Réduction du biais dans la classification de données sismique : méthodes de gestion des jeux de données asymétriques

Chantal van Dinther, Marielle Malfante, Pierre Gaillard, Yoann Cano

Réduction du biais dans la classification de données sismique : méthodes de gestion des jeux de données asymétriques

Chantal VAN DINTHER[1] Marielle MALFANTE[1] Pierre GAILLARD[2] Yoann CANO[3]

[1]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

[3]Université Paris-Saclay, CEA, DIF, F-91297, Arpajon, France

Résumé – Dans ce travail, nous mettons en place et comparons différentes stratégies pour traiter les jeux de données déséquilibrés pour la classification de signaux sismiques. Notre jeu de données contient 80k échantillons répartis en six classes, la classe majoritaire ayant dix fois plus d'échantillons que les classes minoritaires. Les quatre stratégies que nous explorons sont le rééchantillonnage de l'ensemble d'apprentissage, la pondération des classes dans la fonction de perte par l'inverse de la fréquence des échantillons, la pondération des classes par le nombre effectif d'échantillons et l'utilisation de la perte focale. En utilisant des réseaux de neurones convolutionnels (CNN) simples, la *balanced accuracy* (BA) est de 0.17, avec un biais vers la classe majoritaire. Toutes les stratégies visant à traiter les jeux de données déséquilibrés améliorent de manière significative les performances du modèle. Pour notre étude, la meilleure performance n'est pas obtenue en utilisant les techniques de pondération par classe les plus avancées, mais en pondérant simplement la fonction de perte par l'inverse de la fréquence des classes, avec une BA de 0.64. Les questions sur l'optimisation des paramètres du modèles et des paramètres d'apprentissage sont également abordées.

Abstract – In this work, we compare different strategies to deal with class size imbalanced datasets for the classification of seismic signal. Our dataset contains 80k samples and is divided into six classes, where the majority class has ten times as many samples as the minority classes. The four strategies we explore are resampling of the training set, class-weighting in the loss function by inverse of sample frequency, class-weighting by effective number of samples and focal loss. By using a simple convolutional neural network (CNN), we obtain a balanced accuracy of 0.17, with a bias towards the majority class. Any of the strategies to deal with imbalanced datasets significantly improves model performance. For our study, the best performance is not achieved by using the most advanced class-weighting techniques, but by simply weighting the loss function by inverse of class frequency with a BA of 0.64. The optimisation of learning and model parameters are also addressed in this work.

1 Introduction

Ces dernières années, la quantité de données sismiques acquises a considérablement augmenté. Par conséquent, le besoin d'automatiser de manière fiable certaines tâches telles que la détection et la classification s'est intensifié. Comme pour de nombreuses applications traitant des données de terrain, les jeux de données sont déséquilibrés en termes de nombre d'échantillons par classe. En général, l'utilisation de jeux de données asymétriques pour la classification à l'aide de CNN conduit à des modèles biaisés, ce qui diminue la fiabilité des prédictions. Dans cet article, nous comparons quatre stratégies différentes pour faire face à un jeu de données déséquilibré pour la classification sismique à l'aide de CNN. Nous avons choisi un CNN 1D plutôt qu'un RNN parce que notre objectif est un système embarqué et que l'efficacité des calculs est donc importante.

L'état de l'art regroupe différentes méthodologies pour traiter les jeux de données déséquilibrés, avec en particulier des stratégies travaillant sur les jeux de données (rééchantillonnage du jeu de données), et des stratégies travaillant sur la fonction de perte (spécifiques aux réseaux de neurones). Ces approches sont plus traditionnellement développées pour des jeux de données d'images, mais peuvent être transposées à la classification de séries temporelles. L'équilibrage d'un jeu de données peut être réalisé soit en sous-échantillonnant la classe majoritaire, soit en sur-échantillonnant les classes minoritaires en utilisant des méthodes d'augmentation des données par exemple. Indépendamment de la perte potentielle d'informations précieuses, le sous-échantillonnage peut être préférable au sur-échantillonnage [5]. Dans un cadre applicatif tel que la sismologie où les résultats d'analyse servent aussi à mieux comprendre les phénomènes physiques étant à l'origine des données, l'altération des données est fortement déconseillée. En effet, il est difficile dans ce cadre de savoir quelles approches d'augmentation des données n'altèrent pas la réalité physique ayant généré les signaux, et ce même en utilisant des techniques d'interpolation avancées [2, 7, 3]. En outre, il existe un risque accru de surapprentissage dans le cas du suréchantillonnage. Pour ces raisons, le suréchantillonnage sors du cadre de cette étude.

