Baptiste PRAS - Document de Travail Provisoire

Baptiste PRAS

22 mars 2025

Introduction :

Ce `document de travail provisoire` s'inscrit dans le cadre de mon **Travail Encadré de Recherche (TER)**, réalisé sous la supervision de François Landes, et porte sur l'impact du déséquilibre de classes dans l'ensemble d'entraînement sur les performances des tâches de classification. L'objectif est de déterminer si le ptrain optimal se rapproche systématiquement de 0.5 selon la méthode et la loss utilisée. Il est provisoire et sert simplement de notes sur l'avancée de mon travail, et `ne constitue aucunement un rapport de stage` ou un document final.

Ce travail s'appuie en particulier sur les résultats présentés dans **Class Imbalance in Anomaly Detection: Learning from an Exactly Solvable Model**, un article étudiant l'effet du déséquilibre des classes dans un cadre théorique simplifié, où les données suivent une distribution gaussienne et où la fonction de coût utilisée n'est pas entraînable par descente de gradient.

Mon sujet se concentre spécifiquement sur l'étude de l'impact du déséquilibre dans un cadre similaire, mais en remplaçant la fonction de coût originale par des pertes plus courantes telles que la *hinge loss* et la *perceptron loss*. L'objectif est d'analyser comment ces modifications influencent les dynamiques d'apprentissage, en utilisant des méthodes d'optimisation telles que la *descente de gradient* et la *dynamique de Langevin*.

Table des matières

1 Implémentation du Teacher

Pour implémenter le Teacher pour un Perceptron sphérique, on génère un dataset selon une distribution gaussienne centrée en 0 et de variance $\frac{1}{\sqrt{D}}$:

$$X \sim \mathcal{N}(0, I_D)$$

Le dataset est de taille (N, D), où N est le nombre de points de données et D leur dimension. Puisqu'on utilise en entraînement 10 plis dans la cross-validation, on garde un α ($\frac{N}{D}$) de 10 pour garantir un apprentissage correct et éviter l'underfitting ou l'overfitting.

Pour éviter que le dataset soit trivialement linéairement séparable, on peut ajouter du bruit lors de la décision du Teacher de classifier un point en classe -1 ou $+1$. Pour cela, on perturbe la frontière en ajoutant un ϵ lors du calcul de la classe du point. Voici la formule utilisée par le Teacher :

$$Y_N = sign(w \cdot X_N + b + \epsilon)$$

où ϵ est tiré selon une distribution gaussienne décrite précédemment et Y_N est la classe donnée au point par le Teacher.

Ne pas ajouter de bruit mène à un dataset linéairement séparable et donc systématiquement à une accuracy de 1 sur le train set lors d'un entraînement par descente de gradient.

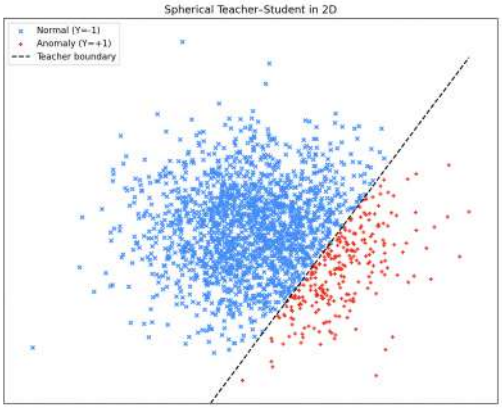

FIGURE 1 – Teacher sans ajout de bruit ($N = 2000$, $D = 2$), le dataset est linéairement séparable.

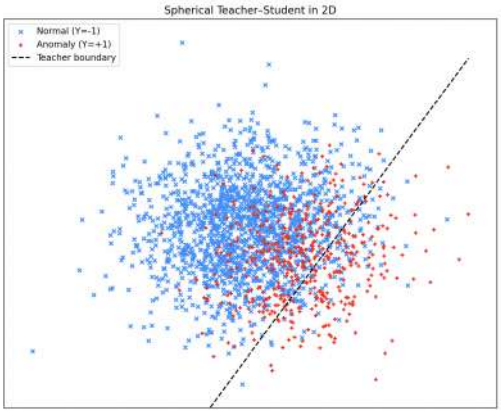

FIGURE 2 – Teacher avec ajout de bruit ($N = 2000$, $D = 2$), le dataset n'est plus linéairement séparable.

