Robust Bi-Tempered Logistic Loss Based on Bregman Divergences

Ehsan Amid [⋆†] **Manfred K. Warmuth** [⋆†] **Rohan Anil** [†] **Tomer Koren** [†]

⋆ Department of Computer Science, University of California, Santa Cruz
† Google Brain
{eamid, manfred, rohananil, tkoren}@google.com

Abstract

We introduce a temperature into the exponential function and replace the softmax output layer of neural nets by a high temperature generalization. Similarly, the logarithm in the log loss we use for training is replaced by a low temperature logarithm. By tuning the two temperatures we create loss functions that are non-convex already in the single layer case. When replacing the last layer of the neural nets by our bi-temperature generalization of logistic loss, the training becomes more robust to noise. We visualize the effect of tuning the two temperatures in a simple setting and show the efficacy of our method on large data sets. Our methodology is based on Bregman divergences and is superior to a related two-temperature method using the Tsallis divergence.

1 Introduction

The logistic loss, also known as the softmax loss, has been the standard choice in training deep neural networks for classification. The loss involves the application of the softmax function on the activations of the last layer to form the class probabilities followed by the relative entropy (aka the Kullback-Leibler (KL) divergence) between the true labels and the predicted probabilities. The logistic loss is known to be a convex function of the activations (and consequently, the weights) of the last layer.

Although desirable from an optimization standpoint, convex losses have been shown to be prone to outliers [14] as the loss of each individual example unboundedly increases as a function of the activations. These outliers may correspond to extreme examples that lead to large gradients, or misclassified training examples that are located far away from the classification boundary. Requiring a convex loss function at the output layer thus seems somewhat arbitrary, in particular since convexity in the last layer's activations does not guarantee convexity with respect to the parameters of the network outside the last layer. Another issue arises due to the exponentially decaying tail of the softmax function that assigns probabilities to the classes. In the presence of mislabeled training examples near the classification boundary, the short tail of the softmax probabilities enforces the classifier to closely follow the noisy training examples. In contrast, heavy-tailed alternatives for the softmax probabilities have been shown to significantly improve the robustness of the loss to these examples [7].

The logistic loss is essentially the logarithm of the predicted class probabilities, which are computed as the normalized exponentials of the inputs. In this paper, we tackle both shortcomings of the logistic loss, pertaining to its convexity as well as its tail-lightness, by replacing the logarithm and exponential functions with their corresponding "tempered" versions. We define the function $\log_t : \mathbb{R}_+ \to \mathbb{R}$ with *temperature* parameter $t \geqslant 0$ as in [15]:

$$\log_t(x) \coloneqq \frac{1}{1-t}(x^{1-t} - 1)\,. \tag{1}$$

Preprint.

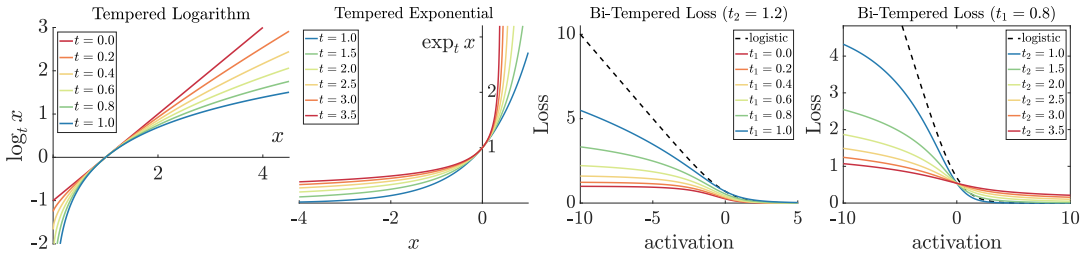

Figure 1: Tempered logarithm and exponential functions, and the bi-tempered logistic loss: (a) $\log_t$ function, (b) $\exp_t$ function, bi-tempered logistic loss when (c) $t_2 = 1.2$ fixed and $t_1 \leqslant 1$, and (d) $t_1 = 0.8$ fixed and $t_2 \geqslant 1$.

The $\log_t$ function is monotonically increasing and concave. The standard (natural) logarithm is recovered at the limit $t \to 1$. Unlike the standard log, the $\log_t$ function is bounded from below by $-1/(1-t)$ for $0 \leqslant t < 1$. This property will be used to define bounded loss functions that are significantly more robust to outliers. Similarly, our heavy-tailed alternative for the softmax function is based on the tempered exponential function. The function $\exp_t : \mathbb{R} \to \mathbb{R}_+$ with temperature $t \in \mathbb{R}$ is defined as the inverse[1] of $\log_t$, that is,

$$\exp_t(x) \coloneqq [1 + (1-t)\,x]_+^{1/(1-t)} \,, \tag{2}$$

where $[\,\cdot\,]_+ = \max\{\,\cdot\,, 0\}$. The standard exp function is again recovered at the limit $t \to 1$. Compared to the exp function, a heavier tail (for negative values of x) is achieved for $t > 1$. We use this property to define heavy-tailed analogues of softmax probabilities at the output layer.

The vanilla logistic loss can be viewed as a logarithmic (relative entropy) divergence that operates on a "matching" exponential (softmax) probability assignment [10, 11]. Its convexity then stems from classical convex duality, using the fact that the probability assignment function is the gradient of the dual function to the entropy on the simplex. When the $\log_{t_1}$ and $\exp_{t_2}$ are substituted instead, this duality still holds whenever $t_1 = t_2$, albeit with a different Bregman divergence, and the induced loss remains convex[2]. However, for $t_1 < t_2$, the loss becomes non-convex in the output activations. In particular, $0 \leqslant t_1 < 1$ leads to a bounded loss, while $t_2 > 1$ provides tail-heaviness. Figure 1 illustrates the tempered $\log_t$ and $\exp_t$ functions as well as examples of our proposed bi-tempered logistic loss function for a 2-class problem expressed as a function of the activation of the first class. The true label is assumed to be class one.

Tempered generalizations of the logistic regression have been introduced before [6, 7, 21, 2]. The most recent two-temperature method [2] is based on the Tsallis divergence and contains all the previous methods as special cases. However, the Tsallis based divergences do not result in proper loss functions. In contrast, we show that the Bregman based construction introduced in this paper is indeed proper, which is a requirement for many real-world applications.

