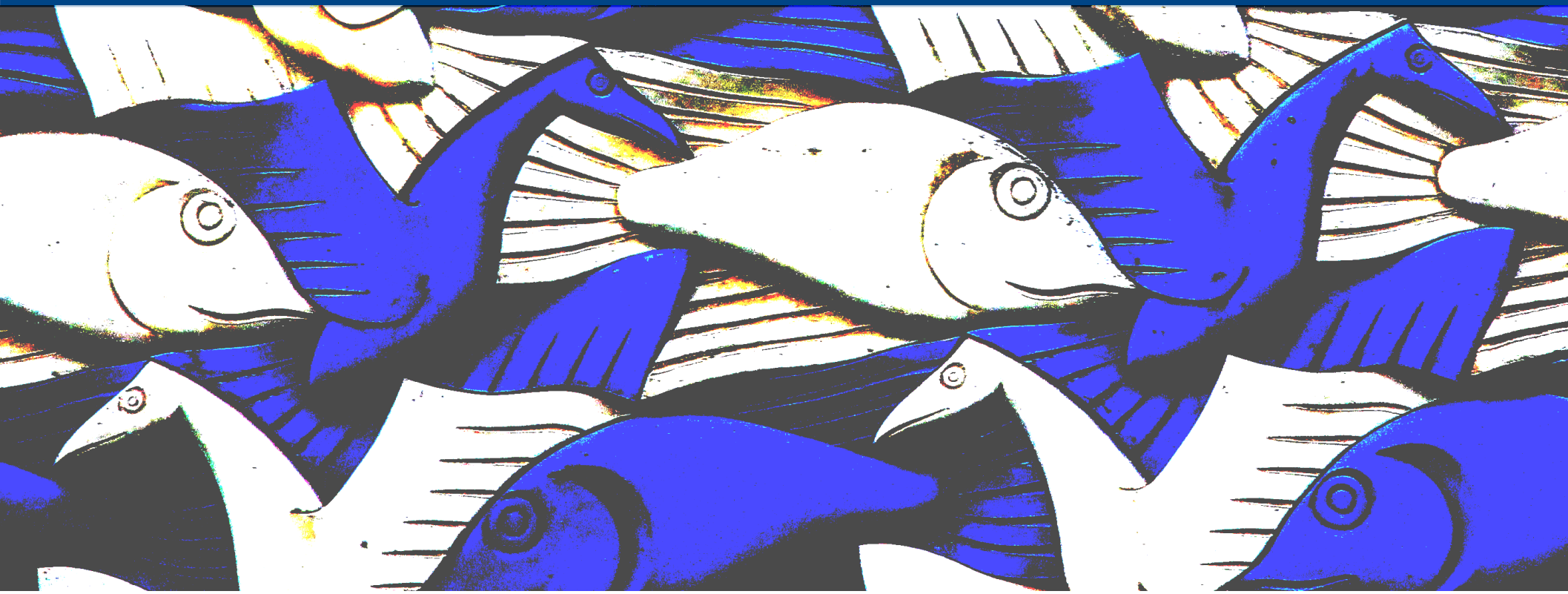
Intelligence collective et apprentissage



Raisonnement et science de la décision

Master M1 MIAGE – Ingénierie Métier

Université Toulouse 1 Capitole

Umberto Grandi

Intelligence collective

Un mot très utilisé et avec une interprétation souvent un peu large...

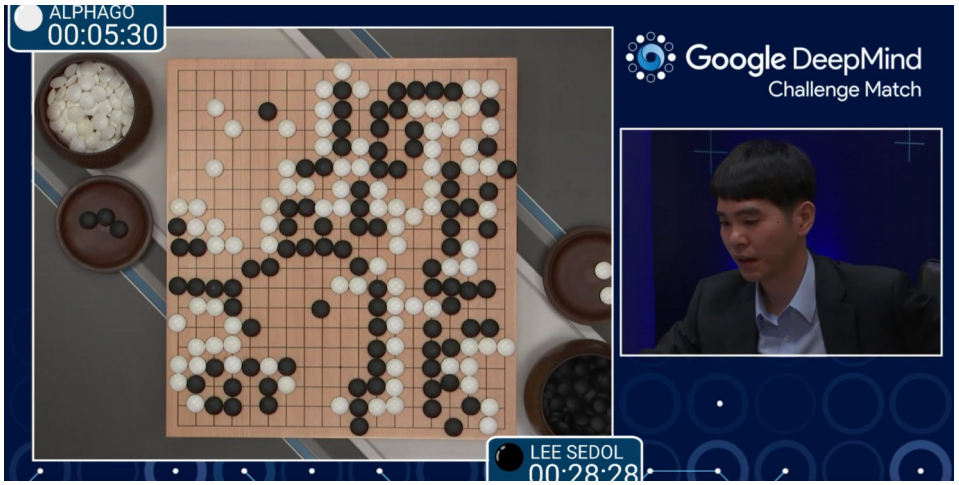
On parle des méthodes d'intelligence collective quand on utilise la combinaison des préférences, choix, comportements et idées d'un groupe de personnes pour **améliorer notre compréhension d'un phénomène**. La récente disponibilité des **données** a rendu ces techniques très puissantes :

