Proceedings of Machine Learning Research 101:822–835, 2019 ACML 2019

Focused Anchors Loss: cost-sensitive learning of discriminative features for imbalanced classification

Bahram K. Baloch* K152817@NU.EDU.PK
Sateesh Kumar* K152821@NU.EDU.PK
Sanjay Haresh* K152179@NU.EDU.PK
Abeerah Rehman K152840@NU.EDU.PK
Tahir Syed TAHIR.SYED@NU.EDU.PK
National University of Computer and Emerging Sciences

Abstract

Deep Neural Networks (DNNs) usually suffer performance penalties when there is a skewed label distribution. This phenomenon, class-imbalance, is most often mitigated peripheral to the classification algorithm itself, usually by modifying the amount of examples per class, for oversampling at the expense of computational efficiency, and for undersampling at the expense of statistical efficiency. In our solution, we combine discriminative feature learning with cost-sensitive learning to tackle the class imbalance problem by using a two step loss function, which we call the Focused Anchors loss (FAL). We evaluate FAL and its variant, Focused Anchor Mean Loss (FAML), on 6 different datasets in comparison of traditional cross entropy loss and we observe a significant gain in balanced accuracy for all datasets. We also perform better than time-costly re-sampling and ensemble methods like SMOTE and Near Miss in 4 out of 6 datasets across F1-score, AUC-ROC and balanced accuracy. We also extend our evaluation to image domain and use long-tailed CIFAR10 to evaluate our loss function where we consistently report significant improvement in accuracy. We then go on to test our loss function under extreme imbalance on a propriety dataset and achieve a gain of 0.1 AUC-ROC over the baseline.

Keywords: imbalanced classification, discriminative learning, cost-sensitive learning, deep learning

1. Introduction

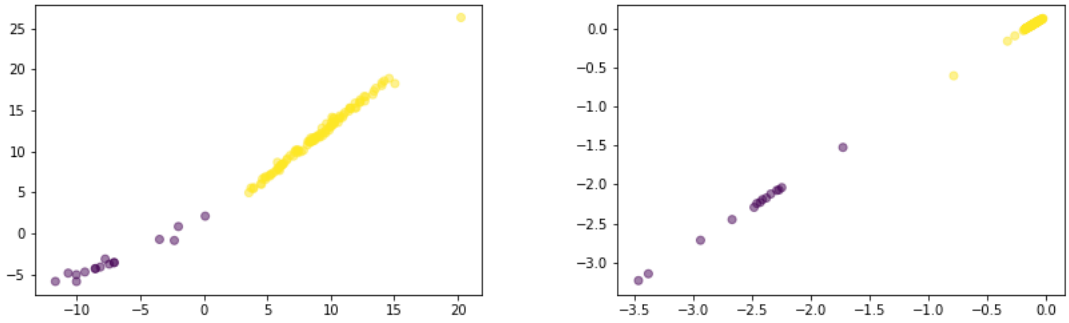
In recent years, Deep Neural Networks (DNNs) have led the benchmarks on a variety of problems, such as image classification, image segmentation and object detection. However, part of the success on these benchmarks can safely be attributed to the inherent uniformity of the large-scale datasets available in the stated domains. In practice, these models tend to struggle on datasets with skewed distributions of labels, which are very common in real world applications such as fraud detection, anomaly detection, face recognition and medical imaging.

A number of techniques have been proposed to remedy this. The most common and widely used techniques are re-sampling and cost-sensitive methods. *Re-sampling* methods Liu et al. (2009); Bowyer et al. (2011) balance the data distribution by either oversampling

. *The first three authors contributed equally

the minority class or undersampling the majority class. Oversampling methods have several disadvantages. Firstly, they are computationally expensive due to the increase in number of data points. Secondly, they are more prone to over-fitting due to the duplication of data points. Undersampling methods, on the other hand, may end up throwing away important data points. *Cost-sensitive* methods take a different approach to deal with imbalance by assigning weights to the data points in accordance to their class proportion in the dataset. However, recently, cost-sensitive methods that are class-agnostic have been gaining traction such as Lin et al. (2017). These methods assign weights according to the "hardness" to classify of a data point.