1.1 Our replacement of the softmax output layer in neural nets

Consider an arbitrary classification model with multiclass softmax output. We are given training examples of the form $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is a fixed dimensional input vector and the target $\boldsymbol{y}$ is a probability vector over k classes. In practice, the targets are often one-hot encoded binary vectors in k dimensions. Each input $\boldsymbol{x}$ is fed to the model, resulting in a vector $\boldsymbol{z}$ of inputs to the output softmax. The softmax layer has typically one trainable weight vector $\boldsymbol{w}_i$ per class i and yields the predicted class probability

$$\hat{y}_i = \frac{\exp(\hat{a}_i)}{\sum_{j=1}^{k} \exp(\hat{a}_j)} = \exp\left(\hat{a}_i - \log \sum_{j=1}^{k} \exp(\hat{a}_j)\right), \text{ for linear activation } \hat{a}_i = \boldsymbol{w}_i \cdot \boldsymbol{z} \text{ for class } i.$$

We first replace the softmax function by a generalized heavy-tailed version that uses the $\exp_{t_2}$ function with $t_2 > 1$, which we call the *tempered softmax function*:

$$\hat{y}_i = \exp_{t_2}\left(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}})\right), \quad \text{where} \quad \lambda_{t_2}(\hat{\boldsymbol{a}}) \in \mathbb{R} \quad \text{is s.t.} \quad \sum_{j=1}^{k} \exp_{t_2}\left(\hat{a}_j - \lambda_{t_2}(\hat{\boldsymbol{a}})\right) = 1 \,.$$

[1] When $0 \leqslant t < 1$, the domain of $\exp_t$ needs to be restricted to $-1/(1-t) \leqslant x$ for the inverse property to hold.
[2] In a restricted domain when $t_1 = t_2 < 1$, as discussed later.

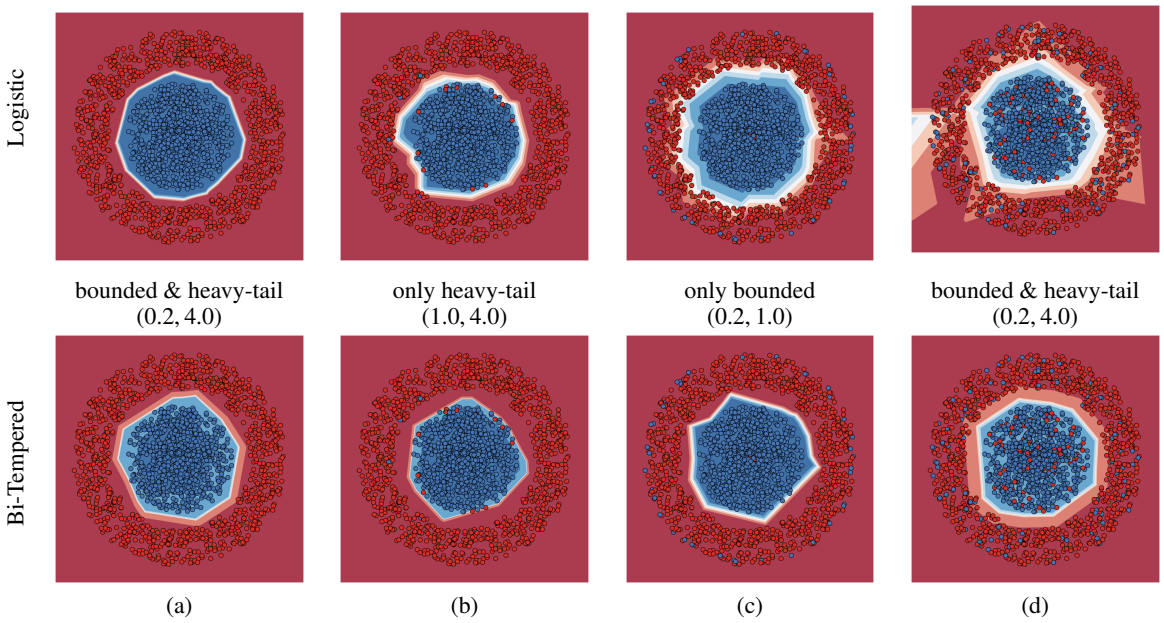

Figure 2: Logistic vs. robust bi-tempered logistic loss: (a) noise-free labels, (b) small-margin label noise, (c) large-margin label noise, and (d) random label noise. The temperature values (t_1, t_2) for the bi-tempered loss are shown above each figure.

This requires computing the normalization value $\lambda_{t_2}(\hat{\boldsymbol{a}})$ (for each example) via a binary search or an iterative procedure like the one given in Appendix A. The relative entropy between the true label $\boldsymbol{y}$ and prediction $\hat{\boldsymbol{y}}$ is replaced by the tempered version with temperature $0 \leqslant t_1 < 1$,

$$\sum_{i=1}^{k} \left(y_i \left(\log_{t_1} y_i - \log_{t_1} \hat{y}_i \right) - \frac{1}{2-t_1} \left(y_i^{2-t_1} - \hat{y}_i^{2-t_1} \right) \right) \overset{\text{if } \boldsymbol{y} \text{ one-hot}}{=} -\log_{t_1} \hat{y}_c - \frac{1}{2-t_1} \left(1 - \sum_{i=1}^{k} \hat{y}_i^{2-t_1} \right).$$

where $c = \operatorname{argmax}_i y_i$ is the index of the one-hot class. We motivate this loss in later sections. When $t_1 = t_2 = 1$, then it reduces to the vanilla logistic loss for the softmax. On the other hand, when $0 \leqslant t_1 < 1$, then the loss is bounded, while $t_2 > 1$ gives the tempered softmax function a heavier tail.

