MPRI-project-battery

final MPRI project

HMM classifier

analyse de la variance des features pour voir leur pertiance :

```
Feature : charge_nb variance of 3380.037504869084
Feature : voltage_measured variance of 0.08091295447782859
Feature : current_measured variance of 1.2593335117769673
Feature : temperature_measured variance of 22.893801758680418
Feature : current_charge variance of 0.8663792149241187
Feature : voltage_charge variance of 2.2496041828303306
Feature : ambiant_temp variance of 0.0
```

On remarque que ambiant_temp a une variance de 0.0 donc on peut l'enlever.

Cération de modèles

On va créer un modèle par qualité de batterie dans l'ensemble d'entraînement. Pour extraire ces données on utilise pandas

```
quality_1 = df.loc[df['quality' == 1]]
```

mais il faut ensuite faire attention a enlever la colonne quality des données qu'on va donner au HMM.

On regroupe ensuite les données par batterie pour par nombre de charge.

Problèmes de mise en forme des données

lors de l'extraction des données on rencontre très souvent des NaN qui vont empêcher le bon fonctionnement du HMM il faut trouver une solution pour s'en débarasser.

Sampleing des données

il y a beaucoup de données, il faut trouver une façon intelligente d'en extraire un sous ensemble

Une approche décevante a été d'utiliser une fonction logarithmique pour donner plus d'importance aux premiers échantillons de la liste.

```
def log_sampeling(arr, nb_samples):
    _max = len(arr)-1
    _exposant = 2.0
```

```
    ids = np.round(np.power(np.linspace(0,1,nb_samples),_exposant) * _max)
    return arr[ids.astype(int)]
```

Problème numéro de la charge

Dans les premières implémentations l'algorithme utilisait toujours la colonne 'charge_nb' du dataset. On a ensuite réalisé que cette information ne devait pas être connu dans l'entraînement car elle faussait la prédiction. En effet, un numéro de charge bas fait référence (souvent) à une batterie plus neuve et donc permet facilement de qualifer la batterie comme étant de bonne qualité. Mais les données de test finale seront mélangée ce qui signifie que cette information ne pourra plus être utilisée.

Toute ces spéculation on pu être vérifiée avec l'importance des features dans le model KDTree. On a pu voire de le numéro de la charge représentait à lui seul 50% de l'information de décision.

Type de covariance

après analyse des modèles de covariances du HMM le meilleur trouvé est le 'full'

Problème de surapprentissage

Il a été avéré que le modèle souffrait de surrapprentissage dans une version antérieur. L'erreur qui était faite était que le la prédiction et donc le calcul du score ne se fesait que sur l'ensemble de validation. On a pu deviner que le model overfittait car il était capable de faire des prédiction a plus de 80% de réussite.