

Projet IA 316

Simon Albergel & Guillaume Balezo

Google News



Google News - Environnement

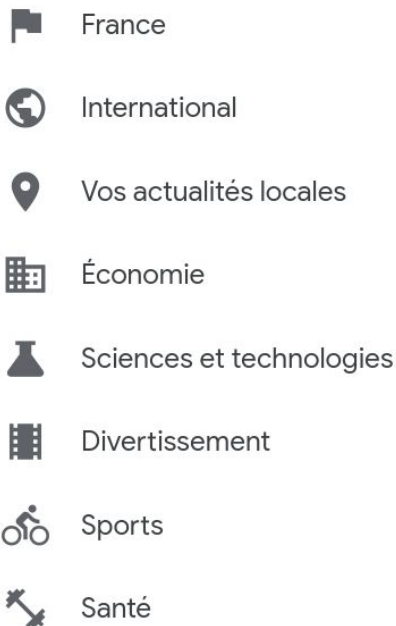
Objectifs:

- Fidéliser les clients aux produits Google
- Maximiser rétention

Particularités:

- Nouveaux articles publiés constamment
- Importance des articles exponentiellement décroissante dans le temps
- Articles populaires appartiennent à la même thématique

Google News - Modélisation des données



Un article possède trois features :

- Sa date de parution (en heure depuis le début de l'expérience)
 - Chaque jour, n_{articles} sont publiés
 - 50% d'entre eux le sont le matin selon une loi normale centrée sur 10h
 - 50% d'entre eux le sont le soir selon une loi normale centrée sur 17h
- Sa Catégorie
 - Tiré selon une gaussienne selon la headline du jour
- Sa Source
 - Tiré uniformément

Google News - Modélisation des données

Un user possède alors :

- Le jour de sa prochaine connexion (entier naturel)
 - Dans combien de jour l'utilisateur va-t-il revenir selon les articles qui lui ont été présentés
- Préférences de catégorie et de source
 - Les préférences sont tirées selon une loi uniforme (discrète pour les catégories et continue pour les sources) afin de maximiser l'entropie



Google News - Environnement

Initialisation :

- L'ensemble des articles est généré
- Tous les users se connectent au premier jour

Google News - Environnement

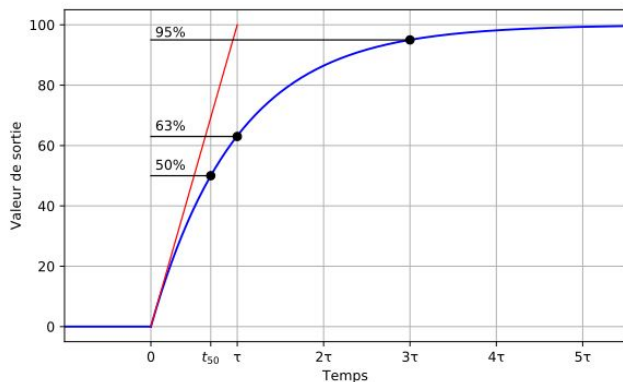
A chaque step :

- On incrémente le temps (date + 1 et prochaine connexion - 1)
- On récupère la liste des users se connectant ce jour
- On simule les horaires de connexion de chaque user
- On récupère le contexte (les articles publiés à l'heure de connexion de chaque user se connectant)
- On récupère l'action de l'agent selon le contexte
- On simule quand l'user reviendra et on en déduit le reward
- On met à jour les dates de prochaines connexion des users concernés

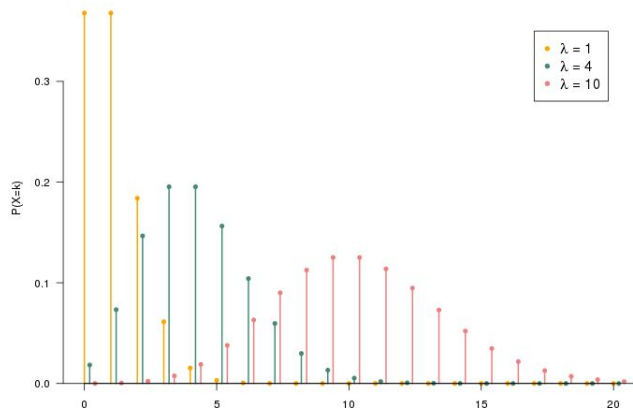
Google News - Calcul du reward

Génération des intervalles de temps de connections:

1. ps : produit scalaire entre les embeddings user et item recommandé
2. $\lambda = 1/ps * [1 - \exp(-\Delta t / \tau)] * \text{Amplitude}$
3. Tirage de la date de prochaine visite selon loi de Poisson(λ)



Evolution exponentielle



Fonction de masse loi de Poisson

Google News - Notre Environnement

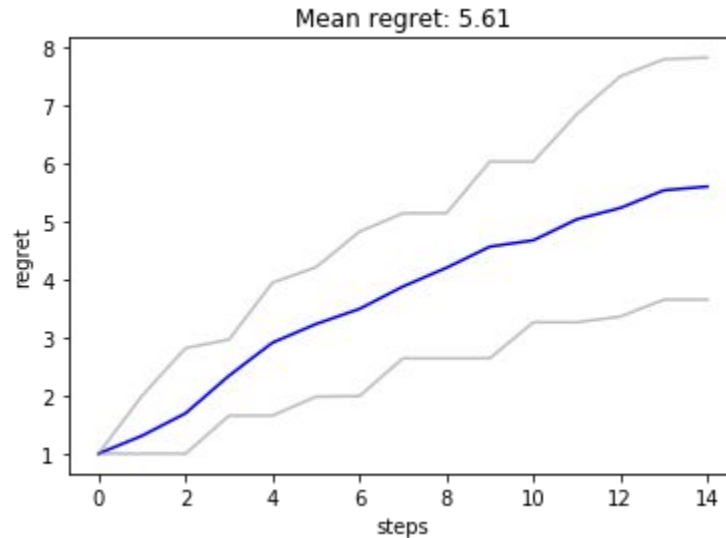
Reward: $\frac{1}{(\sum_{users} \Delta_t)^2}$

Contraintes:

- Seulement une recommandation
- Une connection quotidienne maximale
- Ajouts des articles quotidiennement

Google News - Nos résultats

- De grandes difficultés à faire tourner l'environnement
- Résultats peu satisfaisants



Google News - Conclusion

- Si la modélisation de notre environnement semble pertinente théoriquement, elle est trop complexe en pratique
- Simplifier notre environnement
 - Exemple : ne traiter la différence entre deux heures au sein d'une même journée plus que comme un bruit.

↕ 2 frames

```
<ipython-input-37-6422395d4640> in compute_best_reward(self)
    87         lambda_list_temp.append(lambda_temp)
    88         print(lambda_list_temp)
--> 89         next_logs_temp = np.random.poisson(np.round(np.asarray(lambda_list_temp).astype(np.float64), decimals=5), size=(1, len(lambda_list_temp)))
    90         best_logs.append(min(next_logs_temp))
    91         k+=1
```

```
mtrand.pyx in numpy.random.mtrand.RandomState.poisson()
```

```
common.pyx in numpy.random.common.disc()
```

```
common.pyx in numpy.random.common.discrete_broadcast_d()
```

```
common.pyx in numpy.random.common.check_array_constraint()
```

```
ValueError: lam value too large
```