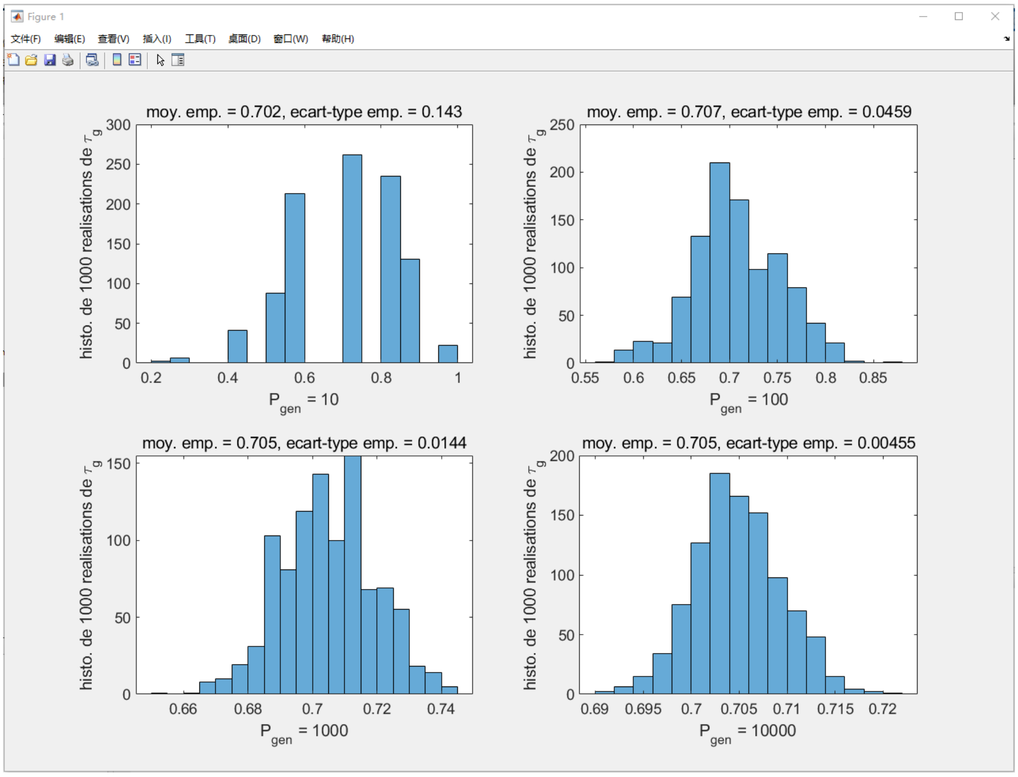
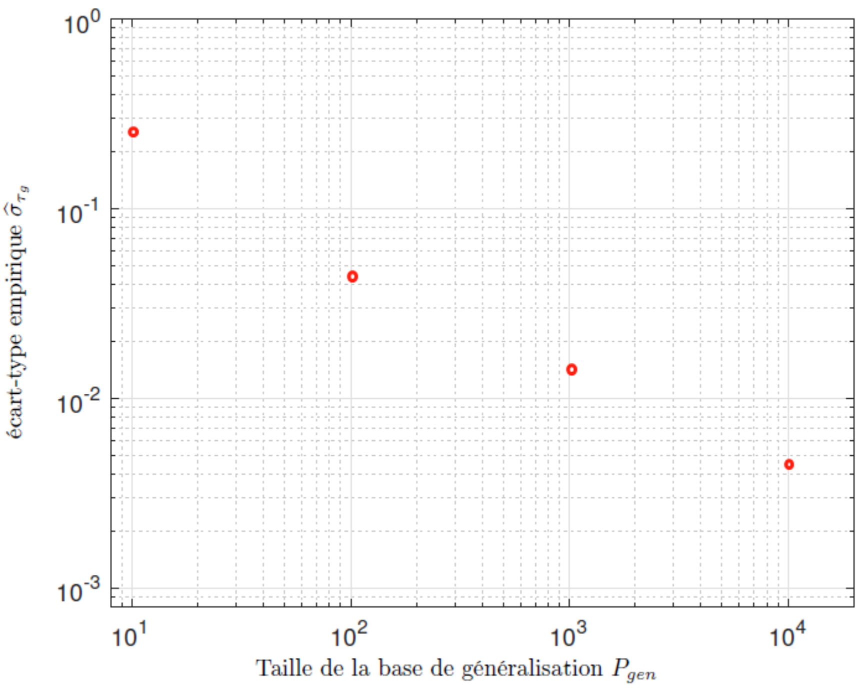
Pour la première partie, on trouve que pour la base de données égale 1, les deux classes sont bien séparés, dans ce cas-là, l'algorithme de Hebb est moins efficace que le PI, et le PI est presque coïncident que le discriminateur. Et pour le P_{app} plus grand, l'exactitude de PI est plus haut. Pour les choix de la base plus haut, les deux classes sont moins séparés, et dans ce cas-là les effets de Hebb et PI sont proches.
Pour la partie 2, on obtient un résultat ;





Pour la relation $\sigma_{\tau_g} = \sqrt{\frac{\mu_{\tau_g} \ (1-\mu_{\tau_g})}{P_{app}}}$, les 4 résultats bien correspondent la relation. Du coup P_{app} est plus grand, σ_{τ_g} est plus petit. Ce sont tout les cas M égale à 1000, pour M égale à 1, je pense que c'est moins intéressant parce qu'il y a seulement une réalisation et il y'n a pas le écart-type. Mais je pense si P_{app} est assez grand, peut-être la densité de σ_{τ_g} correspond à la loi de binomial.

Pour la partie 3, pour les différents P_{app}, on trouve que le taux de réussite d'apprentissage le PI est presque 1 et plus haut que le Hebb, le taux de réussite de généralisation augmente avec P_{app}. Après on choisit différent P_{gen} et on trouve qu'au cas de grand P_{gen} le taux de PI est plus grand que le PI et ils ce sont séparés un peu. C'est intéressant que pour le P_{app} envions 40, il y a un grand changement sur le τ_g .

Pour la partie 4 je ne comprends pas pourquoi on utilise $\frac{\sigma}{\sigma+\sqrt{N}}$

Et j'ai attendu que dans la cas choisir nouvelle base de gen, tout les τ_{app}= 1 et je trouve que les τ_g sont proches pour le PI et RA, quelles sont les différences entre le PI et RA et pourquoi on formule cette algorithme.