1.2 An illustration

We provide some intuition on why both boundedness of the loss as well as tail-heaviness of the tempered softmax are crucial for robustness. For this, we train a small two layer feed-forward neural network on a synthetic binary classification problem in two dimensions. The network has 10 and 5 units in the first and second layer, respectively. Figure 2(a) shows the results of the logistic and our bi-tempered logistic loss on the noise-free dataset. The network converges to a desirable classification boundary (the white stripe in the figure) using both loss functions. In Figure 2(b), we illustrate the effect of adding small-margin label noise to the training examples, targeting those examples that reside near the noise-free classification boundary. The logistic loss clearly follows the noisy examples by stretching the classification boundary. On the other hand, using *only* the tail-heavy tempered softmax function ($t_2 = 4$ while $t_1 = 1$, i.e. KL divergence as the divergence) can handle the noisy examples by producing more uniform class probabilities. Next, we show the effect of large-margin noisy examples in Figure 2(c), targeting examples that are located far away from the noise-free classification boundary. The convexity of the logistic loss causes the network to be highly affected by the noisy examples that are located far away from the boundary. In contrast, *only* the boundedness of the loss ($t_1 = 0.2$ while $t_2 = 1$, meaning that the outputs are vanilla softmax probabilities) reduces the effect of the outliers by allocating at most a finite amount of loss to each example. Finally, we show the effect of random label noise that includes both small-margin and large-margin noisy examples in Figure 2(d). Clearly, the logistic loss fails to handle the noise, while our bi-tempered logistic loss successfully recovers the appropriate boundary. Note that to handle the random noise, we exploit *both* boundedness of the loss ($t_1 = 0.2 < 1$) as well as the tail-heaviness of the probability assignments ($t_2 = 4 > 1$).

The theoretical background as well as our treatment of the softmax layer of the neural networks are developed in later sections. In particular, we show that special discrete choices of the temperatures

result in a large variety of divergences commonly used in machine learning. As we show in our experiments, tuning the two temperatures as continuous parameters is crucial.

1.3 Summary of the experiments

We perform experiments by adding synthetic label noise to MNIST and CIFAR-100 datasets and compare the results of our robust bi-tempered loss to the vanilla logistic loss. Our bi-tempered loss is significantly more robust to label noise; it provides 98.56% and 62.55% accuracy on MNIST and CIFAR-100, respectively, when trained with 40% label noise (compared to 97.64% and 53.17%, respectively, obtained using logistic loss). The bi-tempered loss also yields improvement over the state-of-the-art results on the ImageNet-2012 dataset using both the Resnet18 and Resnet50 architectures (see Table 2).

2 Preliminaries

2.1 Convex duality and Bregman divergences on the simplex

We start by briefly reviewing some basic background in convex analysis. For a continuously-differentiable strictly convex function $F : \mathcal{D} \to \mathbb{R}$, with convex domain $\mathcal{D}$, the Bregman divergence between $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \mathcal{D}$ induced by F is defined as

$$\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = F(\boldsymbol{y}) - F(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot f(\hat{\boldsymbol{y}}) \,,$$

where $f(\hat{\boldsymbol{y}}) \coloneqq \nabla F(\hat{\boldsymbol{y}})$ denotes the gradient of F at $\hat{\boldsymbol{y}}$ (sometimes called the link function of F). Clearly $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) \geqslant 0$ and $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 0$ iff $\boldsymbol{y} = \hat{\boldsymbol{y}}$. Also the Bregman divergence is always convex in the first argument and $\nabla_y \, \Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = f(\boldsymbol{y}) - f(\hat{\boldsymbol{y}})$, but not generally in its second argument.

Bregman divergence generalizes many well-known divergences such as the squared Euclidean $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{2} \left\| \boldsymbol{y} - \hat{\boldsymbol{y}} \right\|_2^2$ (with $F(\boldsymbol{y}) = \frac{1}{2} \left\| \boldsymbol{y} \right\|_2^2$) and the Kullback-Leilbler (KL) divergence $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_i (y_i \log \frac{y_i}{\hat{y}_i} - y_i + \hat{y}_i)$ (with $F(\boldsymbol{y}) = \sum_i (y_i \log y_i - y_i)$). Note that the Bregman divergence is not symmetric in general, i.e., $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) \neq \Delta_F(\hat{\boldsymbol{y}}, \boldsymbol{y})$. Additionally, the Bregman divergence is invariant to adding affine functions to the convex function F: $\Delta_{F+A}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}})$, where $A(\boldsymbol{y}) = b + \boldsymbol{c} \cdot \boldsymbol{y}$ for arbitrary $b \in \mathbb{R}$, $\boldsymbol{c} \in \mathbb{R}^k$.

For every differentiable strictly convex function F (with domain $\mathcal{D} \subseteq \mathbb{R}_+^k$), there exists a convex dual $F^* : \mathcal{D}^* \to \mathbb{R}$ function such that for dual parameter pairs $(\boldsymbol{y}, \boldsymbol{a})$, $\boldsymbol{a} \in \mathcal{D}^*$, the following holds: $\boldsymbol{a} = f(\boldsymbol{y})$ and $\boldsymbol{y} = f^*(\boldsymbol{a}) = \nabla F^*(\boldsymbol{a}) = f^{-1}(\boldsymbol{a})$. However, we are mainly interested in the dual of the function F when the domain is restricted to the probability simplex $S^k \coloneqq \{\boldsymbol{y} \in \mathbb{R}_+^k | \sum_{i=1}^k y_i = 1\}$. Let $\check{F}^* : \check{\mathcal{D}}^* \to \mathbb{R}$ denote the convex conjugate of the restricted function $F : \mathcal{D} \cap S^k \to \mathbb{R}$,

$$\check{F}^*(\boldsymbol{a}) = \sup_{\boldsymbol{y}' \in \mathcal{D} \cap S^k} \left(\boldsymbol{y}' \cdot \boldsymbol{a} - F(\boldsymbol{y}') \right) = \sup_{\boldsymbol{y}' \in \mathcal{D}} \inf_{\lambda \in \mathbb{R}} \left(\boldsymbol{y}' \cdot \boldsymbol{a} - F(\boldsymbol{y}') + \lambda \, (1 - \sum_{i=1}^k y_i') \right) ,$$

where we introduced a Lagrange multiplier $\lambda \in \mathbb{R}$ to enforce the constraint $\boldsymbol{y}' \in S^k$. At the optimum, the following relationships hold between the primal and dual variables:

$$f(\boldsymbol{y}) = \boldsymbol{a} - \lambda(\boldsymbol{a}) \, \mathbf{1} \quad \text{and} \quad \boldsymbol{y} = f^{-1} \left(\boldsymbol{a} - \lambda(\boldsymbol{a}) \, \mathbf{1} \right) = \check{f}^*(\boldsymbol{a}) \,, \tag{3}$$

where $\lambda(\boldsymbol{a})$ is chosen so that $\boldsymbol{y} \in S^k$. Note the dependence of the optimum λ on $\boldsymbol{a}$.