Huang et al. (2016) observed that minority class generally has less examples with high variability. Due to this, the number of "imposter" nearest neighbours, which should have lied on the different side of a decision boundary, increase for the minority class, this can be seen in figure 1a. *Thus, the problem of class imbalance is two-fold: (1) imposter neighbours and (2) hard examples.* Although cost-sensitive learning caters for hard examples, it does not deal with the problem of imposter neighbours. To deal with this problem, we use discriminative feature learning i.e. explicitly constraining the network to learn distinct representations of data Wen et al. (2016).



(a) Using Cross-Entropy.

(b) Using FAL.

Figure 1: Visualization of the embedding space using cross entropy and FAL loss functions on MNIST dataset after synthetically inducing imbalance in it. We observe that our proposed method has the effect of creating a well-separated cluster for each class in the embedding space. It is also visible that the feature vectors for the minority class have significantly less intra-class variance.

In our solution, we propose to combine discriminative and cost-sensitive learning to better model imbalanced datasets. We handle the two problems using a two stage loss function. In the first stage, following a similar thought as Huang et al. (2016), we use a variation of Large Margin Gaussian Mixture Loss (LGM) Wan et al. (2018) to increase the inter-class variance and reduce the intra-class variance in the learned feature vectors. Then, we apply Focal Loss Lin et al. (2017) over the obtained class probabilities to handle the hard examples. As is visible from Fig. 1b, our loss leads to well separated clusters of each class in the embedding space with a very small cluster for the minority class. This palliates the intrusion by imposter neighbours when compared with the "de facto" loss - cross entropy in

Fig. 1a. In order to further demonstrate the effectiveness of our loss function, we evaluate it on six highly imbalanced fraud detection datasets along with long-tailed CIFAR10.

Contributions This work belongs in a smaller class of recent approaches that change the way over-represented and under-represented examples are viewed. The principal contributions are:

- Combining discriminative feature learning with cost-sensitive learning to tackle the class imbalance problem, which to the best our knowledge, has not been utilized before.
- Synthesizing a two-stage loss function using a variation of LGM and Focal Loss. We call this loss the Focused Anchor Loss (FAL).

2. Literature Review

Previous attempts at mitigating class imbalance can be classified into two main approaches : Re-sampling Methods and Cost-Sensitive Learning.

2.1. Re-sampling Methods

The re-sampling methods neutralize the problem of class imbalance by either oversampling the minority class or undersampling the majority class. The general problem with oversampling is that it only replicates the data and no new information is incorporated in the dataset. Smote Bowyer et al. (2011) is a well known algorithm in this regard, as it works by generating new minority class data points through interpolation. Several variations of Smote have been proposed,some of these are described in Batista et al. (2004). He et al. (2008) is another re-sampling based algorithm which works by giving weight to minority classes according to the difficulty in learning, and thus more data is generated for the class which is harder to learn. However, since oversampling leads to increased computation cost, undersampling is often preferred over oversampling. In fact, Drummond and Holte (2003) experimentally proves that undersampling methods generally perform better than oversampling methods. Another general class of algorithms try to learn the underlying distribution of data and try to generate new data-points belonging to the minority class.Mariani et al. (2018) used Generative adversarial Networks (GANs) for generating examples of minority class in order to tackle imbalanced datasets.

2.2. Cost-Sensitive Learning

In cost-sensitive learning instead of modifying the distribution of the training data, the classes are weighted differently in the loss function to tackle the imbalance problem in the algorithm itself. There are two ways to implement cost-sensitive learning. One way is to assign cost to data points according to the proportion of data points belonging to its class Zadrozny et al. (2003) and the other way is to modify the loss function so that mistakes on minority class are heavily penalized essentially fitting the class sensitive framework to the classifier itself. Cao et al. (2013) follows the first paradigm and presents a weighted form of SVM classifier. Bahnsen et al. (2015) combine the idea of cost-sensitive learning with

decision trees and presents a cost-sensitive measure for pruning a subtree of the decision tree. Khan et al. (2018) uses a learn-able cost matrix for weighting the classes. The paper argues that weights of cost matrix should be selected based on the distribution of classes and not the frequency. It provides with a cost function for cost matrix which uses class separability and histogram of classes. Focal loss Lin et al. (2017) is a unique loss function which penalizes hard examples in a dataset, fairly assuming that the hard examples will belong to the minority class.

