux sources de données :

- les méta-données
- un profil utilisateur

Étapes

1. Pré-traitement des éléments

- Représentation vectorielle avec mots-clefs
- Très dépendant du domaine
- Poids, structure
- Cf Recherche d'Information (NFE204)
- Feature selection (Gini, χ^2 , entropie)

2. Modélisation des profils

- Combinaison de feedback et d'attributs
- Souvent proches des techniques classiques de classement ou régression
- Pour chaque utilisateur, un modèle est entraîné ("documents étiquetés")
- Nearest-neighbors, Classifieur Bayésien naïf, Règles d'associations, ou régression (linéaire, par ex.)

3. Filtrage / recommandation

Avantages, inconvénients

Avantages

- Les autres ne comptent plus!
- Cold start pour les éléments
- Explications grâce aux caractéristiques des éléments
- Les classifieurs textuels abondent, facilité d'implémentation et d'intégration dans des systèmes existants

Inconvénients

- Sur-spécialisation : enfermement dans des "bulles", absence de nouveauté
- Cold-start utilisateur
- Pour les utilisateurs, il faut un certain nombre de notes pour éviter le sur-apprentissage

Systemes de recommandation hybrides

Recommandation hybride

- Les systèmes de filtrage par le contenu et de filtrage collaboratif ont des performances avantages et inconvénients divers
- Des sources de données diverses
- Combiner plusieurs systèmes est presque une évidence
- On cherche à compenser les faiblesses d'un système par un autre
- On parle de **systemes hybrides**

Recommandation hybride

Il y a plusieurs façons de construire des systèmes hybrides :

- L'approche "ensembliste", où on combine les résultats de plusieurs systèmes sans les modifier (un content-based et un CF, par exemple). Utilisée traditionnellement en ML.
- L'approche monolithique, dans laquelle on tente de modifier des algorithmes existants avec des caractéristiques d'autres systèmes (un collaborative filtering avec prise en compte d'attributs)

On peut combiner des systèmes de même type (exemple : plusieurs content-based). Les systèmes ayant gagné le Netflix prize combinaient des approches de voisinage et de facteurs latents.

Recommandation hybride

Il y a également différentes manières de combiner :

- approche "weighted" : pondération des scores données par plusieurs systèmes
- approche "switching" : selon les possibilités et les besoins, on utilise l'un ou l'autre des systèmes de reco mobilisé (exemple : éviter le cold-start, ou choisir le système avec la meilleure accuracy)
- approche cascade : un système de reco raffine les recommandations proposées par un autre (biais dans l'apprentissage), avant une sortie unifiée
- approche "feature augmentation" : les résultats d'un système servent d'entrée (supplémentaire) pour le suivant
- approche meta : le modèle d'un système sert d'entrée à un autre.
Exemple : content-based pour améliorer la sélection de voisinage en CF.

Recommandation sociale

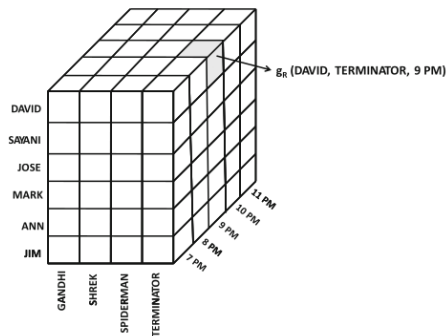
Recommandation sociale

"Social" a de nombreux sens dans le domaine de la recommandation, différents et importants :

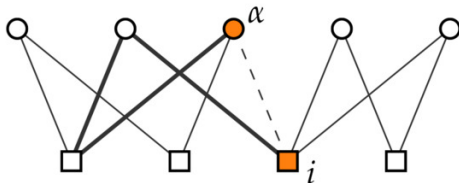
- Le contexte social de l'utilisateur :
 - Temps : les recommandations devraient être différentes selon le contexte temporel (heure de la journée, semaine de l'année, jour de la semaine, etc.)
 - Lieu : les recommandations gagnent en précision en tenant compte du lieu (voyage, achats, etc.)
 - Entourage : cinéma (film différent en famille, avec des amis)
- Le "réseau social" :
 - Autorité, importance (PageRank)
 - Prédiction de liens dans un graphe (biparti)
 - Influence
 - Réseau de confiance

Contexte social

- Modèle multidimensionnel (OLAP), très riche
- La requête "quels items pour quel utilisateur" est généralisée :
 - quel "item-moment" pour quel utilisateur
 - quel "utilisateur-moment" pour quel item
- Traitement "en tranches"



Prédiction de liens



- Internet, le Web, et les réseaux sociaux ont donné une structure forte à de nombreuses activités humaines
- faire de la recommandation peut être vu comme "chercher à prévoir les liens **pertinent** pour le réseau... mais pas encore apparus
- dans certains cas, comme Facebook, susciter des liens est même central pour la plateforme

