Jensen-Tsallis Divergence for Supervised Classification under Data Imbalance

Antonio Squicciarini[a], Thomas Trigano[b] and David Luengo[a]

a) Universidad Politécnica de Madrid (Spain), b) Shamoon College of Engineering (Israel)

Introduction

Due to its unique properties (such as boundedness, symmetry, and the ability to handle more than two distributions), the Jensen-Shannon Divergence (JSD) has been applied in various DL contexts: supervised classification, adversarial training, domain generalization, etc.

In this study, we analyze the use of JSD and Jensen-Tsallis Divergence (JTD) in supervised classification under data imbalance.

Main Contributions:

- Regularization interpretation of Jensen-Shannon Divergence.
- Extending this interpretation to Jensen-Tsallis Divergence, highlighting its added flexibility.
- Empirical evidence showing JTD enhances generalization in supervised classification.
- Investigation of JSD and JTD effects in imbalanced classification scenarios.

Related works

[Pereyra et al., 207] introduced a **confidence penalization** term at the output, penalizing low-entropy output distributions:

$$\mathcal{L}_p = D_{KL}(e^{(y)}||\tilde{p}) - \gamma\mathbb{H}[\tilde{p}], \tag{1}$$

where $e^{(y)}$ is a one-hot encoded vector, $\tilde{p}$ is the probability vector estimated by the network, γ is a hyperparameter that controls the strength of the confidence penalty, D_{KL} denotes the Kullback-Leibler Divergence, and $\mathbb{H}$ indicates Shannon's entropy.

According to [Mukhoti et al., 2020], Eq. (1) is related with the focal loss [Lin et al., 2017] (widely used to deal with calibration issues) as follows:

$$\mathcal{L}_f = -(1-\tilde{p}_y)^{\gamma}\log\tilde{p}_y \geq \mathcal{L}_p = D_{KL}(e^{(y)}||\tilde{p}) - \gamma\mathbb{H}[\tilde{p}]. \tag{2}$$

Theoretical Background

Assume a general functional class $\mathcal{F}$, where each $f \in \mathcal{F}$ maps an input $\boldsymbol{x} \in \mathbb{X}$ to the probability simplex Δ^{K-1}, i.e., to a categorical distribution over K classes $y \in \mathbb{Y} = \{1, 2, \ldots, K\}$. We seek $f^* \in \mathcal{F}$ that minimizes a risk, $R_{\mathcal{L}}(f) = \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}\left(e^{(y)}, f(\boldsymbol{x})\right)\right]$, for some loss function $\mathcal{L}$ and joint distribution $\mathcal{D}$ over $\mathbb{X}\times\mathbb{Y}$, where $e^{(y)}$ is a K-dimensional vector with one at index y and zero elsewhere. In practice, $\mathcal{D}$ is unknown and, instead, we use $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$, which are assumed to be identically and independently sampled from $\mathcal{D}$, to minimize an empirical risk $\frac{1}{N}\sum_{i=1}^N \mathcal{L}\left(e^{(y_i)}, f(x_i)\right)$.

Jensen-Tsallis Divergence

For two probability distributions, the JSD is defined as follows:

$$JSD^{\pi}(p,\rho) = \pi_1 D_{KL}(p||m) + \pi_2 D_{KL}(\rho||m) = \mathbb{H}[m] - \pi_1\mathbb{H}[p] - \pi_2\mathbb{H}[\rho], \tag{3}$$

where $p \in \Delta^{K-1}$ and $\rho \in \Delta^{K-1}$ are the two discrete probability distributions over K classes, $\pi \in \Delta$ is the weight distribution that assigns different importances to the two distributions, $m = \pi_1 p + \pi_2\rho$ is the weighted average distribution.

In the **typical supervised learning scenario**, where the labels follow a one-hot distribution $e^{(y)}$:

$$JSD^{\pi}(e^{(y)},\tilde{p}) = \mathbb{H}\left[\pi_1 e^{(y)} + \pi_2\tilde{p}\right] - \pi_2\mathbb{H}[\tilde{p}]. \tag{4}$$

The **Jensen-Tsallis Divergence** (JTD) is a generalization of the JSD that introduces the q-logarithm $log^{(q)}$:

$$JTD^{\pi}_q(e^{(y)},\tilde{p}) = \sum_{j=1}^{K}\left(\pi_1 e_j^{(y)} + \pi_2\tilde{p}_j^q\right)\log^{(q)}(m_j) - \pi_2\mathbb{H}_q[\tilde{p}], \tag{5}$$