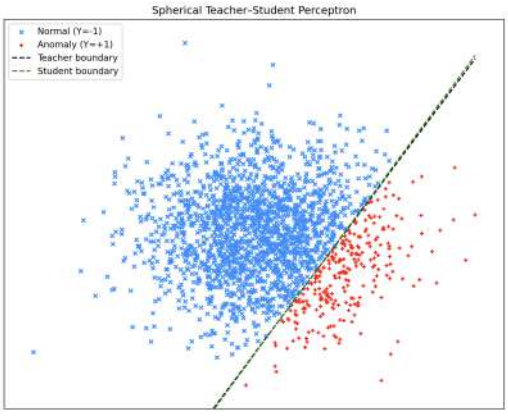

FIGURE 3 – Sans ajout de bruit, le Student chevauche le Teacher, il a presque appris les poids du Teacher et obtient une accuracy de 1 sur le train set.

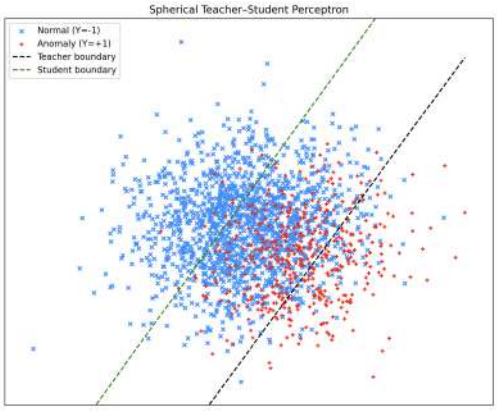

FIGURE 4 – Avec ajout de bruit, le Student n'apprend pas parfaitement les poids du Teacher et obtient une accuracy nettement inférieure à 1 sur le train set.

2 Descente de gradient (avec ajout de bruit)

2.1 Contexte

On commence quelques analyses en entraînant le Student. On choisit $n\_splits = 10$ pour la cross-validation, $test\_size = 0.2$, $eta = 0.1$ et $maxiter = 100$. On entraîne le student sur 42 valeurs de *ptrain* différentes, générées selon cette commande en Python (valeurs disponibles dans la section 6) :

$$np.linspace(0, 1, 44)[1 : -1]$$

Il faut un minimum de $N = 500$, $D = 50$ pour avoir des résultats exploitables, en deçà de 500 points de données, les performances chutent très vite et on fait face à de l'underfitting.

On entraîne le Student par descente de gradient en cherchant à minimiser la fonction de perte suivante :

$$E(w, b) = \sum_{n=1}^{N} f(X_n, Y_n, w, b)$$

où Y_n est la vérité terrain donnée par le teacher, X_n un point de donnée, w et b les poids et le biais du Student, et f est la fonction de loss utilisée. Ici, on utilise deux fonctions de loss différentes, la hinge loss et la loss du perceptron :

$$f_{hinge}(X_n, Y_n, w, b) = \max(0, 1 - Y_n \cdot (w \cdot X_n + b))$$

$$f_{perceptron}(X_n, Y_n, w, b) = \max(0, -(Y_n \cdot (w \cdot X_n + b)))$$

Pour les sections 2 à 5, la modification de *ptrain* se fait via du over- et under-sampling dans le dataset. Pour tout le document, le score est donné par la *balanced accuracy*. Le code permet de paralléliser l'exécution des différents *ptrain* pour aller plus vite.

2.2 Résultats

On donne ci-dessous quelques premiers résultats obtenus. On rappelle que l'on garde $\alpha = 10$.

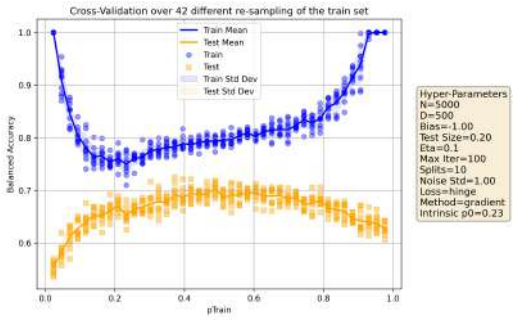

FIGURE 5 – Apprentissage par descente de gradient avec la hinge loss sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.23$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal *ptrain* est 0.4884.