Les travaux sur la fonction de coût sont spécifiques au modèles de classification qui utilisent des réseaux de neurones. Pour faire face aux jeux de données déséquilibrés, il est possible de (i) pondérer les classes par l'inverse de leur fréquence d'apparition [8, 13] ; (ii) pondérer les classes par le nombre effectif d'échantillons [4] ; et (iii) d'utiliser des fonctions de perte alternatives, par exemple la perte focale [10]. Ces trois stratégies seront étudiées.

La pondération par le nombre effectif d'échantillon affine la pondération classique en prenant en compte le chevauchement potentiel des informations : c'est-à-dire que chaque échantillon supplémentaire n'apporte pas la même quantité d'information. En pratique, la fonction de coût de l'entropie croisée par classe

est pondérée par $\frac{1-\beta}{1-\beta^n}$, avec n le nombre d'échantillons et β un hyperparamètre (entre 0.9 et 0.999). La fonction de coût de perte focale adopte une autre stratégie et vise à augmenter le poids des échantillons difficiles à apprendre ou des valeurs aberrantes. L'entropie croisée est alors pondérée par le facteur de modulation suivant : $-\alpha_t(1-p_t)^\gamma$, avec le coefficient d'équilibrage α et $\gamma \geq 0$ comme paramètre de focalisation. On notera que les valeurs des fonctions de coût entre les différentes stratégies ne sont pas comparables en raison des différents poids appliqués.

2 Jeu de données

Le jeu de données considéré pour cette étude est construit à partir d'enregistrements sismiques du réseau RESIF RD [6] composé de 43 stations d'enregistrement en France et du jeu de données STEAD [11]. STEAD est un jeu de données sismiques mondial normalisé pour les applications machine learning comportant des stations situées dans des endroits similaires à ceux du réseau RESIF RD. Les données sont acquises par des capteurs à trois composantes (X, Y, Z), seule la composante verticale contenant l'information recherchée est conservée. Les données considérées sont acquises sur 32 mois, du 01/04/2019 au 23/11/2021. La plupart des canaux ont une fréquence d'échantillonnage de 50 Hz. Les données acquises à fréquence d'échantillonnage supérieure (jusque 100Hz) sont filtrées et sous-échantillonnées à 50Hz afin d'éviter les biais dans le jeux de données final. Dans l'étude actuelle, nous distinguons six classes, à savoir les Tremblements de Terre (TT), les Explosions de Carrières (EC), les Explosions Marines (EM), les Tremblements de Terre sans magnitude définissable ($\mathrm{TT_{inc}}$), les événements Induits Présumés (IP) et le Bruit (B). Bien que IP et $\mathrm{TT_{inc}}$ soient également des tremblements de terre, nous avons décidé de leur consacrer des classes distinctes pour les raisons suivantes : Les IP sont des tremblements de terre que l'on soupçonne d'être induits par l'activité industrielle et qui peuvent donc se produire à des endroits spécifiques avec des propriétés physiques spécifiques. De même, les $\mathrm{TT_{inc}}$ sont des tremblements de terre d'une magnitude indéfinissable. Cela peut être dû soit à la proximité des stations, soit au fait qu'ils sont trop petits pour être calculés. Dans le premier cas, cela signifie que la physique du champ proche peut être présente et, par conséquent, qu'ils peuvent être physiquement différents.

Les cinq classes positives sont issues du réseau RESIF RD. Les échantillons de bruit (classe négative) sont obtenus à du réseau STEAD, assurant qu'aucune classe positive ne soit présente dans les fenêtres de bruit.

Pour transformer les données brutes en un jeu de données de travail pour CNN, les étapes de pré-traitement suivantes sont appliquées aux enregistrements continus dans l'ordre indiqué : 1) suppression de la moyenne ; 2) application de la réponse de l'instrument ; 3) filtre passe-bande Butterworth à phase nulle, d'ordre 4 entre 1 et 10 Hz pour limiter le bruit d'origine humaine ; 4) rééchantillonnage à 50 Hz ; 5) extraction de fenêtres de 45 secondes (sur la base des connaissances métier) ; 6) normalisation maximale des échantillons entre 0 et 1 ; 7) division du jeu de données en un ensemble d'entraînement (70%), de validation (10%) et de test (20%) après brassage aléatoire des échantillons pour obtenir une indépendance spatiale et temporelle. Ce dernier est nécessaire car les données sismiques varient en fonction de l'heure (de la journée et de la saison) et de l'endroit. L'ensemble de validation est utilisé pour la sélection du modèle. Le jeu de données final est présenté dans le tableau 1, le déséquilibre des classes est nettement visible.

