

Optimisation de la précision des méthodes 'sum' et 'support'

Rappel et objectifs

Dans le module CDM (détection d'anomalies dans l'exécution d'un plan), seuls deux des trois méthodes sont paramétrables : les méthodes 'sum' et 'support'. Il est donc nécessaire d'optimiser autant que possible ces paramètres.

Listons les différents paramètres qui influe sur ces méthodes :

- Paramètres inconnus : le bruit.
- Paramètres observables : nombre d'objectif dans la bibliothèque de plan, nombre d'actions atomiques dans la bibliothèque de plan.
- Paramètre modifiable : précision des méthodes.

Nous omettons les autres paramètres de la bibliothèque de plan qui sont moins impactant que le nombre d'objectif et le nombre d'action atomique.

Etant donné qu'il nous est impossible de connaître le bruit, il nous faut trouver un optimal de précision peu importe le bruit (entre 0 et 30%). Nous pourrons ensuite faire de légers ajustements en fonction des deux paramètres observables cités précédemment. L'optimisation est basée uniquement sur l'efficacité de l'algorithme (pourcentage de réussite).

Protocole de test

Nous allons calculer l'efficacité des méthodes 'sum' et 'support' en fonction de la précision et du bruit. Nous faisons varier la précision de 0 à 5% par pas de 0.1% et le bruit de 0 à 30% par pas de 1%. Pour chaque configuration, nous effectuons 10 itérations sur 10 bibliothèques de plan généré automatiquement, soit une moyenne sur 100 testes. Cet ensemble de test sera représenté sous forme d'un plan avec pour axe (x : bruit / y : précision / z : efficacité).

Nous allons effectuer ce banc de test en variant le nombre d'action atomique et le nombre d'objectifs dans la bibliothèque de plan. Tous les tests seront réalisés avec des anomalies.

Analyse des résultats

Nous espérons l'apparition d'un sommet unique sur le plan qui représenterais un optimal de précision pour chaque banc de test. Cela nous permettrait de prouver l'existence d'un optimal de précision peu importe le bruit. Cela nous permettrait aussi d'adapter cet optimal en fonction du nombre d'action atomique et du nombre d'objectif dans la bibliothèque de plan.

Pour finalité, nous voudrions trouver une fonction : $f(nbG, nbA) = p$

Avec : $nbG = \text{nombre d'objectif}$, $nbA = \text{nombre d'action terminal}$ et $p = \text{précision optimal}$.

Résultats

Affichage

Les résultats sont contenus dans un tableau à trois colonnes représentant respectivement la précision, le bruit et l'efficacité correspondante. Il y a 30 valeurs possible pour le bruit et 50 pour la précision, il y a donc 1500 lignes à ce tableau.

Pour traiter ces données, j'ai utilisé le logiciel RStudio. Je joins donc les scripts utilisés pour l'importation des données et pour la représentation de ces dernières.

Importation

```
# import support result
sup510 <- read.table("resultPrecSupport-5-10.txt", header = TRUE)
sup545 <- read.table("resultPrecSupport-5-45.txt", header = TRUE)
sup5100 <- read.table("resultPrecSupport-5-100.txt", header = TRUE)
sup1010 <- read.table("resultPrecSupport-10-10.txt", header = TRUE)
sup2010 <- read.table("resultPrecSupport-20-10.txt", header = TRUE)
```

```
# import sum result
sum510 <- read.table("resultPrecSum-5-10.txt", header = TRUE)
sum545 <- read.table("resultPrecSum-5-45.txt", header = TRUE)
sum5100 <- read.table("resultPrecSum-5-100.txt", header = TRUE)
sum1010 <- read.table("resultPrecSum-10-10.txt", header = TRUE)
sum2010 <- read.table("resultPrecSum-20-10.txt", header = TRUE)
```

