

FAQ

Contents

1 25 06 2021

- 1.1 Mieux que le pipeline ideal
- 1.2 Facteur de nuisance comme erreur de mesure
- 1.3 Formule générative pour les toys
- 1.4 Grandeurs mesurées / objectifs
- 1.5 Métrique de succès
- 1.6 Notations

1 25 06 2021

1.1 Mieux que le pipeline ideal

- La figure 16 est problématique : toutes mes méthodes font mieux (en moyenne et en moustaches) que l'optimal - sauf TP, qui est très correct, voire excellent, dans toutes les autres configurations, mais très mauvais dans ce cas.

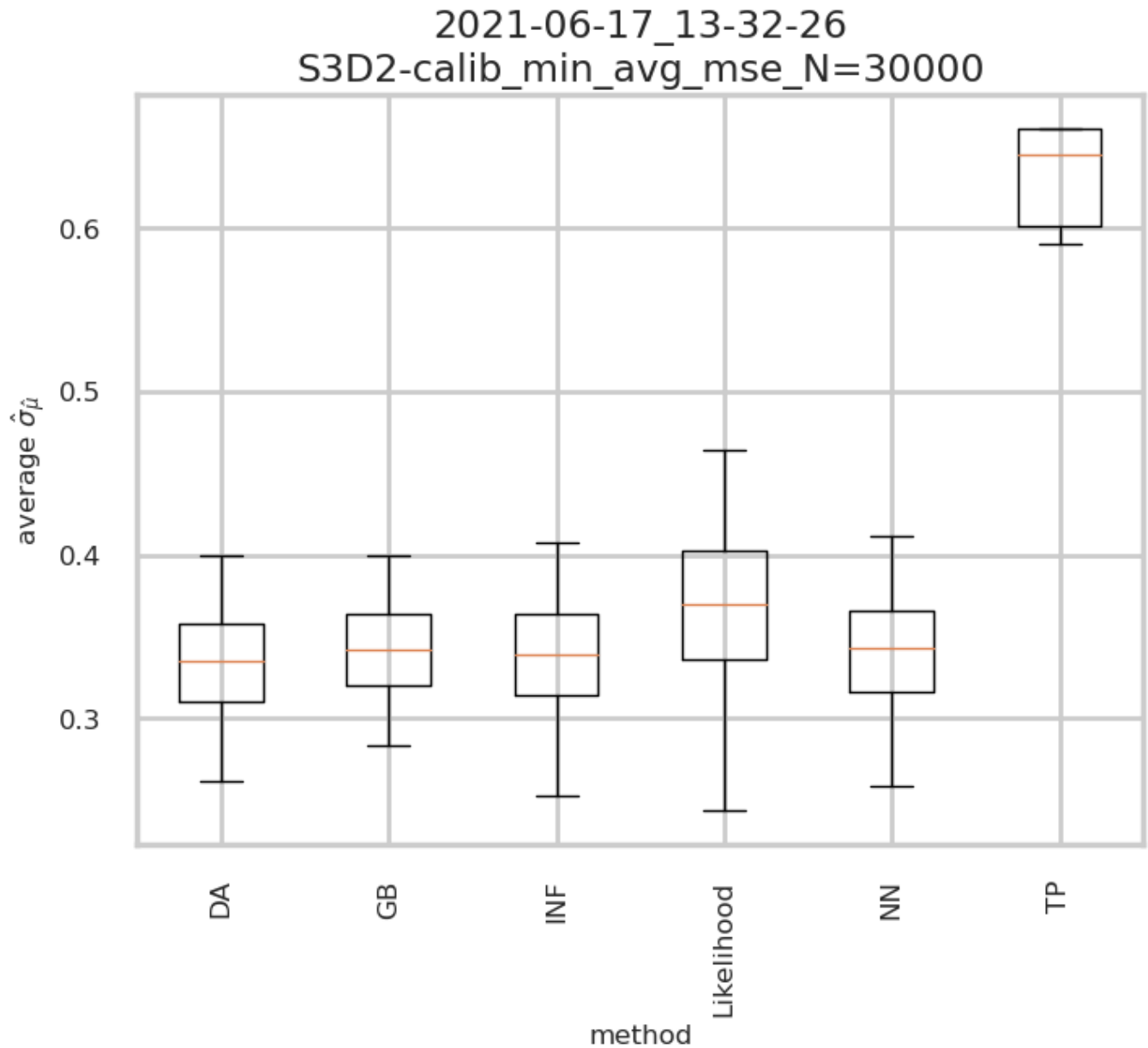


Figure 1: Anciennement figure 16 : Confidence interval length using **hybrid** and ideal (label likelihood) pipeline on the 3D toy

Plusieurs explications sont possibles.