1) *Wikipedia* est une des encyclopédie la plus vaste et précise, crée seulement à partir des contributions des utilisateurs.

2) *Google* avec *PageRank* utilise les informations que les créateurs de sites WEB fournissent sur leur pages (les liens) pour générer son classement.

Apprentissage automatique

Un autre mot à la mode, récemment en combinaison avec le mot « deep »...



L'apprentissage automatique est un secteur de l'intelligence artificielle qui étudie des algorithmes capable d'apprendre. Cela a permis des progrès technologiques incroyables dans les dernières années :

- *Spam filters* : des logiciels qui apprennent à partir des exemples à reconnaître le spam dans vos boite mail.
- *Reconnaissance des visages* : tag automatisé sur FB...
- *AlphaGo* de *Google DeepMind* a gagné en 2016 au jeu de Go contre le champion du monde Lee Sedol.
- ...

Exemple : filtre pour le spam

Supposons que vous recevez souvent du spam avec les mots « pharmacie en ligne ». Rapidement vous **généraliserez** vos observations dans une règle « si une mail contient ces mots alors va directement dans la poubelle ».

Vous avez crée un modèle mentale qui réponde partiellement à la question « qu'est-ce que le spam ? »

Un **algorithme d'apprentissage** doit être capable de créer des règles à partir des données :

- Certains techniques (*arbres des décisions*...) sont transparents dans leurs réponses ,en nous donnant des raisons et des modèles en leur support.

- D'autres, comme les *réseaux de neurones,* sont des « black box » : ils répondent aux questions mais c'est difficile de comprendre le modèle sous-jacente.

Exemple : système de recommandation

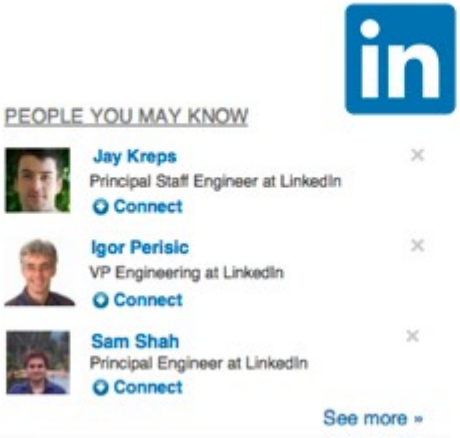
Dans ce cours on apprendra comment construire un simple système de recommandation :

1) Comment construire la matrice d'évaluation ou matrice *item/user*

2) Deux mesures de similarité : la corrélation euclidienne et de Pearson

3) Deux approches à la recommandation : algorithmes *user-based* et *item-based*

Multiples applications



Collaborative filtering

Supposons qu'on veut découvrir des nouvelles chansons à écouter...

On demandera à nos amis quelles chansons ils aiment, et de nous recommander des titres... mais pas à n'importe qui! Nous savons bien lesquels entre nos amis ont **le même goût musicale que nous**.

La clé pour construire un **système automatique de recommandation** est donc trouver une réponse à la question:
Qu'est-ce que le goût ?

Répresenter les préférences

Tout part de l'**observation du comportement** des utilisateurs, c'est à dire leurs **préférences**. Plusieurs possibilités pour les représenter:

- Évaluations : entre 0 et 5 (*Tripadvisor, Amazon*…)
- Achat en ligne: 1 si acheté, 1 si regardé, 0 si pas acheté
- *Facebook* (vieille version) : 1 si j'aime, 0 si je n'aime pas

Je représente les évaluations des utilisateurs dans une matrice :

	I Shot the Sheriff	*Mother's little helper*	*Hells Bells*	*Single Ladies*
Benoit	3			1
Julien		2.5	2	4.5
Valérie	2		2	4

Matrice utilisateurs/produits

Cette matrice s'appelle **matrice utilisateurs/produits** (user/items) :

- *Users= {Benoit, Julien, Valérie}*
- *Items = {I Shot the Sheriff, Moter's Little Helper, Hells Bells, Single Ladies}*

	I Shot the Sheriff	*Mother's little helper*	*Hells Bells*	*Single Ladies*
Benoit	3			1
Julien		2.5	2	4.5
Valérie	2		2	4

Observez que la matrice est **creuse**: la plupart des utilisateurs n'ont pas donné d'avis sur un grande nombre de produits.