Traitement des données

```
# install packages
require(rgl)

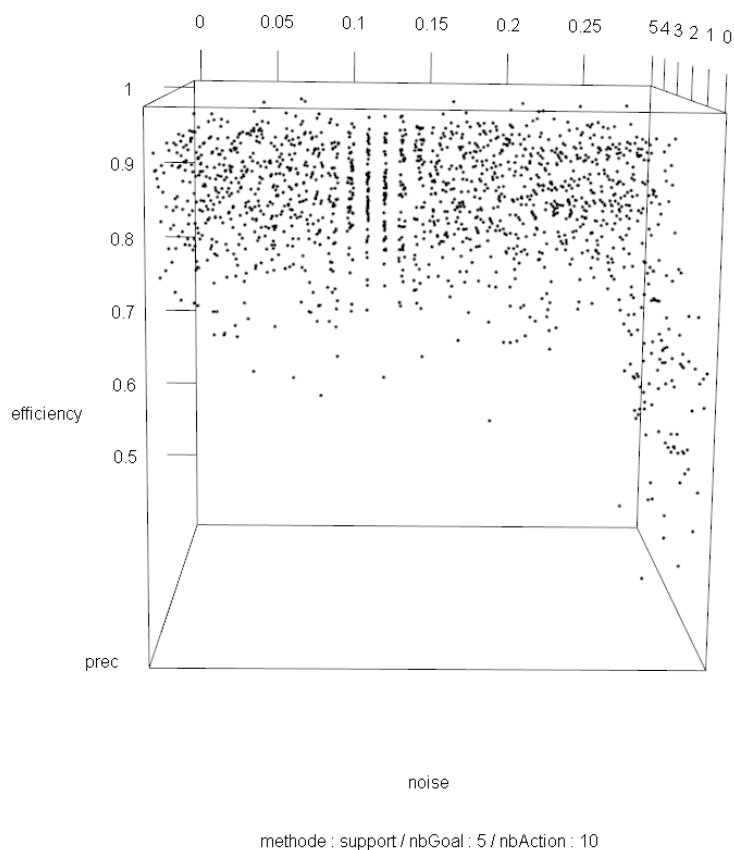
# plotting 3d
with (sum5100, plot3d(noise, prec, efficiency, sub = "methode : sum / nbGoal : 5 / nbAction : 100"))
```

Résultats

Les données brutes seront disponibles en annexe de ce document.

Méthode support

Pour la méthode 'support', tous les tests ayant été fait avec des anomalies, les résultats plafonnent entre 80% et 100% de réussite peu importe le nombre d'objectif et le nombre d'actions et la précision. En effet nous avons vu que le point faible de cette méthode était les faux positifs mais quand il s'agit de trouver une anomalie, elle est extrêmement performante.

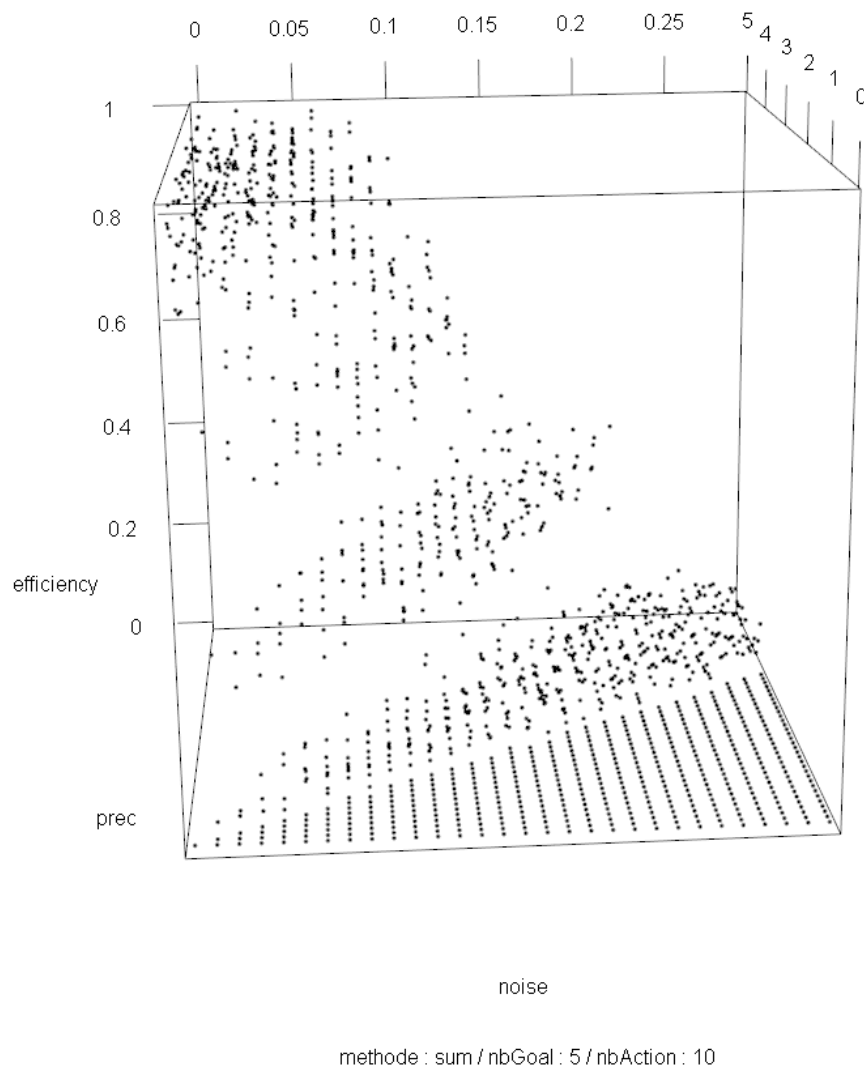


Touts les configurations sont semblable à cette dernière. Nous observons de très bon résultat peu importe la précision avec cependant une marche vers 27% de bruit qui fait considérablement baisser l'efficacité. Pour cette méthode, l'impact de la précision étant anecdotique (avec des anomalies), il serait intéressant de vois son impacte sans anomalies pour définir un optimal potentiel et éviter un maximum les faux positifs.