The JTD can be expressed using the Tsallis divergence, $D_T^{(q)}$, as follows:

$$JTD^{\pi}_q(e^y,\tilde{p}) = \pi_1 D_T^{(q)}(e^y||m) + \pi_2 D_T^{(q)}(\tilde{p}||m)$$
$$= -\pi_1\log^{(q)}(\pi_1 + \pi_2\tilde{p}_y) - \pi_2\left(\sum_{j=1\wedge j\neq y}^{K}\tilde{p}_j\log^{(q)}(\pi_2) + \tilde{p}_y\log^{(q)}\left(\frac{\pi_1}{\tilde{p}_y} + \pi_2\right)\right).$$

Here, the last term plays a regularization role over the confidence output $\tilde{p}_y$ of the network.

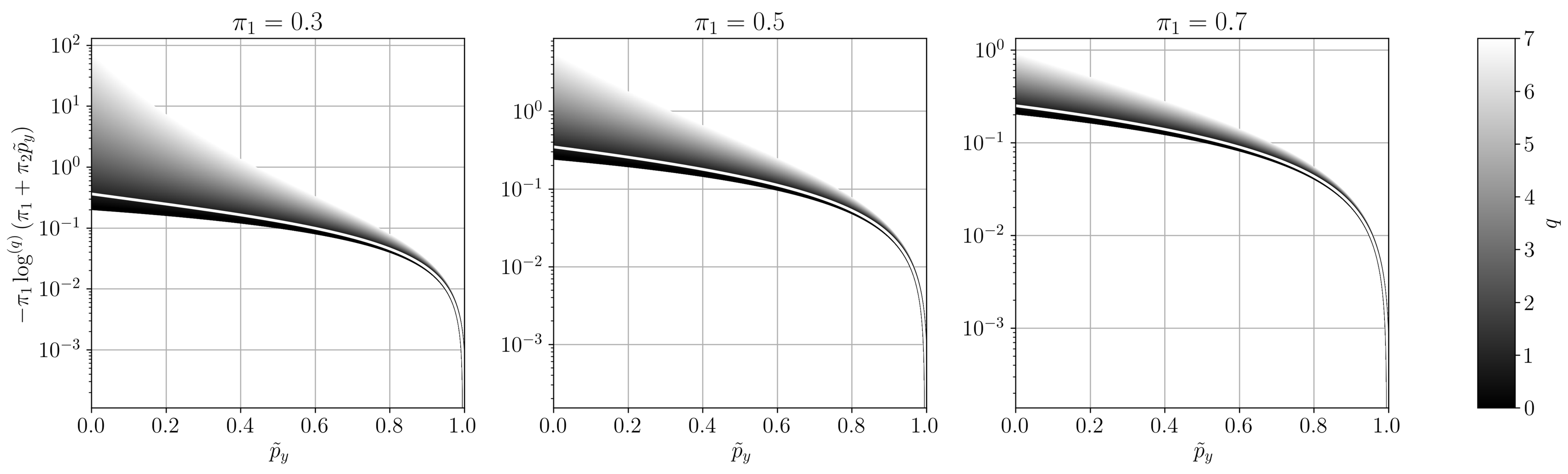


Figure: Numerical representation of the first term of the JTD $-\pi_1\log^{(q)}(\pi_1 + \pi_2\tilde{p}_y)$.