2.3. Discriminative Feature Learning

Discriminative feature learning is another class of algorithms that warrant a place here. Although, discriminative feature learning is not specifically applied to data imbalance but it has been shown to produce great results in certain works. Huang et al. (2016) introduces the idea of reducing the bias of class boundary by constraining the network to respect inter-class and intra-class margins over clusters of classes. They introduce quintuplet loss which is an extension to the triplet loss Schroff et al. (2015). Wan et al. (2018) introduces another loss function which works by learning mixture of Gaussian distribution over classes in the dataset and imposing a large margin to maximize the inter-class variance.

3. Methodology

In this section, we synthesize a loss function for imbalanced classification by combining the discriminative prowess of Large-Margin Gaussian Mixture Loss Wan et al. (2018) and the cost-sensitivity of Focal Loss Lin et al. (2017). We explain both the techniques before introducing our loss function.

3.1. Focal Loss

Focal loss is designed to deal with highly imbalanced datasets. The loss function uses hard examples as a proxy to solving class imbalance. Hard examples are those examples which have large errors i.e the model miss-classifies them with high confidence. They introduce a modulating factor which decays the error contribution from easy examples to prevent them from overwhelming the loss function. This effectively focuses the model to hard examples.

As shown in figure 2, the incurred cost on positive examples for which the algorithm is already predicting a probability greater than 0.5 is not substantial but when summed over all the data points in the dataset, this loss overwhelms the loss from hard examples. In focal loss this cost is reduced significantly.

Focal loss is defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

where p_t is defined as:

$$p_t = \begin{cases} p & y = 1 \\ 1-p & y = 0 \end{cases}$$

In Eq. 1, γ acts as the modulating factor. As shown in figure 3, the higher the value of γ, the lesser the cost incurred by well classified examples. And α_t is defined as,

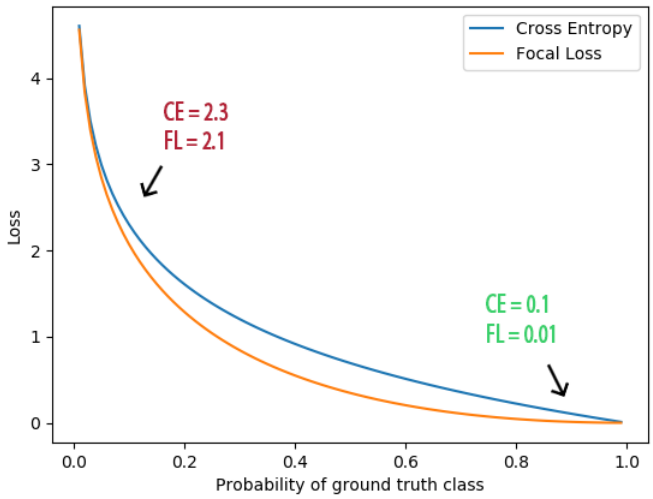


Figure 2: Comparison of Focal loss(FL) and Cross Entropy. The difference between Focal loss and Cross Entropy is the cost incurred on examples which are being classified correctly (Easy Examples). As shown, the cost incurred by FL when the classifier outputs 0.9 for a positive example is only 0.01 while for Cross entropy it is 0.1. This decrease in incurred cost makes the classifier to focus on examples which are not being classified correctly (Hard Examples)

$$\alpha_t = \begin{cases} \alpha & y = 1 \\ 1 - \alpha & y = 0 \end{cases}$$

where $\alpha \in [0, 1]$ is a weighting factor for error contribution from majority and minority classes. α-balanced loss works better in practice than the non-α-balanced loss.