2.2 Matching losses

Next, we recall the notion of a *matching loss* [10, 11, 3, 16]. It arises as a natural way of defining a loss function over activations $\hat{\boldsymbol{a}} \in \mathbb{R}^k$, by first mapping them to a probability distribution using a *transfer function* $s : \mathbb{R}^k \to S^k$ that assigns probabilities to classes, and then computing a *divergence* Δ_F between this distribution and the correct target labels. The idea behind the following definition is to match the transfer function and the divergence via duality.

Definition 1 (Matching Loss)**.** Let $F : S^k \to \mathbb{R}$ a continuously-differentiable, strictly convex function and let $s : \mathbb{R}^k \to S^k$ be a transfer function such that $\hat{\boldsymbol{y}} = s(\hat{\boldsymbol{a}})$ denotes the predicted probability distribution based on the activations $\hat{\boldsymbol{a}}$. Then the loss function

$$L_F(\hat{\boldsymbol{a}} \mid \boldsymbol{y}) \coloneqq \Delta_F(\boldsymbol{y}, s(\hat{\boldsymbol{a}})) \,,$$

is called the *matching loss* for s, if $s = \check{f}^* = \nabla \check{F}^*$.

This matching is useful due to the following property.

Proposition 1. *The matching loss $L_F(\hat{\boldsymbol{a}} \mid \boldsymbol{y})$ is convex w.r.t. the activations $\hat{\boldsymbol{a}} \in \mathbf{range}((\check{f}^*)^{-1})$.*

Proof. Note that $\check{F}^*$ is a strictly convex function and the following relation holds between the divergences induced by F and $\check{F}^*$:

$$\Delta_F\big(\boldsymbol{y}, \hat{\boldsymbol{y}}\big) = \Delta_{\check{F}^*}\big((\check{f}^*)^{-1}(\hat{\boldsymbol{y}}),\, (\check{f}^*)^{-1}(\boldsymbol{y})\big). \tag{4}$$

Thus for any $\hat{\boldsymbol{a}}$ in the range of $(\check{f}^*)^{-1}$,

$$\Delta_F\big(\boldsymbol{y}, \check{f}^*(\hat{\boldsymbol{a}})\big) = \Delta_{\check{F}^*}\big(\hat{\boldsymbol{a}}, (\check{f}^*)^{-1}(\boldsymbol{y})\big).$$

The claim now follows from the convexity of the Bregman divergence $\Delta_{\check{F}^*}$ w.r.t. its first argument. □

The original motivating example for the matching loss was the logistic loss [10, 11]. It can be obtained as the matching loss for the softmax function

$$\hat{y}_i = [\check{f}^*(\hat{\boldsymbol{a}})]_i = \frac{\exp(\hat{a}_i)}{\sum_{j=1}^k \exp(\hat{a}_j)},$$

which corresponds to the relative entropy (KL) divergence

$$L_F(\hat{\boldsymbol{a}} \mid \boldsymbol{y}) = \Delta_F\big(\boldsymbol{y}, \check{f}^*(\hat{\boldsymbol{a}})\big) = \sum_{i=1}^k y_i\,(\log y_i - \log \hat{y}_i) = \sum_{i=1}^k \big(y_i\, \log y_i - y_i\, \hat{a}_i)\big) + \log\Big(\sum_{i=1}^k \exp(\hat{a}_i)\Big)\,,$$

induced from the negative entropy function $F(\boldsymbol{y}) = \sum_{i=1}^k (y_i \log y_i - y_i)$. We next define a family of convex functions F_t parameterized by a temperature $t \geqslant 0$. The matching loss $L_{F_t}(\hat{\boldsymbol{a}} \mid \boldsymbol{y}) = \Delta_{F_t}\big(\boldsymbol{y}, \check{f}_t^*(\hat{\boldsymbol{a}})\big)$ for the link function $\check{f}_t^*$ of $\check{F}_t^*$ is convex in the activations $\hat{\boldsymbol{a}}$. However, by letting the temperature t_2 of $\check{f}_{t_2}^*$ be larger than the temperature t_1 of F_{t_1}, we construct bounded non-convex losses with heavy-tailed transfer functions.

3 Tempered Matching Loss

We start by introducing a generalization of the relative entropy, denoted by Δ_{F_t}, induced by a strictly convex function $F_t : \mathbb{R}_+^k \to \mathbb{R}$ with a temperature parameter $t \geqslant 0$. The convex function F_t is chosen so that its gradient takes the form[3] $f_t(\boldsymbol{y}) \coloneqq \nabla F_t(\boldsymbol{y}) = \log_t \boldsymbol{y}$. Via simple integration, we obtain that

$$F_t(\boldsymbol{y}) = \sum_{i=1}^k \big(y_i \log_t y_i + \tfrac{1}{2-t}(1 - y_i^{2-t})\big)\,.$$

Indeed, F_t is a convex function since $\nabla^2 F_t(\boldsymbol{y}) = \mathrm{diag}(\boldsymbol{y}^{-t}) \geq 0$ for any $\boldsymbol{y} \in \mathbb{R}_+^k$. In fact, F_t is strongly convex, for $0 \leqslant t \leqslant 1$:

Lemma 1. *The function F_t, with $0 \leqslant t \leqslant 1$, is B^{-t}–strongly convex over the set $\{\boldsymbol{y} \in \mathbb{R}_+^k : \|\boldsymbol{y}\|_{2-t} \leqslant B\}$ w.r.t. the L_{2-t}-norm.*

See Appendix B for a proof. The Bregman divergence induced by F_t is then given by

$$\begin{aligned}\Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= \sum_{i=1}^k \big(y_i \log_t y_i - y_i \log_t \hat{y}_i - \tfrac{1}{2-t} y_i^{2-t} + \tfrac{1}{2-t}\hat{y}_i^{2-t}\big) \\ &= \sum_{i=1}^k \Big(\tfrac{1}{(1-t)(2-t)}\, y_i^{2-t} - \tfrac{1}{1-t}\, y_i \hat{y}_i^{1-t} + \tfrac{1}{2-t}\, \hat{y}_i^{2-t}\Big).\end{aligned} \tag{5}$$

The second form may be recognized as β-divergence [4] with parameter $\beta = 2 - t$. The divergence (5) includes many well-known divergences such as squared Euclidean, KL, and Itakura-Saito divergence as special cases. A list of additional special cases is given in Table 3 of Appendix C.