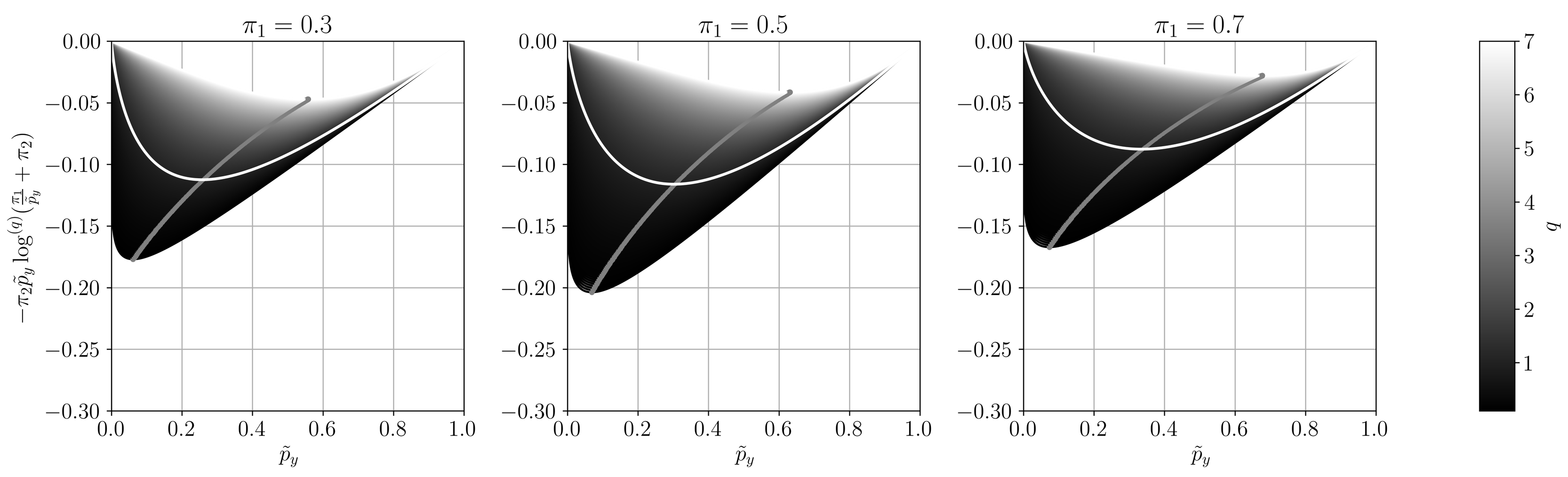


Figure: Numerical representation of regularization component $-\pi_2\tilde{p}_y\log^{(q)}\left(\frac{\pi_1}{\tilde{p}_y} + \pi_2\right)$.

Numerical Simulations

Illustrative Experiment – CIFAR-10: The network trained using the CE achieves 100% accuracy on the training set, whereas the ones trained using the JSD and the JTD functions do not. However, networks trained using the JTD with larger values of q outperform the others on the test set.

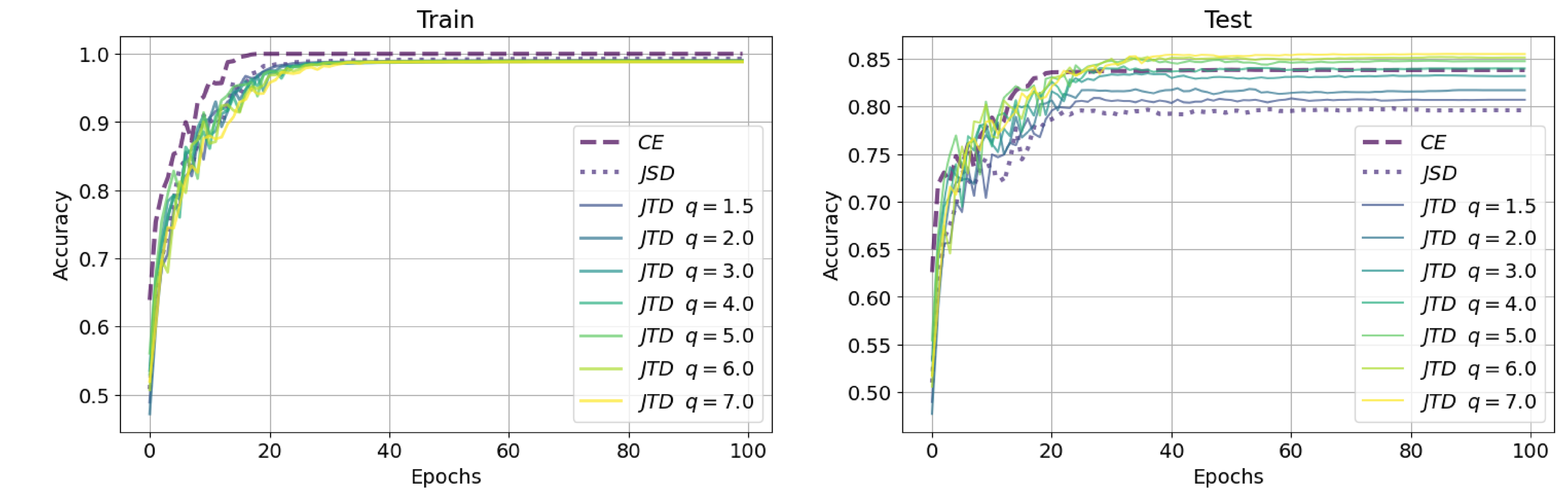
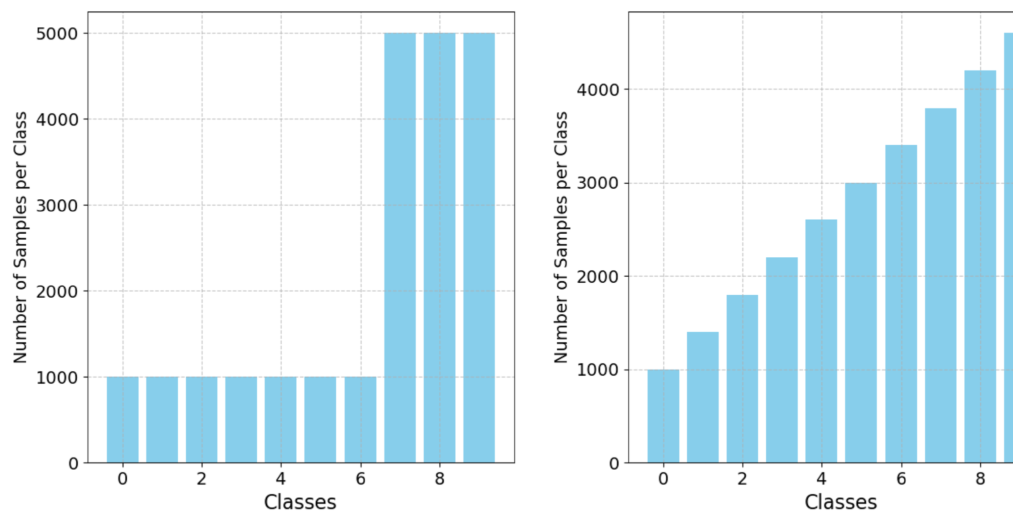