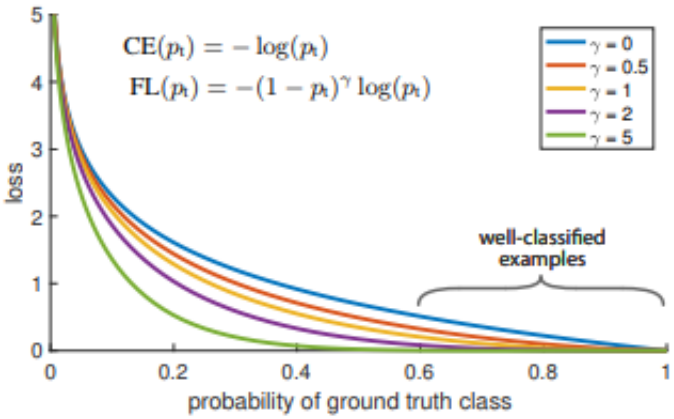


Figure 3: Focal loss: impact of hyperparameter γ.

3.2. Large Margin Gaussian Mixture (LGM) Loss

Large Margin Gaussian Mixture (LGM) Loss is a discriminative loss function which assumes a Gaussian Mixture distribution over the embedding space. It fits a Gaussian for each class k with mean μ_k and covariance Σ_k. The probability of the extracted feature x_i given the corresponding class label $z_i \in [1, K]$ can be expressed as in Eq. 2,

$$p(x_i|z_i) = \mathcal{N}(x_i, \mu_{z_i}, \Sigma_{z_i}) \tag{2}$$

Similarly, the corresponding posterior probability can be expressed as in Eq. 3.

$$p(z_i|x_i) = \frac{\mathcal{N}(x_i; \mu_{z_i}, \Sigma_{z_i})p(z_i)}{\sum_{k=1}^{K} \mathcal{N}(x_i; \mu_k, \Sigma_k)p(k)} \tag{3}$$

where $p(k)$ is the prior probability of class k. Maximizing the above probability, for the true class, can act as a classification loss but does not encourage separation among the gaussians. To fix this, a margin m is added to the mahalanobis distance computed for $\mathcal{N}(x_i; \mu_{z_i}, \sigma_{z_i})$ which enforces large margin among the gaussians in the embedding space. Cross-Entropy is then computed over the posterior probability distribution and the one-hot labels as shown in Eq. 4. An extra regularization term is added known as the likelihood regularization term given in Eq. 5 which penalizes the distance of the feature vector x_i and the class mean μ_{z_i} resulting in tighter class distribution and more space for the large margin. The final formulation of the LGM loss is given in Eq. 6

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^{N} \log p(z_i|x_i) \tag{4}$$

$$\mathcal{L}_{lkd} = -\sum_{i=1}^{N} \log \mathcal{N}(x_i; \mu_{z_i}, \Sigma_{z_i}) \tag{5}$$

$$\mathcal{L}_{GM} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{lkd} \tag{6}$$

3.3. Focused Anchors Loss (FAL)

We aim to leverage the discriminative power of LGM and the cost-sensitivity of Focal Loss to solve the data imbalance problem. We present the formulation of our two-stage loss for the binary class problem for simplicity. In the first stage, we get the probabilities of each class as done in LGM and in the second stage we apply Focal Loss over these probabilities to get the final loss. In order words, we have backed focal loss into LGM. Below we first describe the modifications that we have made to LGM loss and then present the final form of our loss function.

LGM is aimed at explicit probabilistic modeling of the learned features. But, since we only want the discriminative aspect of the loss function, we drop the probabilistic perspective. Under our modification, LGM becomes a simple euclidean distance between feature vectors x_i and the means, we call them *anchors*, ϕ_i and the probability $p(z_i|x_i)$ is calculated by taking a softmax over negative of the euclidean distance from each of the anchors ϕ_i. The likelihood regularization term also becomes the sum of distance between feature vector

and corresponding class anchors. But, because we are dealing with class imbalance, the regularization residual from majority class may overwhelm the loss and the network may not pay any attention to the minority class examples. Therefore, we take the means of the distances weighted by the respective class ratios as given in Eq. 8.