The following corollary is the direct consequence of the strong convexity of F_t, for $0 \leqslant t < 1$.

[3] Here, the $\log_t$ function is applied elementwise.

Corollary 1. *Let* $\max(\|\boldsymbol{y}\|_{2-t},\ \|\hat{\boldsymbol{y}}\|_{2-t}) \leqslant B$ *for* $0 \leqslant t < 1$. *Then*

$$\frac{1}{2B^t}\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_{2-t}^2 \leqslant \Delta_{F_t}(\boldsymbol{y},\hat{\boldsymbol{y}}) \leqslant \frac{B^t}{2\,(1-t)^2}\|\boldsymbol{y}^{1-t}-\hat{\boldsymbol{y}}^{1-t}\|_{\frac{2-t}{1-t}}^2 .$$

See Appendix B for a proof. Thus for $0 \leqslant t < 1$, $\Delta_{F_t}(\boldsymbol{y},\hat{\boldsymbol{y}})$ is upper-bounded by $\frac{2B^{2-t}}{(1-t)^2}$. Note that boundedness on the simplex also implies boundedness in the L_{2-t}-ball of radius 1. Thus, Corollary 1 immediately implies the boundedness of the divergence $\Delta_{F_t}(\boldsymbol{y},\hat{\boldsymbol{y}})$ with $0 \leqslant t < 1$ over the simplex. Alternate parameterizations of the family $\{F_t\}$ of convex functions and their corresponding Bregman divergences are discussed in Appendix C.

3.1 Tempered softmax function

Now, let us consider the convex function $F_t(\boldsymbol{y})$ when its domain is restricted to the probability simplex S^k. We denote the constrained dual of $F_t(\boldsymbol{y})$ by $\check{F}_t^*(\boldsymbol{a})$,

$$\check{F}_t^*(\boldsymbol{a}) = \sup_{\boldsymbol{y}'\in S^k}\left(\boldsymbol{y}'\cdot\boldsymbol{a} - F_t(\boldsymbol{y}')\right) = \sup_{\boldsymbol{y}'\in\mathbb{R}_+^k}\ \inf_{\lambda_t\in\mathbb{R}}\left(\boldsymbol{y}'\cdot\boldsymbol{a} - F_t(\boldsymbol{y}') + \lambda_t\left(1-\sum_{i=1}^k y_i'\right)\right). \tag{6}$$

Following our discussion in Section 2.1 and using (3), the transfer function induced by $\check{F}_t^*$ is[4]

$$\boldsymbol{y} = \exp_t\left(\boldsymbol{a}-\lambda_t(\boldsymbol{a})\,\mathbf{1}\right), \quad \text{with } \lambda_t(\boldsymbol{a}) \text{ s.t. } \sum_{i=1}^k \exp_t\left(a_i-\lambda_t(\boldsymbol{a})\right) = 1. \tag{7}$$

3.2 Matching loss of tempered softmax

Finally, we derive the matching loss function L_{F_t}. Plugging in (7) into (5), we have

$$L_t(\hat{\boldsymbol{a}}\mid\boldsymbol{y}) = \Delta_{F_t}\left(\boldsymbol{y},\exp_t(\hat{\boldsymbol{a}}-\lambda_t(\hat{\boldsymbol{a}}))\right).$$

Recall that by Proposition 1, this loss is convex in activations $\hat{\boldsymbol{a}} \in \mathbf{range}((\check{f}_t^*)^{-1})$. In general, $\lambda_t(\boldsymbol{a})$ does not have a closed form solution. However, it can be easily approximated via an iterative method, e.g., a binary search. An alternative (fixed-point) algorithm for computing $\lambda_t(\boldsymbol{a})$ for $t > 1$ is given in Algorithm 1 of Appendix A.

4 Robust Bi-Tempered Logistic Loss

A more interesting class of loss functions can be obtained by introducing a "mismatch" between the temperature of the divergence function (5) and the temperature of the probability assignment function, i.e. the tempered softmax (7). That is, we consider loss functions of the following type:

$$\forall\, 0\leqslant t_1<1<t_2\colon\ L_{t_1}^{t_2}(\hat{\boldsymbol{a}}\mid\boldsymbol{y}) := \Delta_{F_{t_1}}\left(\boldsymbol{y},\exp_{t_2}(\hat{\boldsymbol{a}}-\lambda_{t_2}(\hat{\boldsymbol{a}}))\right),\ \text{ with } \lambda_t(\hat{\boldsymbol{a}}) \text{ s.t. } \sum_{i=1}^k \exp_t\left(a_i-\lambda_t(\boldsymbol{a})\right) = 1. \tag{8}$$

We call this the *Bi-Tempered Logistic Loss*. Note that for the prescribed range of the two temperatures, the loss is bounded and has a heavier-tailed probability assignment function compared to the vanilla softmax function. As illustrated in our 2-dimensional example in Section 1, both properties are crucial for handling noisy examples. The derivative of the bi-tempered loss is given in Appendix E. In the following, we discuss the properties of this loss for classification.