3 Cadre Expérimental

Modèle de classification. Nous étudions l'efficacité des stratégies de gestion des jeux de données déséquilibrés mentionnés ci-dessus à l'aide d'un CNN simple.

Le réseau comporte trois couches convolutives suivies de trois couches denses. Les couches convolutives comportent 32 filtres de taille 3 se déplaçant avec un pas de 1 et étant activés par une fonction *ReLU*. Un *zero padding* est appliqué dans les convolutions. Le *Maxpooling*, avec un filtre de taille 2, est appliqué après une couche convolutive sur deux. Les trois couches denses forment la partie de classification du réseau. À l'aide d'une fonction softmax, nous obtenons une probabilité pour chacune des classes. Au cours de l'apprentissage, les données sont présentés au réseau par *batch* de 500 échantillons. Le callback EarlyStopping de Tensorflow est utilisé pour économiser du temps et des coûts de calcul si aucune amélioration de la perte n'est obtenue après 40 *epochs* consécutives. Deux optimiseurs, à savoir Adam et Root Mean Squared Propagation (RMSprop) associés à une régularisation L1L2 ont été comparés. Les performances et comportement étant similaires, seul les résultats sur Adam sont présentés (voir Sec. 4).

Métriques et procédure d'évaluation. Face à un problème de déséquilibre des classes (multi-classes), l'*accuracy* simple est fortement impactée par la classe majoritaire et n'est donc pas une métrique suffisante pour évaluer les performances globales des modèles. La *balanced accuracy* pour la classification multi-classes qui est définie comme la moyenne du *recall* obtenu pour chaque classe i [1] est une alternative, et est utilisée en tant que métrique principale :

$$BA = \frac{\sum_{i=1}^{m} Recall_i}{m} \quad \text{et} \quad Recall_i = \frac{TP_i}{TP_i + FN_i}$$

où m représente le nombre total de classes, TP_i le nombre de vrais positifs (*True Positive*) et FN_i le nombre de faux négatifs (*False Negative*) pour la classe i. Les bias du modèles apparaissent sur les métriques par classes : la *precision* et le *recall*. Ces métriques sont donc également considérées, ainsi que les matrice de confusion (CM). Sans être présentées ici, les courbes d'apprentissage (*epochs* versus perte) des données d'apprentissage et de validation sont contrôlées pour la sélection des modèles, l'arrêt précoce, le contrôle du sur-apprentissage et du sous-apprentissage.

4 Expériences & Résultats

Sans aucune stratégie de compensation du jeu de données déséquilibré, les performances du modèle sont faibles avec une BA de 0.17, et biaisés : le modèle ne prédit que les tremblements de terre et aucune autre classe. Voir la 1ère colonne du tableau 2 où le *recall* est maximum pour les tremblements de terre et nul pour les autres classes.

Chacune des quatre stratégies proposées augmente de manière significative les performances du modèle. L'ensemble des résultats est présenté dans les colonnes 2 à 5 du tableau 3. On notera que les meilleurs résultats sont obtenus par

TABLE 1 : Vue d'ensemble du jeu de données déséquilibré avec les échantillons par classe pour le total et les jeux d'entrainement, de validation et de test.

Classe		Nombre d'échantillons				
Nom complet	Abréviation	Total	Proportion	Entraînement	Validation	Test
Tremblement de terre	TT	61345	76.70%	44756	6384	10205
Explosion de carrière	EC	5957	7.40%	4328	641	988
Explosion marine	EM	4886	6.20%	3592	498	796
Tremblement de terre de magnitude inconnue	TT_{inc}	3918	4.80%	2822	405	691
Événements induits présumés	IP	2264	2.80%	1640	243	381
Bruit	B	1736	2.10%	1221	167	348

TABLE 2 : Vue d'ensemble du *recall* et de la *precision* par classe à l'aide de l'optimiseur d'Adam dans les cas suivants : non prise en compte du jeu de données déséquilibré (« naïf » ; colonnes 1-2), rééchantillonnage de l'ensemble d'entraînement (colonnes 3-4), ajustement de la perte par pondération de la classe par l'inverse de la fréquence d'échantillonnage (colonnes 5-6), pondération de la classe par le nombre effectif d'échantillons (colonnes 7-8) et perte focale (dernières colonnes).