$$p(z_i|x_i) = \frac{e^{-\left\|F(x_i)-\phi_{z_i}\right\|_2^2 - 1\{j=z_k\}m}}{\sum_{k=1}^{K} e^{-\|F(x_i)-\phi_k\|_2^2 - 1\{k=z_k\}m}} \tag{7}$$

$$\mathcal{L}'_{lkd} = \frac{1}{K}\sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \mathbb{1}\{z_i = k\} \left\|x_i - \phi_{z_i}\right\|_2^2}{\sum_{i=1}^{N} \mathbb{1}\{z_i = k\}} \tag{8}$$

We now present the formulation of our loss function. We first use Eq. 7 to find the class probabilities $p(k|x_i)$ for each feature vector x_i from the neural network. Instead of computing cross-entropy over these probabilities and the one-hot labels, we use focal loss to maximize $p(z_i|x_i)$. The modified likelihood regularization in Eq. 8 is then added to the loss from focal loss as in Eq. 9.

$$\mathcal{FAL} = -\alpha(1-p_t)^{\gamma}\log(p_t) + \beta\mathcal{L}'_{lkd} \tag{9}$$

where p_t is defined as:

$$p_t = \begin{cases} p(z_i|x_i) & y = 1 \\ 1 - p(z_i|x_i) & y = 0 \end{cases}$$

and β is a hyper-parameter used to control the likelihood penalty.

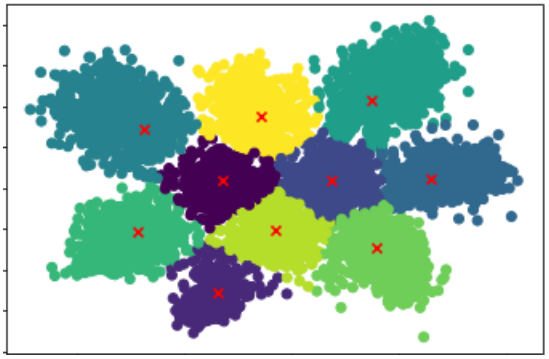


Figure 4: Visualization of the embedding space on the MNIST dataset. The cross represents the class anchors. Each anchor is approximately the mean of the feature vectors of the corresponding class.

3.4. Focused Anchors (Mean Update) Loss (FAML)

We propose another variation of the FAL. We observe that the Maximum Likelihood Estimate of the mean of the class anchors is essentially the mean of the feature vectors belonging to that class in the embedding space. This can be seen very easily from figure 4. So, Instead of learning the anchors, we run them through a hand-crafted update schedule. This is done

by updating the anchor of each class to the mean of the feature vectors $F(x_i)$ being generated for that class. This makes the anchors a better representative of their respective class in the embedding space. This also helps the network to learn as there is one fewer thing to model. Therefore, we achieve a slight increase in performance over FAL. The update rule is given below :

Class anchors are updated as:

$$\phi_k^i = \lambda * \mu_k + (1 - \lambda) * \phi_k^{i-1} \tag{10}$$

Where μ_k is the mean of feature vectors belonging to kth class and λ is a hyperparameter which controls how much importance shall be given to the means. In our experiments, we initialized it to 0.9 and decay it quadratically.

The mean for each class μ_k is computed as:

$$\mu_k = \sum_{i=1}^{n} ar^i \times 1\{z_i == k\} F(x_i) \tag{11}$$

where n is the number of steps we take before updating the anchor, z_i is the class label of the ith example. a and r are hyperparameters which have been introduced in order to model the relative importance of feature vectors as a geometric progression. Since we want to keep the sum of the geometric series to be 1, we compute a as:

$$a = \frac{1 - r^n}{1 - r} \tag{12}$$

4. Experiments

In order to evaluate our proposed loss function we have used two sets of experiments. First one is based on Fraud Detection whereas the second one is based on Image Classification.

4.1. Fraud Detection

Class imbalance is very common in the domain of Fraud Detection. Therefore, we use six highly imbalanced fraud detection datasets to evaluate our loss function.

4.1.1. Experimental Setup

We use a 2-layer network[1], the 1st layer consists of 16 neurons and the 2nd layer consists of 8 neurons. We use Adagrad optimizer with varying learning rate for each dataset. We compare the performance of our loss function with that of Cross entropy as well as Smote and Near Miss. We have used a simple architecture because the datasets used contain only a small number of features and large architectures did not add much in terms of generalization.