Mais c'est normale : Amazon vends environs 480 millions de produits et a 240 millions d'utilisateurs !!

Similarité euclidienne des utilisateurs

Comment décider si deux utilisateurs ont le même goût ?

Corrélation euclidienne entre Julien et Valérie

Dans la matrice utilisateurs/produits :

- Chercher les évaluations en commun entre le deux utilisateurs
- Construire le vecteur d'évaluation de Julien sur ces produits *JULIEN=(a1, b1, …, z1)* et celui de Valérie VALERIE=*(a2, b2, …, z2)*
- Calculer la distance euclidienne entre ces deux vecteurs :

$$Euclid(JULIEN, VALERIE) = \sqrt{(a1 - a2)^2 + \cdots + (z1 - z2)^2}$$

Peut-on visualiser la similarité sur un graphique ?

Similarité euclidienne des utilisateurs

Au tableau...

Problème : la distance euclidienne est **dépendant de l'échelle** :

Deux utilisateurs avec échelles différentes mais goûts similaires ont une haute distance euclidienne : ex. Julien=(3,1,2) et Valérie=(4,2,3) on le même goût mais Valérie utilise une échelle plus haute.

Similarité de Pearson

Une solution possible au problème de la dépendance de l'échelle est d'utiliser la corrélation de Pearson

Similarité de Pearson entre Julien et Valérie

Dans la matrice utilisateurs/produits :

- Chercher les évaluations en commun entre le deux utilisateurs
- Construire le vecteur d'évaluation de Julien sur ces items *JULIEN=(a1, b1, …, z1)* et celui de Valérie *VALERIE=(a2, b2, …, z2)*
- Calculer les deux sommes : *S1=a1+ … +z1* et *S2=a2+ … +z2*
- Calculer la sommes des carrés : $Sq1=a1^2+ \ldots +z1^2$ et $Sq2=a2^2+ \ldots +z2^2$
- Calculer la somme des produits *Ptot=a1* a2 + … + z1* z2*
- Calculer la similarité de Pearson (n est la longueur des vecteurs):

$$Pearson(\ JULIEN, VALERIE) = \frac{Ptot - \frac{S1 \times S2}{n}}{\sqrt{[Sq1 - \frac{S1^2}{n}] \times [Sq2 - \frac{S2^2}{n}]}}$$

Est-il possible de donner une interprétation graphique à cette mesure ?

Trouver les utilisateurs plus proche

Une fois choisi une mesure de similarité, je peux trouver les utilisateurs plus similaires à moi pour avoir des recommandations

Algorithme de recommandation par les utilisateurs les plus proches

- Pour donner des recommandations à Julien :
- Calculer les similarités (euclidienne, Pearson, ou autres...) entre Julien et tous les autres utilisateurs
- Trouver l'utilisateur (ou les 3 ou 5 premiers) à similarité maximale
- Recommander à Julien leurs chansons favorites

Ceci n'est **pas un bon méthode**:

- Si les utilisateurs plus similaires ont écouté plus ou moins les mêmes chansons que moi ? Risque des **recommandation redondantes**
- Si l'utilisateur plus similaire me recommande bizarrement des chanson que tous les autres utilisateurs n'ont pas aimé ?

Recommandation des produits

Construction de la table de recommandation de Benoit pour deux chansons qu'il n'a pas encore écouté (*Back in Black* et *London Calling*) :

	Similarité	*Back in Black*	*Sim. x Back in Black*	*London Calling*	*Sim. x London Calling*
Julien	0,99	3	2,97	2,5	*2,48*
Valérie	0,39	3	1,14		
Leila	0.92			5	4,6
Total			4,14		7,08
SimSum			1,38		1,91
Tot/SimSum			3		3,74

Algorithme de recommandation *user-based*:

- Extraire de la matrice utilis./produit les colonnes relatives au deux chansons
- Multiplier les évaluations d'une personne par sa similarité à Benoit
- Pour chaque chanson, calculer la somme des évaluations pesés, et la diviser par la somme des valeurs positives des similarités des utilisateurs qui ont exprimé une évaluation (*pourquoi on fait cette division?*)
- Choisir la chanson avec le total le plus élevé

Similarité des produits

Pourquoi calculer la similarité des utilisateurs et pas celle des produits ?