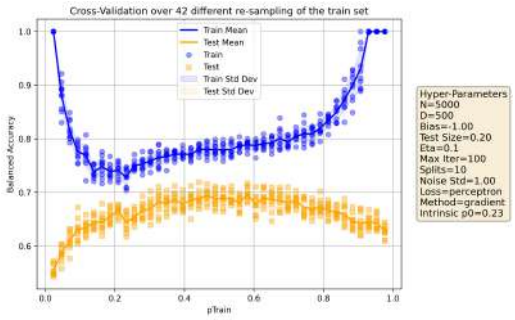

FIGURE 6 – apprentissage par descente de gradient avec la loss perceptron sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.23$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal *ptrain* est 0.5814.

2.3 Conclusion

Lors d'un entraînement par descente de gradient avec ajout de bruit, peu importe la loss utilisée, le taux optimal de *ptrain* semble être ≈ 0.5. Il est cependant plus haut pour la *loss du perceptron*.

3 Dynamique de Langevin (avec ajout de bruit)

3.1 Contexte

Pour la dynamique de Langevin, la première étape a été d'entraîner les hyper-paramètres T et $maxiter$. Après entraînement de T, on peut constater une valeur optimale de 0.01 environ, et après entraînement de $maxiter$, on peut voir que l'on converge généralement en 7000 itérations.

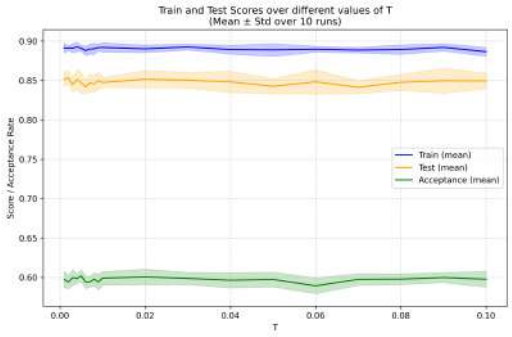

FIGURE 7 – Optimisation du paramètre T avec $N = 5000$, $D = 500$ et $B = -1$. On remarque que 0.01 semble optimal pour les performances de test.

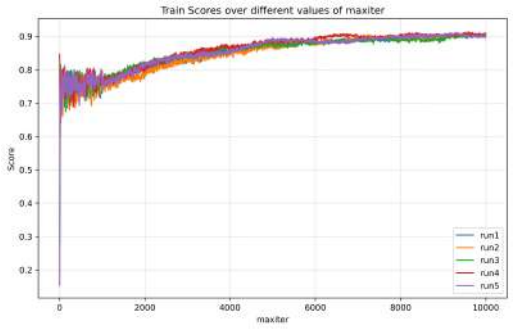

FIGURE 8 – Optimisation du paramètre $maxiter$ avec $N = 5000$, $D = 500$ et $B = -1$. On remarque qu'après 7000 itérations on semble avoir convergé.

3.2 Résultats

On a ensuite pu faire quelques premières exécutions avec cette méthode, avec les mêmes paramètres que pour la descente de gradient (voir section 2) pour $ptrain$, $test\_size$ et n_splits, et avec $T = 0.01$ et $maxiter = 7000$. Sur les 1000 premières itérations, on prend $T = 0.1$ pour se rapprocher plus vite d'une potentielle solution. On garde la hinge loss et la loss du perceptron comme définies précédemment.

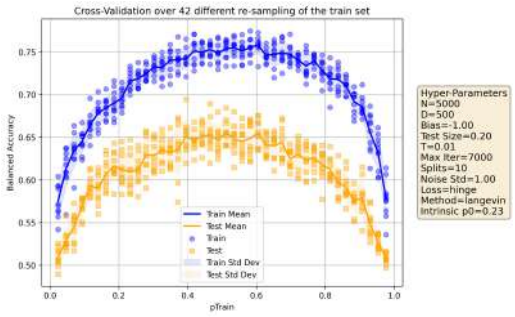

FIGURE 9 – apprentissage par dynamique de langevin avec la hinge loss sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.23$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.5116.

