	Labels	Labels étendus	Plongement
Classification	DeepGraphWithNNTorch.ipynb		gcn.ipynb
Regression	DeepGraphRegression.ipynb		gcnRegression.ipynb

Optimisation

Dans la classe Net (forward) et en préambule de cette classe.

Franck wolfe : S=self.mapping_from_cost(C,n,m)

Puissance itérés ou SVD : S=self.mapping_from_similarity(C,n,m)

Puissances itérés : from svd import iterated_power as compute_major_axis (en préambule)

SVD : compute_major_axis=svd.CustomMajorAxis.apply (en préambule)

Fichiers

— svd.py : contient les classes fonctions pour obtenir une matrice d'assignement à partir d'une matrice de coût ou de similarité. Comprends : la SVD, puissance itérés, Franck Wolfe,

— regression.py : Fonction de regressions avec deux classes : Une classe qui utilise la kernel ridge regression (nécessite une matrice de distances complète) et une classe qui utilise une Knn regression (ne nécessite que la distance des données à prédire aux données de train).

— triangular_loss.py Deux classes qui pénalisent la violation de l'inégalité triangulaire. Une classe qui considère avoir une matrice de coût de substitution entre noeuds (pour les labels, cf tableau ci dessus) et une autre classe qui ne considère qu'un coût de substitution qui est multiplié à la distance euclidienne entre les plongements de noeuds.