4.1 Properness and Monte-Carlo sampling

Let $P_{\text{UK}}(\boldsymbol{x},y)$ denote the (unknown) joint probability distribution of the observed variable $\boldsymbol{x}\in\mathbb{R}^m$ and the class label $y\in[k]$. The goal of discriminative learning is to approximate the posterior distribution of the labels $P_{\text{UK}}(y\mid\boldsymbol{x})$ via a parametric model $P(y\mid\boldsymbol{x};\Theta)$ parameterized by Θ. Thus the model fitting can be expressed as minimizing the following expected loss between the data and the model's label probabilities

$$\mathbb{E}_{P_{\text{UK}}(\boldsymbol{x})}\left[\Delta\left(P_{\text{UK}}(y\mid\boldsymbol{x}),P(y\mid\boldsymbol{x};\Theta)\right)\right], \tag{9}$$

[4]Note that due to the simplex constraint, the link function $\boldsymbol{y} = \check{f}_t^*(\boldsymbol{a}) = \nabla\check{F}_t^*(\boldsymbol{a}) = \exp_t\left(\boldsymbol{a}-\lambda_t(\boldsymbol{a})\right)$ is different from $f_t^{-1}(\boldsymbol{a}) = f_t^*(\boldsymbol{a}) = \nabla F_t^*(\boldsymbol{a}) = \exp_t(\boldsymbol{a})$, i.e., the gradient of the unconstrained dual.

where $\Delta\big(P_{\text{UK}}(y \mid \boldsymbol{x}), P(y \mid \boldsymbol{x}; \Theta)\big)$ is any proper divergence measure between $P_{\text{UK}}(y \mid \boldsymbol{x})$ and $P(y \mid \boldsymbol{x}; \Theta)$. We use $\Delta := \Delta_{F_{t_1}}$ as the divergence and $P(y = i \mid \boldsymbol{x}; \Theta) := P(i \mid \boldsymbol{x}; \Theta) = \exp_{t_2}(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}}))$, where $\hat{\boldsymbol{a}}$ is the activation vector of the last layer given input $\boldsymbol{x}$ and Θ is the set of all weights of the network. Ignoring the constant terms w.r.t. Θ, our loss (9) becomes

$$\mathbb{E}_{P_{\text{UK}}(\boldsymbol{x})}\Big[\sum_i \big(-P_{\text{UK}}(i \mid \boldsymbol{x}) \log_t P(i \mid \boldsymbol{x}; \Theta) + \frac{1}{2-t} P(i \mid \boldsymbol{x}; \Theta)^{2-t}\big)\Big] \tag{10a}$$

$$= -\mathbb{E}_{P_{\text{UK}}(\boldsymbol{x}, y)}\Big[\log_t P(y \mid \boldsymbol{x}; \Theta)\Big] + \mathbb{E}_{P_{\text{UK}}(\boldsymbol{x})}\Big[\frac{1}{2-t} \sum_i P(i \mid \boldsymbol{x}; \Theta)^{2-t}\big)\Big] \tag{10b}$$

$$\approx \frac{1}{N} \sum_n \big(-\log_t P(y_n \mid \boldsymbol{x}_n; \Theta) + \frac{1}{2-t} \sum_i P(i \mid \boldsymbol{x}_n; \Theta)^{2-t}\big) \tag{10c}$$

where from (10b) to (10c), we perform a Monte-Carlo approximation of the expectation w.r.t. $P_{\text{UK}}(\boldsymbol{x}, y)$ using samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. Thus, (10c) is an unbiased approximate of the expected loss (9), thus is a *proper* loss [19].

Following the same approximation steps for the Tsallis divergence, we have

$$\mathbb{E}_{P_{\text{UK}}(\boldsymbol{x})}\Big[\underbrace{-\sum_i P_{\text{UK}}(i \mid \boldsymbol{x}) \log_t \frac{P(i \mid \boldsymbol{x}; \Theta)}{P_{\text{UK}}(i \mid \boldsymbol{x})}}_{\Delta_t^{\text{Tsallis}}\left(P_{\text{UK}}(y|\boldsymbol{x}),\, P(y|\boldsymbol{x};\Theta)\right)}\Big] \approx -\frac{1}{N} \sum_n \log_t \frac{P(y_n \mid \boldsymbol{x}_n; \Theta)}{P_{\text{UK}}(y_n \mid \boldsymbol{x}_n)},$$

which, due to the fact that $\log_t \frac{a}{b} \neq \log_t a - \log_t b$ in general, requires access to the (unknown) conditional distribution $P_{\text{UK}}(y \mid \boldsymbol{x})$. Thus, the approximation $-\frac{1}{N} \sum_n \log_t P(y_n \mid \boldsymbol{x}_n; \Theta)$ proposed in [2] by approximating $P_{\text{UK}}(y_n \mid \boldsymbol{x}_n)$ by 1 is not an unbiased estimator of (9) and therefore, not proper.

4.2 Bayes-risk consistency

Another important property of a multiclass loss is the Bayes-risk consistency [18]. Bayes-risk consistency of the two-temperature logistic loss based on the Tsallis divergence was shown in [2]. As expected, the tempered Bregman loss (8) is also Bayes-risk consistent, even in the non-convex case.

Proposition 2. *The multiclass bi-tempered logistic loss $L_{t_1}^{t_2}(\hat{\boldsymbol{a}} \mid y)$ is Bayes-risk consistent.*

5 Experiments

We demonstrate the practical utility of the bi-tempered logistic loss function on a wide variety of image classification tasks. For moderate size experiments, we use MNIST dataset of handwritten digits [13] and CIFAR-100, which contains real-world images from 100 different classes [12]. We use ImageNet-2012 [5] for large scale image classification, having 1000 classes. All experiments are carried out using the TensorFlow [1] framework. We use P100 GPU's for small scale experiments and Cloud TPU-v2 for larger scale ImageNet-2012 experiments. An implementation of the bi-tempered logistic loss is available online at: `https://github.com/google/bi-tempered-loss`.