Algorithme de recommandation par les produits plus proches

- Sélectionner un produit *PROD1* (par exemple celui qui vient d'être acheté)
- Sélectionne un deuxième produit *PROD2*, et trouver tous utilisateurs qui ont évalué tous les deux produits *PROD1* et *PROD2*
- Construire le vecteur d'évaluation de ces deux produits *PROD1=(a1, b1, …, z1)* et PROD2=*(a2, b2, …, z2)*
- Calculer la similarité (Euclidien, Pearson…) de ce deux vecteurs
- Répéter pour tous produits, et recommander les 3 ou 5 avec max similarité

Utilisé par *Amazon* par exemple. Très utile pour le marketing...

La pratique (*item-based*)

Les algorithmes qu'on a vu utilisent tous la matrices utilisateurs/produits pour donner des recommandation : faisable pour des milliers des produits, mais lent.

Algorithme de recommandation item-based

- *(Offline)* Construire la matrice de similarité produits/produits
- *(Offline)* Pour chaque produit PROD, enregistrer les N produits les plus similaires à PROD et leurs similarités
- *(Real-time)* Un utilisateurs se connecte : extraire son vecteur d'évaluations
- *(Real-time)* Créer une matrice avec les produits déjà évaluées comme lignes et ceux sans évaluations comme colonnes
- *(Real-time)* Calculer le score de recommandation normalisé (voir table) pour chacun des produits pas évalués
- *(Real-time)* Recommander ceux avec le score le plus élevé

Hypothèse à la base de cet algorithme : la similarité entre produits change **moins souvent** que celle entre utilisateurs

Item-based (exemple)

Chansons pas évalués

	Évaluation	*Back in Black (similarité)*	*Back in Black (éval * sim)*	*London Calling (similarité)*	*London Calling (éval * sim)*
Hell Bells	4	0,4	1,5	0,25	1
Single Ladies	1	-0,6	-0,6	-0,1	-0,1
Smoke on the water	3,5	0,2	0,7	0,34	1,19
Total		1,2	1,6	0,69	2,09
Normalisé			1,33		3,03

Observation : la similarité de Pearson peut être négative, donc je somme les valeurs positives des similarités.

En général :

- Item-based est significativement plus **rapide**
- Item-based donne des meilleurs résultats sur **matrices creuses**
- User-based est plus facile à implanter, et produit des données qui peuvent être **plus parlants** pour les utilisateurs

Où trouver les données ?

Et si on n'est pas propriétaire d'un gros site de vente en ligne, d'écoute de musique ou d'évaluation ?

1. Plusieurs sites proposent des ***API*** *(Application Programming Interfaces)* : un langage et des classes pour « parler » avec un site ou une base de données et par exemple obtenir des données.

Exemple : Twitter, GoogleMaps, Tripadvisor...

2. Consulter des **bases des données** disponibles gratuitement (souvent pour des fins scientifiques) à propos des films, musique, livres, nourriture, sites de rencontre, produit d'Amazon..

- https://gist.github.com/entaroadun/1653794

Conclusions

Faire des recommandations signifie donner un **modèle du goût** d'un utilisateur : la similarité entre utilisateurs ou objets est une façon de définir ce concept

Pour donner des recommandation à USER on peut:

- Trouver les utilisateurs plus similaires à USER et lui recommander leurs objets préférés (**algo utilisateurs plus proches**)
- Recommander les produits avec la plus grande évaluation moyenne pesée par la similarité entre utilisateurs (**algo user-based**)
- Recommander les produits plus similaires aux derniers que USER à acheté (**algo produits plus proches**)
- Recommander les produits avec la plus grande évaluation moyenne pesée par la similarité entre produits (**algo item-based**)

Deux mesures de similarité : corrélation euclidienne et Pearson (il y en a d'autres...)

Liens avec autres cours du Master IM

- *Conception de sites Web (semestre II)* : vous alliez construire la page WEB d'un magasin en ligne avec base des données associé : l'étape suivante est recueillir les données des utilisateurs, de leurs achats et construire une système de recommandation...
- *Programmation structurée* : nous avons introduit plusieurs algorithmes (*c'était des algorithmes ou des formules ?*). Pouvez-vous les écrire dans un langage de programmation ? Reconnaissez-vous la « forme » de ces algorithmes (boucles for…) ?
- *Bases de données* : la matrice utilisateurs/produits pour des grandes applications est une base de données

Références

Ce cours est basé sur le matériel suivant :

- *Toby Segaran. Programming Collective Intelligence. O'Reilly, 2007. Chapter 2.*

Si vous êtes intéressé pour aller plus loin dans l'étude de l'apprentissage automatique, voici un bon article avec plein des références :

- http://binaire.blog.lemonde.fr/2015/06/23/pasapas/