Figure: Learning curves over CIFAR-10. ResNet34, trained for 100 epochs, batch size of 256, starting learning rate of 0.01, and a cosine decay learning rate applied. Stochastic Gradient Descent (SGD) with Nesterov momentum 0.9 and weight decay $1e-4$ was employed. No data augmentation was applied to showcase the regularization effect of the cost function.

Imbalanced Data Experiment: Varying degrees of artificial imbalance to different open datasets based on the framework of [Buda et al., 2018]

$$\rho = \frac{\max_j\{|C_j|\}}{\min_j\{|C_j|\}}$$
$$\mu = \frac{|\{j \in \{1,\ldots,K\}: C_j \text{ is minority}\}|}{K}$$



((a)) Step $\rho = 5$ $\mu = 0.7$ ((b)) Linear $\rho = 5$

Figure: Number of samples in each class after applying the imbalance strategy on CIFAR-10. C_j is the set of samples contained in the dataset belonging to class j, and K is the total number of classes.

Table: Average accuracy and relative standard deviation over 5 runs - CIFAR10

CIFAR10		Imbalance Type					
		Linear (ρ)			Step (ρ - μ)		
Loss		2.0	10.0	50.0	2.0 - 0.5	10.0 - 0.5	50.0 - 0.5
BL	92.43±0.81	90.77±0.27	85.62±0.13	80.48±0.47	89.62±0.44	75.83±0.22	55.01±0.23
CE	**92.90±0.71**	**91.44±0.80**	**86.82±0.29**	**81.66±0.27**	**90.68±0.25**	**77.50±0.20**	55.63±0.12
FL	92.45±0.45	90.74±0.36	85.87±0.16	80.06±0.47	89.94±0.40	75.74±0.22	54.33±0.19
JSD	91.24±0.44	89.21±0.31	84.07±0.23	78.52±0.46	88.54±0.30	74.00±0.19	54.43±0.21
JTD	92.63±0.55	90.93±0.55	**86.71±0.23**	**82.07±0.35**	**90.40±0.27**	**77.81±0.13**	**59.00±0.16**
MAE	84.12±1.98	78.15±7.59	77.14±4.60	72.21±5.90	74.53±6.72	68.86±5.16	55.89±4.55

Table: Average accuracy and relative standard deviation over 10 runs for MNIST - Step Imbalance