Below we describe each of the techniques used in the experiments:

- **Cross-Entropy (CE):** This is the baseline model where we use baseline Cross-Entropy loss.
- **Focal Loss (FL):** Under this setting, we replace Cross-Entropy loss with Focal Loss. The hyperparameters for Focal loss were cross-validated using grid search.

1. We use a small network because of the lower-dimensionality of numerical data

Dataset	Size	Attributes	Ratio
Credit Card	284,807	24	1:578
Home Credit	307511	344	1:11
PaySim	6362620	344	1:737
SatImage	6435	36	1:10
CC Fraud	10000000	8	1:15
Give Me Some Credit(GMSC)	150000	11	1:14

Table 1: Description of the datasets.

- **FAL:** The Cross-Entropy is replaced with the loss proposed in 3.3.The hyperparameter β was set to 0.1 in all experiments.
- **FAML:** The FAML loss proposed in 3.4 is used instead of FAL. The hyperparameter r is set to 1.3 and λ is set to 0.999 while the number of steps n was set to 200 across all experiments.
- **CE+SMOTE:** We combine baseline cross entropy with an over-sampling method, SMOTE Bowyer et al. (2011).
- **CE+NEAR-MISS:** Baseline cross entropy is combined with an under-sampling method, NEAR-MISS Zhang and Mani (2003)
- **FL+LGM:** For sake of completeness we also applied focal loss over LGM Wan et al. (2018).

Each model was run on each dataset. The number of epochs is set to 50 for all datasets.

4.1.2. Evaluation Criteria

Accuracy is not considered to be a good evaluation measure for class imbalanced datasets as the majority class overwhelms the errors on minority class. Hence in this work we use the F1-score, AUC-ROC score and balanced accuracy as the evaluation criteria. These measures have been defined in Drummond and Holte (2003). However, AUC-ROC is not considered to be a good metric in cases where positive is the minority class Davis and Goadrich (2006), which is the case with all our datasets. Therefore, the primary measures in our evaluations are F1-Score and balanced accuracy.

4.1.3. Datasets

Description of datasets have been provided in Table 1. CreditCard, HomeCredit, PaySim, Give Me Some Credit(GMSC) and CCFraud are taken from Kaggle.com while SatImage is taken from UCI Machine Learning Repository.

4.1.4. Results with holdout cross validation

Results from these experiments are summarized below in table 2. We use holdout validation strategy with 80% data in the training set and 20% data in the test set. From the results we observe that both FAML and FAL loss functions produce better AUC, F1-score and balanced accuracy than Cross Entropy as well as Focal Loss *on all datasets*. For instance, on the credit card dataset, FAML achieves a gain of 0.04 F-1. On the SatImage dataset, FAL achieves a gain of 3% balanced accuracy. We also observe a significant increase in balanced accuracy for all datasets, which indicates that our proposed algorithm is able to model the minority class much better than cross entropy. Furthermore, although both Near-Miss and SMOTE are time costly methods, without an equivalent gain in their predictive performance. In fact, both FAL and FAML outperform Near-Miss and SMOTE on 4 out of the 6 datasets used across all metrics.

4.1.5. Results with k-fold cross validation

In order to obtain a more robust estimate of the minority class distribution and to make sure that each of the data point becomes the part of the test set at least once, we have also used 10-fold nested cross-validation. Overall, we observe that each method's performance decreases, this can be attributed to the fact the test is effectively increased by a factor of 10. The results are given in table 3. Since, Paysim is the largest dataset in our experiments, we used FAL as well as FAML on Paysim. We observe that our proposed loss function works considerably better than cross entropy, a gain of 0.18 f1 on PaySim and a slight gain in f1 on all of the other datasets.