Premièrement la conclusion que les méthodes font mieux que l'optimal n'est pas évidente. La taille estimée de l'intervalle de confiance reste très similaire entre le pipeline hybride et idéal sur ce cas.

Deuxièmement il est possible que les intervalles obtenues avec le pipeline hybride n'ai pas le même coverage. Ce calcul passe par plusieurs approximations dont : le fait que la Cramer-Rao bound soit

atteinte; une inversion numérique d'une matrice (la Hessian); une Hessian calculée numériquement avec des différences finies; et un nombre fini de points. On observe peut-être simplement une divergence entre la théorie mathématique et la réalité numérique due à l'implémentation.

1.2 Facteur de nuisance comme erreur de mesure

Question 1 (diapo 1):

Selon mon schéma, x est affecté par le facteur de nuisance, pas β ou γ , ou même μ . En effet, j'interprète le facteur de nuisance comme une erreur de mesure qui s'introduit dans les détecteurs. β , γ et μ sont des constantes physiques. Voulez-vous bien clarifier le mécanisme et le cas échéant rectifier mon schéma.

Les paramètres de nuisance ne sont pas des erreurs de mesure. Mais bien des paramètres du modèle statistique à estimer au même titre que le paramètre d'intérêt. μ est une constante physique, c'est la déviation par rapport à la valeur attendu par la théorie (et les mesures d'expériences précédentes). β et γ ne sont pas des constantes physiques mais des fonctions.

Exemple pour β . β est le nombre estimé (par la théorie en utilisant un simulateur) de backgrounds qui arrivent dans une région donnée de l'espace des mesures. Hors la position des backgrounds dans l'espace des mesures dépend d'un paramètre inconnu/libre α . Inconnu dans le sens où il n'est pas mesurable directement mais va être estimé au même titre que μ . Pour une valeur de α donnée on peut calculer la position des backgrounds et donc estimer combien de background se trouve dans la région donnée (je décris ici un Monte-Carlo en fait). Donc β est une fonction de α (déterministe car la seed pour le Monte-Carlo est fixe).

Pour γ c'est pareil mais avec les signaux.

1.3 Formule générative pour les toys

Question 1.bis: Quelle formule(s) mathématique(s) sont-elles utilisées pour engendrer les données des exemples toy.

Toy 1D

$$p(x|\mu, \alpha) = \mu \mathcal{N}(x|\alpha a, \alpha d) + G(x|k, \alpha) \quad (1)$$

$$G(x|k, \alpha) = \frac{x^{k-1} e^{-\frac{x}{\alpha}}}{\Gamma(k) \alpha^k} \quad (2)$$

Et N est une loi Normale.

Référence : <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gamma.html>

Code :

```
x_b = stats.gamma.rvs(gamma_k, loc=gamma_loc, scale=gamma_scale)
```

//

Avec

```
gamma_k = 2
gamma_loc = 0
gamma_scale = alpha #&nbsp;le paramètre de nuisance
```

//

Pour le toy 3D je reprend le toy du papier d'INFERNO. Mais j'y applique les contraintes que l'on a sur HiggsML : backgrounds connu mais nombre total d'évènement aléatoire.

$$f_b(x|r, \lambda) = \mathcal{N}\left((x_0, x_1) | (2+r, 0) \begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}\right) \text{Exp}((x_2|\lambda)) \quad (3)$$

$$f_s(x|r, \lambda) = \mathcal{N}\left((x_0, x_1) | (1, 1) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{Exp}((x_2|2)) \quad (4)$$

Leading to the likelihood :

$$p(x|r, \lambda, \mu) = f_b(x|r, \lambda) + \mu f_s(x|r, \lambda) \quad (5)$$

1.4 Grandeurs mesurées / objectifs

Question 2 (diapo 2):

Le but est d'obtenir:

- un estimator de μ (appelé μ_{hat})
- éventuellement estimateur de α (mais c'est facultatif)
- un estimateur de la "barre d'erreur sur μ "

Le dernière quantité "barre d'erreur sur μ " est-elle:

- un estimateur de CI μ
- un estimateur de CI μ_{hat}

La "barre d'erreur sur μ " doit elle capturer:

- la variabilité de μ
- la variabilité de α
- la variabilité de x
- autre

La “barre d'erreur de μ ” est l'intervalle de confiance de μ au sens fréquentiste. La vraie valeur de μ est et restera à jamais inconnue. On utilise un estimateur d'intervalle CI qui est une fonction des données X et doit répondre à la propriété : pour toute valeur de μ , avec laquelle on génère des données X , l'intervalle $CI(X)$ doit contenir μ avec une probabilité $P_{coverage}$ choisie.

C'est la méthode qui est actuellement retenue au CERN pour présenter un intervalle de confiance. Le débat sur quel intervalle (fréquentiste ou Bayésien) doit être mis en valeur dans les publications revient sûrement de temps en temps. Dans nos travaux on affiche l'intervalle fréquentiste.

Seul la méthode de régression directe donne un intervalle Bayésien. C'est à dire un intervalle contenant $P_{coverage}$ probabilité du posterior sur μ .

Dans les deux cas une infinité de ces intervalles existent. On prend l'intervalle centré (50% des probabilités de chaque côté) sur la valeur du paramètre la plus probable (maximum likelihood ou maximum a posteriori)

Du coup on a deux types d'intervalles différents. Mais je n'ai compris les différences entre fréquentisme et Bayésianisme qu'après avoir développé la régression directe et le pipeline hybride.

1.5 Métrique de succès

Question 3:

Quelles sont les métriques de succès qui nous permettent empiriquement d'évaluer la qualité des quantités obtenues à la question 2? Quelle est leur formule mathématique?

On doit mesurer les performances d'un estimateur ponctuel $\hat{\mu}$ et d'un estimateur d'intervalle de confiance ci .

Pour l'estimateur ponctuel la mean squared error (MSE) est une métrique classique.

$$MSE(\hat{\mu}) = \frac{1}{N} \sum_{i=1}^N (\hat{\mu}_i - \mu^*)^2 \quad (6)$$

où $\hat{\mu}_i$ est une i ème réalisation de $\hat{\mu}$ et μ^* est la vraie valeur de μ ayant servi à générer les données sur lesquelles on tire $\hat{\mu}_i$.

1.6 Notations

Question subsidiaire:

La notation “alpha” pour le facteur de nuisance est-elle standard. Glen utilise “nu”, pivot utilise “z”, inferno utilise “r” et “lambda”. Je trouve le choix de “alpha” assez mauvais car c'est une lettre utilisée habituellement pour désigner le “significance level” des tests ou les paramètres de mixture des densités, dont nous allons avoir besoin.

C'est vrai que α n'est pas un bon choix.

J'avais éliminé les autres notations.

“z” est la lettre qui suit w, x et y qui sont des données du problème. Hors le paramètre de nuisance n'est pas une donnée du problème. Je voulais éviter la confusion.

Le paramètre de nuisance est très proche du paramètre d'intérêt μ . J'ai voulu prendre des lettres grecques pour tout ce qui touche au modèle statistique. D'où l'idée de garder μ pour le paramètre de nuisance et de prendre β et γ pour les éléments constituant le paramètre de la loi de Poisson.

Je n'utilise plus s et b pour le nombre attendu de background et de signaux car c'était ambiguë avec les classes/catégories S et B ou encore la variable aléatoire S qui représente le nombre de signaux ou la réalisation s de cette variable aléatoire.

J'ai utilisé ν pendant un temps. “nu” pour “nuisance” ça passe bien ! Mais la prononciation de “mu” et “nu” sont trop proche. Ça rend la communication en réunion difficile pour rien.

Donc j'ai pris α . C'est la première lettre de l'alphabet.

Peut-être que changer μ serait plus pertinent ?

J'évite θ que je garde pour désigner un paramètre générique et ϕ que je réserve pour les paramètres/poids des réseaux de neurones. λ est réservé pour l'hyper-paramètre des contrôlant l'impact de régularisations dans les loss functions.

Il reste ψ ou η ou d'autre ?

formatted by Markdeep 1.13 