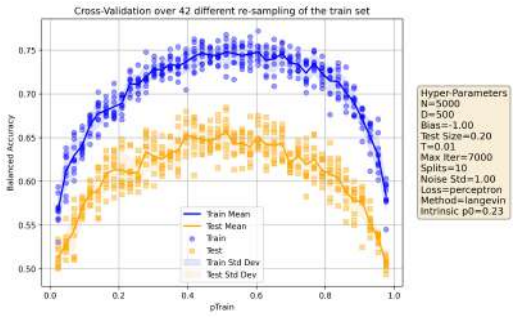

FIGURE 10 – apprentissage par dynamique de langevin avec la loss perceptron sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.23$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.5116.

3.3 Conclusion

Avec la dynamique de langevin, lorsqu'il y a du bruit ajouté au dataset, on peut voir qu'on obtient un taux optimal $ptrain \approx 0.5$.

4 Analyses sans ajout de bruit

4.1 Contexte

Pour recoller au papier et au sujet initial, on réalise les mêmes expériences sans ajout de bruit, c'est-à-dire avec $noise\_std = 0$. On garde toujours 42 valeurs de $ptrain$, $n\_split = 10$, $test\_size = 0.2$, $eta = 0.1$ et $maxiter = 100$. Pour les hyper-paramètres de la dynamique de langevin, on retrouve toujours après entraînement $T = 0.01$ et $maxiter = 7000$.

4.2 Résultats

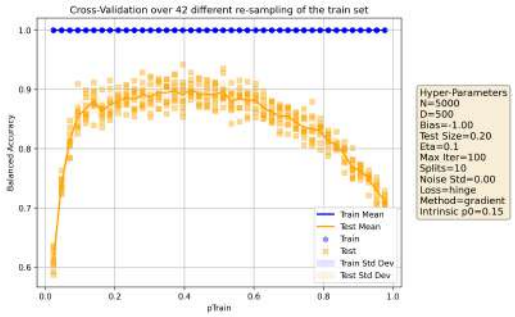

FIGURE 11 – apprentissage par descente de gradient avec la hinge loss sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.15$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.4186.

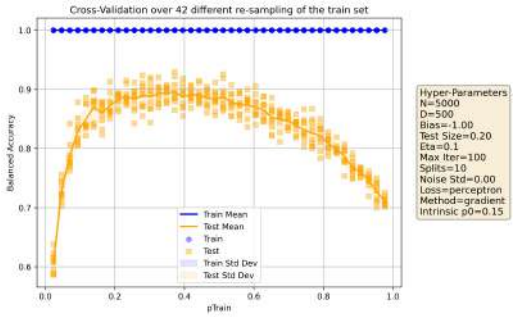

FIGURE 12 – apprentissage par descente de gradient avec la loss perceptron sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.15$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.3721.

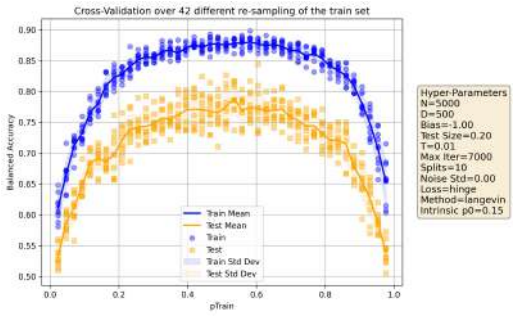

FIGURE 13 – apprentissage par dynamique de langevin avec la hinge loss sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.15$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.5349.

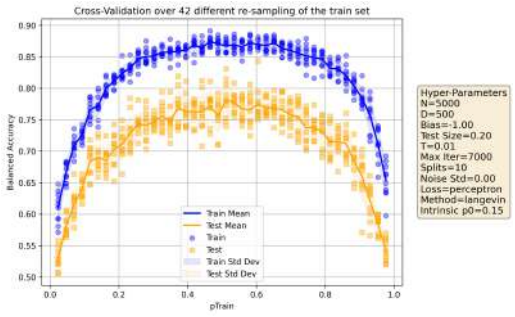

FIGURE 14 – apprentissage par dynamique de langevin avec la loss perceptron sur un dataset de taille $N = 5000$ et $D = 500$, avec $p0 = 0.15$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.5116.