4.2. Image Classification

We use an imbalanced version of CIFAR10 i.e. long-tailed CIFAR10 for evaluating our loss function. In this dataset, the number of training examples per class are reduced by using an exponential function $n = n_i \times \mu_i$ where i is the class index and n_i is the original number of samples and μ is the imbalance factor. The number of examples with respect to the imbalance factor is shown in Figure 5. Cui et al. (2019) from Alphabet inc. came out during the writing of this paper. Therefore, for the sake of completeness we benchmark our loss against their work on CIFAR10. We under-perform by a small margin but our loss can be combined with their weighting scheme and a thorough analysis is required which could not be completed due to time constraints.

4.3. Results on extreme imbalance

We also test our loss function on a large scale real world propriety dataset of suspicious activity reporting (SAR) by a commercial bank. The problem was a binary classification problem and the dataset comprised of 10 million records and an imbalance factor of 1:2222. The results are given in Table. 5. From the results, we observe that our method performs considerably better than the other three techniques used. We acheive a gain of 0.1 AUC over decision tree with Smote and a gain of 0.02 AUC over XGBOOST with BAGAN although both of these techniques are much more data intensive than FAL.

Dataset	Method	AUC-ROC	F1-Score	Balanced-Accuracy
CreditCard	CE	0.9765	0.7892	0.9286
	CE+SM	0.9714	0.7136	0.9312
	CE+NM	0.9546	0.7293	0.9294
	FL	0.9765	0.8044	0.9286
	FAL	0.9795	0.8268	0.9264
	FAML	**0.9819**	0.8398	**0.9345**
	FL+LGM	0.9794	**0.8508**	0.9079
PaySim	CE	0.9870	0.7702	0.9542
	CE+SM	**0.9983**	0.7973	**0.9828**
	CE+NM	0.7951	0.0072	0.8743
	FL	0.9913	0.7724	0.9625
	FAL	0.9962	0.8011	0.9653
	FAML	0.9979	0.8085	0.9825
	FL+LGM	0.9959	**0.8171**	0.9792 -
CCFraud	CE	0.9574	0.6173	0.8321
	CE+SM	0.9580	0.6192	0.8895
	CE+NM	0.9580	0.6189	0.8875
	FL	**0.9580**	0.6192	0.8612
	FAL	0.9579	0.6191	0.8895
	FAML	0.9578	0.6195	0.8897
	FL+LGM	0.9565	**0.6205**	**0.8899**
GMSC	CE	0.8341	0.4210	0.7255
	CE+SM	0.8293	0.4067	**0.7626**
	CE+NM	0.75481	0.3028	0.6978
	FL	0.8343	0.4198	0.7435
	FAL	**0.8373**	0.4230	0.7608
	FAML	0.8344	**0.4268**	0.7620
	FL+LGM	**0.8385**	0.4253	**0.7645**
SatImage	CE	0.9346	0.6165	0.8517
	CE+SM	0.9454	0.3071	0.7425
	CE+NM	0.9077	0.5741	0.8526
	FL	0.9349	0.6224	0.8647
	FAL	**0.9577**	0.6821	**0.8899**
	FAML	0.9497	0.6620	0.8840
	FL+LGM	0.9467	**0.6827**	0.8910
HomeCredit	CE	0.7725	0.3233	0.7036
	CE+SM	0.7394	0.2883	0.6790
	CE+NM	0.7723	**0.3283**	0.7037
	FL	0.7705	0.3147	0.7032
	FAL	0.7741	0.3216	0.7057
	FAML	**0.7777**	0.3251	**0.7094**
	FL+LGM	0.7760	**0.3283**	0.7075

Table 2: Comparison of our method with Cross-Entropy (CE), Focal Loss (FL) and sampling methods. We have used SMOTE (SM) for oversampling and Near-Miss (NM) for undersampling.