Prédiction de liens

Plusieurs indicateurs peuvent être calculés pour estimer la probabilité qu'un lien apparaisse :

- *Common neighbors* : nombre de voisins en communs

$$CN(i,j) = |N_i \cap N_j|$$

non normalisé (problème des grands degrés)

- Indice Jaccard (normalisé pour les degrés de i et j) :

$$\text{Jacc}(i,j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

mais dans les voisinages, les degrés peuvent être forts

- Adamic-Adar (version pondérée de CN) :

$$\text{AdamicAdar}(i,j) = \sum_{k \in N_i \cap N_j} \frac{1}{\log(N_k)}$$

(base log peu importante)

Prédiction de liens

Quand il y a peu de liens, on peut s'en remettre à des mesures reposant sur les chemins dans le graphe :

- Mesure de centralité (Katz)
- Mesure avec marches aléatoire (personalized PageRank)

On peut aussi approcher la prédiction de liens comme un problème de classification (supervisée) :

- Création d'un dataset avec de features (similarités diverses, degrés, etc.)
- La présence/absence d'un lien est une variable binaire
- Une régression logistique, modifiée pour tenir compte du déséquilibre de classes, peut être utilisée

Enfin, on peut approcher le problème avec de la factorisation de matrices.

Influence

- Dans un réseau d'interactions humaines, le bouche à oreille permet la propagation de messages
- Certains messages se diffusent plus (vite) que d'autres (cascades)
- Analogie avec l'épidémiologie, les réseaux informatiques (diffusion)
- Les facteurs les plus importants pour la diffusion :
 - La centralité de l'acteur
 - La force des liens (amitiés longues, statut social)
- Problème de recherche : *influence propagation models*.
On cherche un ensemble de nœuds S (de taille k) tel que la propagation de l'information soit maximisée dans le réseau (par influence)
 - Modèle à seuil
 - Modèle à propagation unique

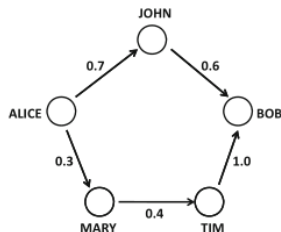
Confiance

- La recommandation algorithmique peut tenter d'utiliser, si possible, la structure du réseau social "amical" des utilisateurs
- Dans la vie hors ligne, les gens demandent à leurs amis des conseils, pour aller au cinéma, voyager, etc.
- **Homophilie** : les utilisateurs sont généralement connectés à d'autres utilisateurs ayant les mêmes goûts
- Cette homophilie rend les recommandations de proches plus fiables, a priori, que celles de n'importe qui.
- Ce "réseau social" n'est pas toujours existant, mais peut aider face au cold-start.
- On peut essayer d'inférer la confiance entre utilisateurs, à partir des liens existants. Il existe des plateformes où l'on demande explicitement aux utilisateurs de dire à quels utilisateurs ils font confiance (Epinions).

La confiance peut permettre de trouver/modifier les peer-groups à partir desquels on calcule les moyennes pour prédire les notes.

Confiance

- En général, comme il y a peu de notes, il y a peu de liens de confiance.
- On doit se fonder sur la **propagation** et l'**agrégation** de confiance.
- Cela repose sur la transitivité :
 - A fait confiance à B, B fait confiance à C, alors A fait confiance à C.
 - Il faut identifier les chemins de A vers C et "propager la confiance"
 - Multiplication de valeurs, avec facteurs d'amortissement
 - Et/ou : agrégation (moyenne, somme)
- nombreux algos de propagation/agrégation (MoleTrust, TidalTrust, TrustWalker)
 - influence des chemins courts
 - marche aléatoire
- gestion de la "distrust" (confiance négative)



Bilan de la recommandation sociale

Avantages

- Amélioration de la qualité des recommandations proposées
- Notamment : les utilisateurs "controversé", ceux qui ne sont pas d'accord avec les autres
- Cold start : si on dispose du réseau social mais pas de notes, on peut proposer des recommandations aux nouveaux utilisateurs
- Résistance aux attaques : la recommandation repose sur des liens "fiables", il est plus dur de piéger le système avec des "faux profils" (Amazon), puisque les faux profils ne sont en général pas liés aux utilisateurs réels.