5.1 Corrupted labels experiments

For our moderate size datasets, i.e. MNIST and CIFAR-100, we introduce noise by artificially corrupting a fraction of the labels and producing a new set of labels for each noise level. For all experiments, we compare our bi-tempered loss function against the logistic loss. For MNIST, we use a CNN with two convolutional layers of size 32 and 64 with a mask size of 5, followed by two fully-connected layers of size 1024 and 10. We apply max-pooling after each convolutional layer with a window size equal to 2 and use dropout during training with keep probability equal to 0.75. We use the AdaDelta optimizer [20] with 500 epochs and batch size of 128 for training. For CIFAR-100, we used a Resnet-56 [9] model without batch norm from [8] with SGD + momentum optimizer trained for 50k steps with batch size of 128 and use the standard learning rate stair case decay schedule. For both experiments, we report the test accuracy of the checkpoint which yields the highest accuracy on an identically label-noise corrupted validation set. We search over a set of learning rates for each experiment. For both experiments, we exhaustively search over a number of temperatures within the range $[0.5, 1)$ and $(1.0, 4.0]$ for t_1 and t_2, respectively. The results are

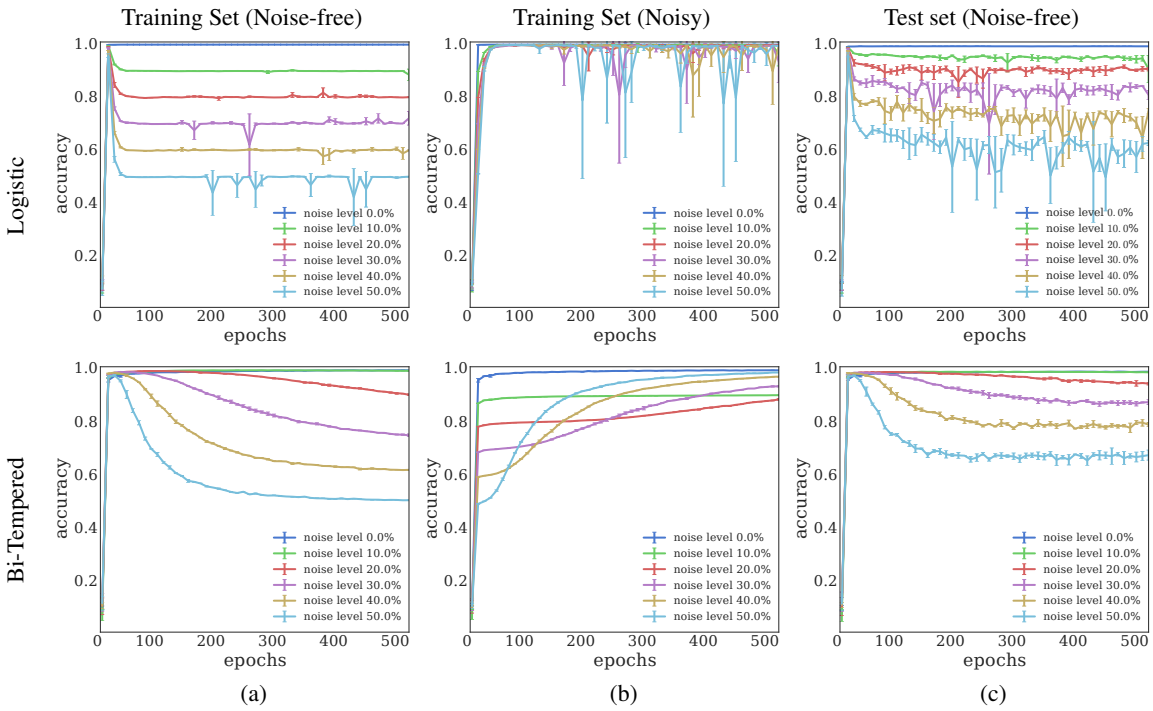

Figure 3: Top-1 accuracy of the models trained using the logistic loss (top) and the bi-tempered loss with $(t_1, t_2) = (0.5, 4.0)$ (bottom) on the noisy MNIST dataset: accuracy on (a) noise-free training set, (b) noisy training set, (c) and noise-free test set. Initially, both models provide better generalization but gradually overfit to the label noise. However, the overfitting for the logistic loss happens much earlier during the optimization. The variance of the model is also much higher for the logistic loss. The bi-tempered loss provides better generalization accuracy overall.

presented in Table 1 where we report the top-1 accuracy on a clean test set. As can be seen, the bi-tempered loss outperforms the logistic loss for all noise levels (including the noise-free case for CIFAR-100). Using our bi-tempered loss function the model is able to continue to perform well even for high levels of label noise whereas the accuracy of the logistic loss drops immediately with a much smaller level of noise. Additionally, in Figure 3, we illustrate the top-1 accuracy on the noise-free and noisy training set, as well the accuracy on the (noise-free) test set for both losses as a function of number of epochs. As can be seen from the figure, initially both models yield a relatively higher test accuracy, but gradually overfit to the label noise in the training set over time. The overfitting to the noise deteriorates the generalization capacity of the models. However, overfitting to the noise happens earlier in the training and is much severe in case of the logistic loss. As a result, the final test accuracy (after 500 epochs) is comparatively much lower than the bi-tempered loss as the noise level increases. Finally, note that the variance of the model is also considerably higher for the logistic loss. This confirms that the bi-tempered loss results in more stable models when the data is noise-corrupted.

Dataset	Loss	Label Noise Level					
		0.0	0.1	0.2	0.3	0.4	0.5
MNIST	Logistic	**99.40**	98.96	98.70	98.50	97.64	96.13
	Bi-Tempered (0.5, 4.0)	99.24	**99.13**	**99.02**	**98.62**	**98.56**	**97.69**
CIFAR-100	Logistic	74.03	69.94	66.39	63.00	53.17	52.96
	Bi-Tempered (0.8, 1.2)	**75.30**	**73.30**	**70.69**	**67.45**	**62.55**	**57.80**

Table 1: Top-1 accuracy on a clean test set for MNIST and CIFAR-100 datasets where a fraction of the training labels are corrupted.

5.2 Large scale experiments

We train state-of-the-art Resnet-18 and Resnet-50 models on the ImageNet-2012 dataset. Note that the ImageNet-2012 dataset is inherently noisy due to some amount of mislabeling. We train on a

4x4 CloudTPU-v2 device with a batch size of 4096. All experiments were trained for 180 epochs, and use the SGD + momentum optimizer with staircase learning rate decay schedule. The results are presented in Table 2. For both architectures we see a significant gain of the robust bi-tempered loss method in top-1 accuracy.