Dataset	Methods	AUC-ROC	Std Error	Bal acc	Std Error	F1	Std Error
CreditCard	CE	0.9776	0.007	0.9275	0.016	0.7851	0.028
	FAL	0.9756	0.007	0.9258	0.0119	0.7936	0.0274
	FAML	**0.9812**	0.009	**0.9321**	0.0163	**0.7945**	0.0225
SatImage	CE	**0.9295**	0.018	0.8532	0.032	0.5295	0.076
	FAL	0.9050	0.09	**0.8724**	0.079	0.5354	0.065
	FAML	0.9126	0.036	0.8625	0.053	**0.5384**	0.068
GMSC	CE	0.8264	0.002	0.7582	0.0156	0.4121	0.0081
	FAL	**0.8285**	0.0022	0.7790	0.0156	0.4214	0.0091
	FAML	0.9049	0.090	**0.8724**	0.079	**0.5353**	0.065
HomeCredit	CE	0.7646	0.0028	0.6953	0.0186	0.3115	0.087
	FAL	0.7704	0.0412	0.7030	0.0658	**0.3165**	0.0296
	FAML	**0.7725**	0.0458	**0.7046**	0.0565	0.3146	0.0305
PaySim	CE	0.9263	0.0032	0.9036	0.0131	0.6388	0.0663
	FAL	0.9756	0.007	0.9257	0.0119	0.7936	0.0274
	FAML	**0.9852**	0.0041	**0.9324**	0.0124	**0.8126**	0.0654

Table 3: 10-fold cross validation results for the 6 fraud datasets.

Imbalance Factor	10	100
Softmax Loss	13.61	29.64
Focal Loss	13.68	30.41
Class Balanced Loss	12.51	25.43
FAL Loss	13.53	27.02

Table 4: Error rate for Long-Tailed CIFAR10.

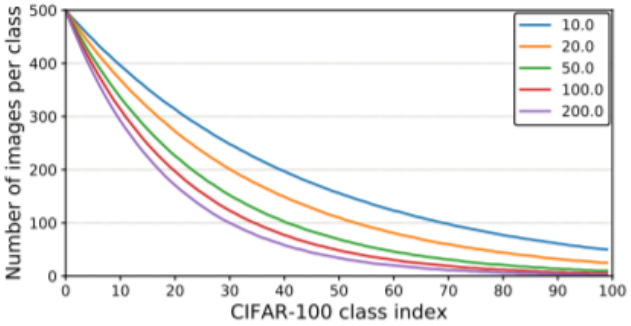


Figure 5: Long-Tailed CIFAR10: number of examples per class with different class imbalance ratio. Image taken from Cui et al. (2019).

Method	AUC-ROC
D-Tree	0.871
D-Tree + Smote	0.881
BAGAN + XG-BOOST	0.950
FAL Loss	**0.977**

Table 5: Results on the proprietary financial data.

5. Conclusions

Imposter neighbours and hard examples are key issues due to which DNNs have difficulty in modeling the datasets with skewed distribution. In our work, we propose an end-to-end solution to these problems by formulating a two stage loss function that combines discriminative prowess of large margin gaussian mixture (LGM) loss and class agnostic cost sensitive learning of focal loss, which we call Focused Anchors Loss (FAL). We evaluate our loss function across range of different settings which include: (i) numeric and image based datasets, (ii) binary and multi-class classification and (iii) Multi-layer Perceptrons and Convolutional Neural Networks. We achieve up-to 17% gain in F1-score and 3% gain in balanced accuracy in our experiments.

References

Alejandro Correa Bahnsen, Djamila Aouada, and Björn E. Ottersten. Example-dependent cost-sensitive decision trees. *Expert Syst. Appl.*, 42(19):6609–6619, 2015.

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004. ISSN 1931-0145.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

Peng Cao, Dazhe Zhao, and Osmar R. Zaïane. An optimized cost-sensitive SVM for imbalanced data learning. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II*, pages 280–292, 2013.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. *CoRR*, abs/1901.05555, 2019.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

Chris Drummond and Robert C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. pages 1–8, 2003.

Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, June 2008.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5375–5384, 2016.

Salman Hameed Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Ahmed Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Netw. Learning Syst.*, 29(8):3573–3587, 2018.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

X. Liu, J. Wu, and Z. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, April 2009. ISSN 1083-4419.

Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and A. Cristiano I. Malossi. BAGAN: data augmentation with balancing GAN. *CoRR*, abs/1803.09655, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. *CoRR*, abs/1803.02988, 2018.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV (7)*, volume 9911 of *Lecture Notes in Computer Science*, pages 499–515. Springer, 2016. ISBN 978-3-319-46477-0.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, page 435, 2003.

J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.