	Naïf		Rééchantillonnage		Fréquence d'apparition inv.		Nombre effectif		Perte focale	
	Recall	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*
TT	100	76.57	53.24	88.89	66.43	91.52	94.31	86.21	93.98	82.24
IP	0	.	19.34	26.40	53.50	22.69	37.45	54.82	13.17	62.75
TT_{inc}	0	.	79.75	19.90	66.67	27.69	29.88	49.39	17.53	40.11
EC	0	.	71.76	33.41	75.51	53.90	69.42	77.53	30.73	59.52
noise	0	.	66.47	34.37	56.29	40.69	46.71	66.67	44.31	55.64
EM	0	.	61.04	30.01	69.08	33.50	34.34	67.86	34.54	49.14

TABLE 3 : Vue d'ensemble des métriques globales pour les quatre stratégies de traitement des jeux de données déséquilibrés (colonnes 2 à 5) et pour le cas de non prise en compte du jeu de données déséquilibré (colonne 1).

Métrique globale	Naïf	Rééchantillonnage	Fréquence d'apparition inv.	Nombre effectif	Perte focale
Balanced Accuracy	0.17	0.59	**0.64**	0.52	0.39
Accuracy	0.77	0.56	0.67	0.83	0.79
F1-score	0.76	0.6	0.70	0.81	0.76

l'ajustement de la fonction de perte en pondérant chaque classe par l'inverse de sa fréquence d'apparition. La *balanced accuracy* monte à 0.64. Cette observation se maintient en changeant d'optimiseur. En outre, le nombre d'échantillons mal classés en tant que TT est le plus faible, comme le montre la precision la plus élevée pour TT (6e colonne du tableau 2).

Compte tenu de la BA de 0.59, la deuxième meilleure option consiste à rééchantillonner le jeu de données. Dans notre expérience, nous avons réduit le nombre d'échantillons TT de 90%. La taille de la classe TT est ainsi du même ordre de grandeur que celle des autres classes. Malheureusement, nous observons toujours un biais en faveur de la classe TT pour les échantillons IP avec un *recall* de seulement 19.34. En outre, les TT réels sont souvent mal classés dans d'autres classes telles que EC, d'où un *recall* de 53.24 pour les TT.

Les deux autres stratégies, pondération du nombre effectif d'échantillons et perte focale, sont moins performantes avec des valeurs de BA de 0.53 et 0.39, respectivement. Par ailleurs, les valeurs de *recall* et de *precision* montrent que le bruit et l'EM sont souvent mal classés en tant que TT pour les deux stratégies. Une normalisation des *batchs* [9] était nécessaire dans les deux cas, sinon le modèle aurait prédit uniquement le TT, comme le modèle « naïf ». Pour calculer les poids dans la pondération du nombre effectif d'échantillons, nous avons utilisé β=0.999. Dans l'application de la perte focale, plusieurs valeurs pour γ et α_t sont testées pour ajuster le niveau de focalisation sur les échantillons difficiles à apprendre. Nous avons constaté que les meilleures valeurs pour notre configuration sont $\gamma = 2$ et $\alpha_t = 0.5$. Néanmoins, la prédiction du modèle reste sous-optimale.

En comparant les performances des différents optimiseurs, nous avons constaté que la plupart des résultats sont similaires quelque soit l'optimiseur. Cependant, nous avons observé que la performance est différente dans le cas de la perte focale, le BA étant plus élevé de 0.1 lorsque l'on utilise RMSprop comme optimiseur par rapport à Adam. Nous avons également remarqué une différence de vitesse d'apprentissage entre les différentes stratégies, la perte focale et la pondération par le nombre effectif d'échantillons étant plus rapides que le

rééchantillonnage et la pondération en fonction de la fréquence d'échantillonnage. Toutefois, cette différence est en partie due à la normalisation des batchs [9, 12].

Après avoir trouvé la meilleure stratégie pour traiter les jeux de données déséquilibrés pour une architecture simple, nous avons poursuivi la recherche d'hyperparamètres afin d'obtenir le meilleur modèle final. Nous avons effectué une recherche en grille (*grid search*) sur les paramètres suivants : optimiseur, taux d'apprentissage, taille des *batchs*, nombre de couches convolutives, nombre de filtres, taille des filtres, fréquence de maxpooling, taux d'abandon (*dropout rate*) et taux de dilatation. Nous pouvons en conclure que le meilleurs modèle est plus profond (avec 9 couches convolutives) que le modèle initial, avec moins de filtres par couche et une normalisation des *batch* appliquée pour une convergence plus rapide [12]. Nous avons testé à nouveau les quatre stratégies décrites ci-dessus en utilisant l'architecture finale. Dans ce cas également, des performances supérieures sont obtenues en appliquant des pondérations de classe en fonction de l'inverse de la fréquence de la classe. L'optimisation des hyperparamètres améliore le BA de 0.64 à 0.78.

