

TP : Modèle d'optimisation locale du ratio Coût/Information

Objectif

Développer un algorithme en Python qui optimise le ratio de coût d'une distribution de fréquences avec des optimisations locales, comme le ferait possiblement un individu avec le langage.

On assigne à chaque mot une fréquence et on compare son ratio coût/information. Si celui-ci est plus faible que la moyenne, on l'utilisera de plus en plus, on augmente ainsi sa fréquence. Inversement si il est plus grand que la moyenne on aura tendance à moins l'utiliser.

Le but de ce TP est de programmer l'algorithme suivant, décrit étape par étape.

Algorithme

On fixe a priori des paramètres γ et ϵ proche de 0 (par exemple $\gamma = \epsilon = 0,1$) dans des variables.

1. **Initialisation des Fréquences** Créez un tableau de N fréquences, noté p_k , avec des valeurs aléatoires comprises entre 0 et 1. On pourra utiliser la fonction `rand()` dans le module `random` de `numpy`. Normalisez ensuite les fréquences pour qu'elles totalisent 1 :

$$p_k = \frac{p_k}{\sum_{j=1}^N p_j}.$$

2. **Calcul du Coût Moyen et de l'Information** À partir des fréquences, calculez la moyenne du **coût d'un mot** $C_k = \log k$, et **l'information d'un mot** $H_k = \log \frac{1}{p_k}$ notée respectivement C et H . Puis leur ratio $C^* = C/H$.
3. **Optimisation locale** Pour chaque mot, on considère le ratio $C_k^* = \frac{C_k}{H_k}$.
 - Si C_k^* se trouve dans l'intervalle $[(1 - \gamma)C^*, (1 + \gamma)C^*]$, laissez p_k inchangé.

- Sinon, ajustez p_k :
 - Augmentez p_k d'un facteur constant $1 + \epsilon$ si le ratio est plus petit que l'intervalle (le mot a un bon ratio).
 - Diminuez p_k d'un facteur constant $1 - \epsilon$ si le ratio est plus grand que l'intervalle (le mot a un mauvais ratio).

4. **Réordonnement et Normalisation** Si des fréquences ont été modifiées, réorganisez les mots en fonction de leurs fréquences dans un ordre décroissant. Renormalisez les fréquences pour qu'elles totalisent à nouveau à 1. Reprenez ensuite à l'étape 2.

Si aucune fréquence p_k n'a été modifiée, l'algorithme s'arrête.

Analyse

- Que peut-on dire de la convergence du modèle ?
- Étudier le cas où $C_k = \log(k + k_0)$.

Affichez dans les deux cas l'évolution des fréquences au cours des itérations pour analyser la convergence.

Critères d'évaluation

- **Correctitude des calculs** : l'implémentation des étapes (normalisation, calcul des coûts et des informations, ratio de coût) doit être exacte.
- **Structure du code** : l'algorithme doit suivre les étapes décrites ci-dessus et être bien organisé.
- **Commentaire et lisibilité** : le code doit être commenté pour expliquer les choix faits à chaque étape.