Les autres graphiques ne sont pas montrés ci car tous semblable à celui-ci-dessus et ce, peu importe le nombre d'action et d'objectif.

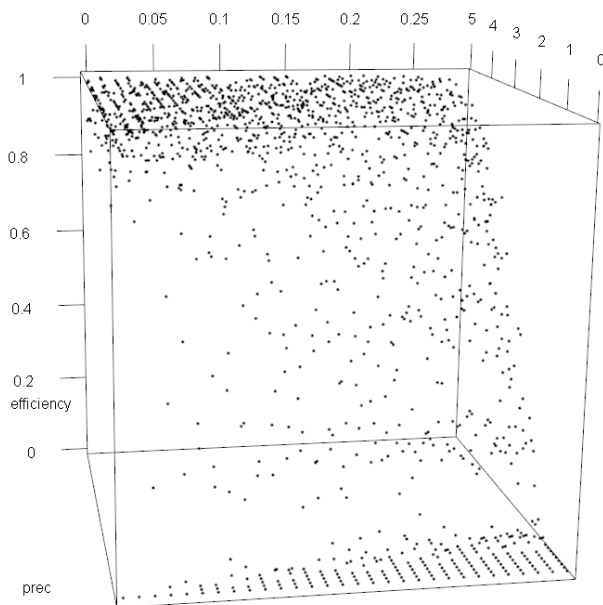
Méthode sum

Pour la méthode 'sum', nous pouvons constater une réelle évolution de l'efficacité en fonction de la précision et du bruit :



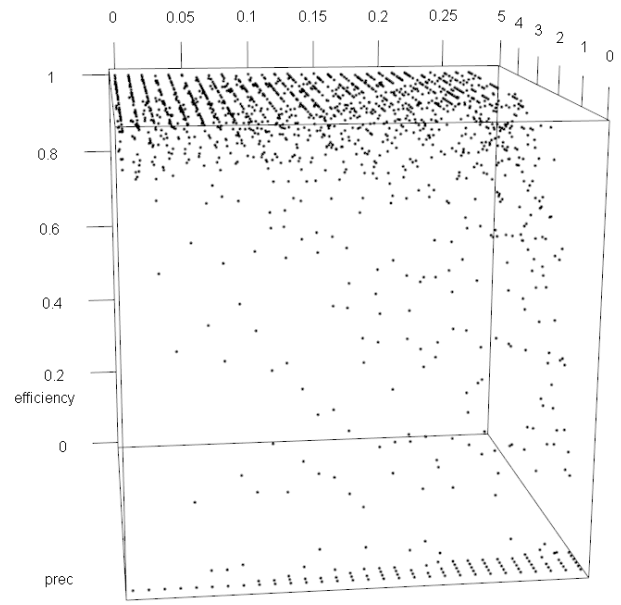
Nous observons que plus la précision est élevée, plus la méthode résiste au bruit. Il y a un seuil à partir de ~3.5% de précision où la résistance au bruit arrête de croître. Il faudra cependant faire des tests sans anomalie pour voir comment la précision influe sur les faux positifs.

Pour cette méthode, seul le nombre d'action à un impact significatif (contrairement au nombre d'objectif) :



noise

methode : sum / nbGoal : 5 / nbAction : 45



noise

methode : sum / nbGoal : 5 / nbAction : 100

Nous voyons avec ces graphiques (avec 45 et 100 actions au lieu de 10) que plus le nombre d'action est élevé, moins le bruit a d'impact sur le résultat. Il serait donc intéressant d'utiliser cette méthode sur une bibliothèque de plan avec un grand nombre d'action atomique distinct.

Les autres graphiques ne sont pas montrés ci car tous semblable à celui-ci-dessus et ce, peu importe le nombre d'objectif.

Conclusion

L'optimisation du paramètre de précision est très différente pour les deux méthodes 'sum' et 'support'. Pour la méthode 'support', aucun des paramètres (précision, nombre d'action et d'objectif) ne fait varier les résultats quand il y a des anomalies. Il faudra donc faire ce même banc de test sans anomalie pour avoir une piste d'optimisation. De plus, la marche dès 27% de bruit, semble constante peu importe les paramètres. Pour la méthode 'sum', la précision est importante pour la résistance au bruit mais il n'y a aucun changement au-delà de 3.5% (pour 10 actions). Il serait donc intéressant de faire les mêmes tests sans anomalies pour optimiser le taux de faux positif. De plus, ces « 3.5% » ne sont pas fixés et dépendent du nombre d'action de la bibliothèque de plan. Il faudrait donc trouver la fonction qui lie ces deux paramètres.