5 Conclusion

Dans les tâches de classification avec un jeu de données dont la taille des classes est déséquilibrée, le biais du modèle peut être réduit en traitant efficacement le déséquilibre du jeu de données. Dans cette étude, nous avons comparé quatre stratégies qui traitent des jeux de données déséquilibrés en termes de taille de classe. Pour notre application sismique, où nous avons un rapport de taille de classe de 1 :10 entre les classes minoritaires et majoritaires, nous avons constaté que la repondération de la perte par rapport à l'inverse de la fréquence de la classe fonctionne le mieux. Par rapport à la non prise en compte du déséquilibre des classes, cela a conduit à une amélioration de la *balanced accuracy* de 0.47, pour un BA de 0.64 (0.78 après optimisation du modèle). Le rééchantillonnage du jeu de données en sous-échantillonnant la classe majoritaire était la deuxième meilleure option. Les stratégies les plus sophistiquées, à savoir la perte focale et la pondération des classes par le nombre effectif d'échantillons, ont donné les résultats les moins favorables. Bien qu'en fin de compte, chacune des quatre stratégies aboutisse à un modèle plus performant que si l'on n'envisageait pas du tout d'ajuster le déséquilibre des classes. Même pour un modèle dont l'architecture est optimisée, notre conclusion est valable.

Il existe d'autres moyens d'améliorer les résultats de la classification, l'un d'entre eux est la mise en œuvre d'une détection d'anomalies. Nous prévoyons donc d'ajouter par la suite un module de détection des anomalies afin de réduire les erreurs de classification des données « inconnues ».

Références

[1] Kay Henning BRODERSEN, Cheng Soon ONG, Klaas Enno STEPHAN et Joachim M BUHMANN : The balanced accuracy and its posterior distribution. *In 2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

[2] Nitesh V CHAWLA, Kevin W BOWYER, Lawrence O HALL et W Philip KEGELMEYER : SMOTE : synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] Baiyun CHEN, Shuyin XIA, Zizhong CHEN, Binggui WANG et Guoyin WANG : RSMOTE : A self-adaptive robust SMOTE for imbalanced problems with label noise. *Information Sciences*, 553:397–428, 2021.

[4] Yin CUI, Menglin JIA, Tsung-Yi LIN, Yang SONG et Serge BELONGIE : Class-balanced loss based on effective number of samples. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[5] Chris DRUMMOND, Robert C HOLTE et OTHERS : C4. 5, class imbalance, and cost sensitivity : why under-sampling beats over-sampling. *In Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8, 2003.

[6] Réseau Sismologique et géodésique FRANÇAIS : CEA/DASE Broad-Band Permanent Network in Metropolitan France [Data set]. 2018. Publisher : RESIF—Réseau Sismologique et géodésique Français,.

[7] Hui HAN, Wen-Yuan WANG et Bing-Huan MAO : Borderline-SMOTE : a new over-sampling method in imbalanced data sets learning. *In Advances in Intelligent Computing : International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, pages 878–887. Springer, 2005.

[8] Chen HUANG, Yining LI, Chen Change LOY et Xiaoou TANG : Learning deep representation for imbalanced classification. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[9] Sergey IOFFE et Christian SZEGEDY : Batch normalization : Accelerating deep network training by reducing internal covariate shift. *In International conference on machine learning*, pages 448–456. PMLR, 2015.

[10] Tsung-Yi LIN, Priya GOYAL, Ross GIRSHICK, Kaiming HE et Piotr DOLLÁR : Focal loss for dense object detection. *In Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[11] S Mostafa MOUSAVI, Yixiao SHENG, Weiqiang ZHU et Gregory C BEROZA : STanford EArthquake Dataset (STEAD) : A global data set of seismic signals for AI. *IEEE Access*, 7:179464–179476, 2019. Publisher : IEEE.

[12] Shibani SANTURKAR, Dimitris TSIPRAS, Andrew ILYAS et Aleksander MADRY : How does batch normalization help optimization ? *Advances in neural information processing systems*, 31, 2018.

[13] Yu-Xiong WANG, Deva RAMANAN et Martial HEBERT : Learning to model the tail. *Advances in neural information processing systems*, 30, 2017.