4.3 Conclusion

Sans ajout de bruit, dans le cas d'un entraînement par *descente de gradient*, on obtient une performance de 1 sur le *train set* puisque le dataset est linéairement séparable. On remarque un taux optimal $ptrain$ supérieur à $p0$ et légèrement inférieur à 0.5 (≈ 0.4).

Avec un entraînement par la *dynamique de langevin*, le *train set* n'obtient pas une performance de 1, et on remarque un taux optimal $ptrain \approx 0.5$.

5 Analyses avec la loss du papier

5.1 Contexte

Enfin, on effectue la même analyse que précédemment en utilisant la *dynamique de langevin* sur la loss du papier, avec et sans ajout de bruit. On garde les mêmes hyper-paramètres que précédemment. Pour rappel, la loss du papier est la suivante :

$$f_{square}(X_n, Y_n, w, b) = \frac{1}{2} * (sign(w \cdot X_n + b) - Y_n)^2$$

5.2 Résultats

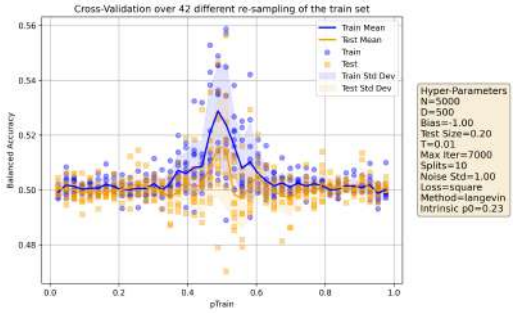

FIGURE 15 – apprentissage par dynamique de langevin avec la square loss sur un dataset de taille $N = 5000$ et $D = 500$, avec ajout de bruit, avec $p0 = 0.23$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.4884.

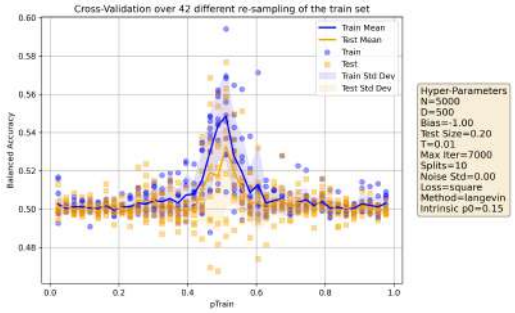

FIGURE 16 – apprentissage par dynamique de langevin avec la square loss sur un dataset de taille $N = 5000$ et $D = 500$, sans ajout de bruit, avec $p0 = 0.15$ (taux de déséquilibre intrasèque) et $ptest = p0$. On remarque que le taux de déséquilibre optimal $ptrain$ est 0.5116.

5.3 Conclusion

Avec la loss du papier, qu'il y ait un ajout de bruit ou non, on remarque des performances très dégradées mais un $ptrain$ optimal ≈ 0.5.

Par la suite, on va changer la méthode de modification de $ptrain$ pour utiliser du reweighting directement dans la loss, plutôt qu'en faisant du over- et under-sampling. On referra alors les mêmes analyses pour voir si l'on observe ou non des changements.

6 Ressources

Le code utilisé est disponible sur mon github.

Index	**Valeur**	**Index**	**Valeur**	**Index**	**Valeur**
1	0.0233	15	0.3488	29	0.6744
2	0.0465	16	0.3721	30	0.6977
3	0.0698	17	0.3953	31	0.7209
4	0.0930	18	0.4186	32	0.7442
5	0.1163	19	0.4419	33	0.7674
6	0.1395	20	0.4651	34	0.7907
7	0.1628	21	0.4884	35	0.8140
8	0.1860	22	0.5116	36	0.8372
9	0.2093	23	0.5349	37	0.8605
10	0.2326	24	0.5581	38	0.8837
11	0.2558	25	0.5814	39	0.9070
12	0.2791	26	0.6047	40	0.9302
13	0.3023	27	0.6279	41	0.9535
14	0.3256	28	0.6512	42	0.9767

TABLE 1 – Liste des 42 valeurs de p_{train} avec leur index.