MNIST	Imbalance Type								
	Step (ρ - μ)								
Loss	10.0 - 0.5	25.0 - 0.5	50.0 - 0.5	100.0 - 0.5	250.0 - 0.5	500.0 - 0-5	1000.0 - 0.5	2500.0 - 0.5	5000.0 - 0.5
BL	97.44±0.19	95.96±0.44	94.34±0.46	91.79±0.6	85.8±1.21	79.49±1.33	72.7±1.89	60.4±3.56	**54.58±1.84**
CE	97.83±0.13	96.53±0.27	94.88±0.42	92.79±0.57	87.1±1.27	81.18±1.5	73.73±1.76	61.5±2.79	**54.72±1.83**
FL	97.64±0.22	96.17±0.31	94.37±0.46	91.99±0.74	86.28±1.43	79.96±1.47	71.38±2.07	60.06±3.26	**53.69±1.73**
JSD	96.5±0.4	94.71±0.61	92.88±0.61	90.69±0.77	85.46±1.66	81.69±1.6	**74.72±1.8**	**63.63±3.74**	**55.5±2.13**
JTD	**98.34±0.14**	**97.33±0.16**	**96.17±0.35**	**94.05±1.33**	**89.06±1.24**	**83.96±1.18**	74.54±2.42	61.22±3.73	**55.28±2.1**
MAE	86.75±8.13	82.61±7.56	82.55±7.16	77.57±9.51	68.92±11.3	64.66±8.24	56.31±8.46	45.55±6.98	37.52±9.29

Table: Average accuracy and relative standard deviation over 10 runs for MNIST - Linear Imbalance

MNIST	Imbalance Type								
	Linear (ρ)								
Loss	10.0	25.0	50.0	100.0	250.0	500.0	1000.0	2500.0	5000.0
BL	98.29±0.09	98.02±0.11	97.88±0.18	97.5±0.29	96.61±0.66	95.67±0.58	94.29±0.85	91.63±1.54	90.45±1.2
CE	98.52±0.11	98.26±0.12	98.05±0.22	97.74±0.31	96.81±0.37	95.81±0.57	94.57±1.13	92.36±1.48	90.98±1.35
FL	98.46±0.11	98.12±0.13	97.91±0.19	97.54±0.3	96.52±0.48	95.53±0.65	94.21±1.28	91.93±1.4	90.3±1.28
JSD	97.3±0.27	96.99±0.2	96.8±0.24	96.51±0.29	95.88±0.55	95.09±0.62	93.67±1.09	91.79±1.46	89.93±1.56
JTD	**98.66±0.1**	**98.42±0.13**	**98.29±0.13**	**98.18±0.34**	**97.45±0.47**	**96.26±0.68**	**95.31±1.3**	**93.22±1.42**	**91.56±1.32**
MAE	88.59±7.71	89.45±6.88	88.25±7.67	84.49±8.93	86.13±7.99	83.28±9.63	83.42±8.39	78.94±10.86	80.16±8.12

Table: Sensitivity analysis of JTD generalization with respect π and q over CIFAR10 with ResNet34

π_1	q							
	0.5	1.0	2.0	3.0	4.0	5.0	6.0	7.0
0.1	87.36	88.43	90.99	91.63	87.35	57.76	10	10
0.2	89.85	90.44	91.38	91.79	91.97	88.24	10	10
0.3	90.17	90.68	91.69	92.19	92.52	82.50	89.72	83.74
0.4	90.39	91.10	92.15	92.64	92.64	92.85	92.24	92.79
0.5	90.85	91.18	91.90	92.27	92.47	92.69	92.47	92.69
0.6	90.55	91.53	91.83	91.99	91.80	90.79	84.80	58.25
0.7	90.57	91.08	91.75	90.93	87.89	43.77	24.35	19.02
0.8	90.12	90.53	91.34	87.01	41.80	29.09	10	18.25
0.9	88.63	90.32	90.61	59.28	29.52	18.24	10	10

Conclusions

- JSD and JTD can be interpreted as loss functions with intrinsic confidence regularization in supervised learning applications, and also as a priors over the output confidence of the network.
- JTD can outperform the JSD, CE, and other loss functions by tuning the parameter q.
- Analysis of imbalanced data classification scenarios conducted using different datasets, highlighting that the JTD emerges as one of the best loss functions for generalization in this case.

REFERENCES

M. Buda et al., "A systematic study of the class imbalance problem in convolutional neural networks", *Neural Networks* 106, 249–259, 2018.

T.-Y. Lin et al., "Focal loss for dense object detection", *ICCV* 2017.

J. Mukhoti et al., "Calibrating deep neural networks Using focal loss", *NeurIPS* 2020.

G. Pereyra et al., "Penalizing Confident Output Distributions", *ICLR* 2017.