Inconvénients

- Difficulté d'avoir des données

Évaluation des systèmes de recommandation

Évaluation de systèmes de recommandation

- Similitude avec l'évaluation de la classification/régression
- Toutefois, aspect "pratique" : multi-facettes
- Il faut un processus clair pour avoir la "bonne" évaluation
- Méthodes "online" (à chaud)
 - A/B testing (taux de conversion)
 - Complexité dans les grands systèmes
- Méthodes "offline" (à froid)
 - Datasets larges, connus, robustes
- Études utilisateurs (user studies)

Études utilisateurs

- Des sujets d'études sont recrutés, doivent accomplir des tâches précises
- Retours collectés avant et après, et résultat de l'interaction
- Notes, classement, appréciations générales

Avantages

- Divers scénarios / interfaces testables
- Précision du retour

Inconvénients

- Utilisateurs averti que "c'est un test" (biais)
- Difficulté à recruter / tester de nombreux utilisateurs (représentativité de l'échantillon)

Évaluation à chaud

- Comme des études, mais en général avec "les vrais utilisateurs" du système en place
- Moins de biais (échantillon)
- Souvent employé pour comparer divers algorithmes
- A/B testing :
 - On prend plusieurs (deux) échantillons de même tailles/profils
 - Chaque algo est testé sur un échantillon
 - Un algo est souvent "une modification de l'ancien" et l'autre algo est "celui en place"
- On compare souvent avec un "taux de conversion" (clic sur des articles, par exemple)
- Similitude avec les essais cliniques en médecine (essai randomisé contrôlé, ou randomized controlled trial ou RCT), avec "groupe contrôle"
- La segmentation peut ne pas être stricte (un utilisateur reçoit A ou B aléatoirement)

Évaluation à chaud

Avantages

- Qualité des observations
- Lien avec apprentissage par renforcement et multi-armed bandits (optimisation gloutonne)

Inconvénients

- Il faut en général beaucoup d'utilisateurs dans le système pour que ça marche
- Il faut la main sur le système pour mettre en place les tests
 - Évaluation académique (reproductibilité) limitée
 - Robustesse / généralisabilité réduites (manque de tests)

Évaluation à froid

- On utilise les données compilées sur un système
- Notes, user/item id, horodatage éventuel (peu de détails)
- En général, ça peut devenir un standard (réutilisation académique/industrie) :
 - Netflix Prize
 - MovieLens
 - LastFm, Flixster, etc.

Objectifs de l'évaluation

- Accuracy (justesse, pertinence)
 - métriques de régression, classification : MSE/RMSE
- Coverage (couverture)
 - sparsity
 - peut-être artificiellement réduite (prédiction par défaut)
 - % des utilisateurs/items pour lesquels on peut prédire k éléments
- Nouveauté
 - recommandations non vues avant / non connues
- "Sérendipité" (nouveau ET "moins évident")
- Diversité
 - Listes "un peu" variées (sinon, risque d'échec)
 - Améliore nouveauté et sérendipité
- Robustesse (résister aux attaques)
- Scalabilité (temps de calcul)

Justesse

- MSE

$$MSE = \frac{\sum_{(u,j) \in E} (\hat{r}_{uj} - r_{uj})^2}{|E|}$$

- Plus c'est bas, moins il y a d'erreur, mieux c'est
- RMSE : ramené en terme de notes.
- (R)MSE : pénalisé par grosses erreurs!
- MAE : Mean Absolute Error

$$MAE = \frac{\sum_{(u,j) \in E} |\hat{r}_{uj} - r_{uj}|}{|E|}$$

- Version normalisée entre 0 et 1 (en divisant par $r_{max} - r_{min}$)

Évaluation des classements

- **Corrélation de classements**
 - Pearson
 - Kendall (+1/-1 si même ordre, sommés pour toutes paires d'items pour u)
- **Utilité : corriger l'impact du "bas du classement"**
 - (N)DCG : Discounted Cumulative Gain :

$$DCG = \frac{1}{m} \sum_{u=1}^m \sum_{j \in I_u} \frac{2^{rel_{uj}} - 1}{\log_2(v_j + 1)}$$

rel_{uj} : pertinence (note brute)

- MAP@k (Mean Average Precision) : fraction d'éléments pertinents pour une liste de k éléments pour chaque utilisateur, moyennée
 - R-score : $Rs = \sum_{j \in I_u} \frac{\max\{r_{uj} - C_u, 0\}}{2^{(v_j - 1)/\alpha}}$ avec v_j rang de j dans I_u
composantes note et classement. Peut être limité à L éléments
- **AUC : Area Under (Receiver Operating) Curve**
 - évaluations binaires

Thématiques avancées

Sujets à approfondir

- Prise en compte élaborée du temps et du lieu
- Recommandation de groupe
- Résistance aux attaques
- Gestion élaborée de la vie privée

Références

Remerciements

Ce livre repose notamment sur :

- le livre *Recommender systems, the textbook*, de Charu C. Aggarwal. Springer, 2016.

Version 1.0 du 2021-05-25