Model	Logistic	Bi-tempered (0.9,1.05)
Resnet18	71.333 ± 0.069	$\mathbf{71.618} \pm 0.163$
Resnet50	76.332 ± 0.105	$\mathbf{76.748} \pm 0.164$

Table 2: Top-1 accuracy on ImageNet-2012 with Resnet-18 and 50 architectures.

6 Conclusion and Future Work

Neural networks on large standard datasets have been optimized along with a large variety of variables such as architecture, transfer function, choice of optimizer, and label smoothing to name just a few. We proposed a new variant by training the network with tunable loss functions. We do this by first developing convex loss functions based on temperature dependent logarithm and exponential functions. When both temperatures are the same, then a construction based on the notion of "matching loss" leads to loss functions that are convex in the last layer. However by letting the temperature of the new tempered softmax function be larger than the temperature of the tempered log function used in the divergence, we construct tunable losses that are non-convex in the last layer. Our construction remedies two issues simultaneously: we construct bounded tempered loss functions that can handle large-margin outliers and introduce heavy-tailedness in our new tempered softmax function that seems to handle small-margin mislabeled examples. At this point, we simply took a number of benchmark datasets and networks for these datasets that have been heavily optimized for the logistic loss paired with vanilla softmax and simply replaced the loss in the last layer by our new construction. By simply trying a number of temperature pairs, we already achieved significant improvements. We believe that with a systematic "joint optimization" of all commonly tried variables, significant further improvements can be achieved. This is of course a more long-term goal. We also plan to explore the idea of annealing the temperature parameters over the training process.

Acknowledgement

We would like to thank Jerome Rony for pointing out the errors in the MNIST experiments.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Ehsan Amid, Manfred K. Warmuth, and Sriram Srinivasan. Two-temperature logistic regression based on the Tsallis divergence. In *22nd International Conference on Artificial Intelligence and Statistics (AISTATS 19)*, 2019.

[3] Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Technical report, University of Pennsylvania, November 2005.

[4] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Nan Ding. *Statistical machine learning in the t-exponential family of distributions*. PhD thesis, Purdue University, 2013.

[7] Nan Ding and S. V. N. Vishwanathan. *t*-logistic regression. In *Proceedings of the 23th International Conference on Neural Information Processing Systems*, NIPS'10, pages 514–522, Cambridge, MA, USA, 2010.

[8] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *International Conference on Learning Representations (ICLR)*, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] D. P. Helmbold, J. Kivinen, and M. K. Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, November 1999.

[11] J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[12] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[13] Yann LeCun and Corinna Cortes. The MNIST database of handwritten digits, 1999.

[14] Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. In *Proceedings of the 25th international conference on Machine learning*, pages 608–615. ACM, 2008.

[15] Jan Naudts. Deformed exponentials and logarithms in generalized thermostatistics. *Physica A*, 316:323–334, 2002.

[16] M. D. Reid and R. C. Williamson. Surrogate regret bounds for proper losses. In *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*, pages 897–904, 2009.

[17] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[18] Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(May):1007–1025, 2007.

[19] Robert C. Williamson, Elodie Vernet, and Mark D. Reid. Composite multiclass losses. *Journal of Machine Learning Research*, 17(223):1–52, 2016.

[20] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[21] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pages 8778–8788, 2018.

A An Iterative Algorithm for Computing the Normalization

Algorithm 1 Iterative algorithm for computing $\lambda_t(\boldsymbol{a})$ (from [2])

Input: Vector of activations $\boldsymbol{a}$, temperature $t > 1$
$\mu \leftarrow \max(\boldsymbol{a})$
$\tilde{\boldsymbol{a}} \leftarrow \boldsymbol{a} - \mu$
while $\tilde{\boldsymbol{a}}$ not converged **do**
  $Z(\tilde{\boldsymbol{a}}) \leftarrow \sum_{i=1}^{k} \exp_t(\tilde{a}_i)$
  $\tilde{\boldsymbol{a}} \leftarrow Z(\tilde{\boldsymbol{a}})^{1-t}(\boldsymbol{a} - \mu\,\mathbf{1})$
end while
Return: $\lambda_t(\boldsymbol{a}) \leftarrow -\log_t \frac{1}{Z(\tilde{\boldsymbol{a}})} + \mu$

B Strong Convexity and Smoothness

The following material for strong convexity and strong smoothness are adopted from [17].

Definition 2 (σ-Strong Convexity)**.** A continuous function F is σ-strongly convex w.r.t. the norm $\|\cdot\|$ over the convex set $\mathcal{S}$ if $\mathcal{S}$ is contained in the domain of F and for any $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{S}$, we have

$$F(\boldsymbol{v}) \geqslant F(\boldsymbol{u}) + \nabla F(\boldsymbol{u}) \cdot (\boldsymbol{v} - \boldsymbol{u}) + \frac{\sigma}{2} \|\boldsymbol{v} - \boldsymbol{u}\|^2 .$$

Lemma 2. *Assume F is twice differentiable. Then F is σ-strongly convex if*

$$\left(\nabla^2 F(\boldsymbol{u})\, \boldsymbol{v}\right) \cdot \boldsymbol{v} \geqslant \sigma \|\boldsymbol{v}\|^2, \quad \forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{S} .$$

Lemma 3. *Let F be a σ-strongly convex function over the non-empty convex set $\mathcal{S}$. For all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{S}$, we have*

$$\frac{\sigma}{2} \|\boldsymbol{u} - \boldsymbol{v}\|^2 \leqslant \Delta_F(\boldsymbol{v}, \boldsymbol{u}) .$$

Proof of Lemma 1. We have $\nabla^2 F(\boldsymbol{u}) = \operatorname{diag}(\boldsymbol{u}^{-t})$. Applying Lemma 3, note that the function

$$(\nabla^2 F_t(\boldsymbol{u}) \cdot \boldsymbol{v}) \cdot \boldsymbol{v} = \sum_i \frac{v_i^2}{u_i^t} ,$$

is unbounded over the set $\mathcal{S} = \{\boldsymbol{v} \in \mathbb{R}_+^d : \|\boldsymbol{v}\|_{2-t} \leqslant B\}$ and the minimum happens at the boundary $\{\|\boldsymbol{v}\|_{2-t} = B\}$.

$$\min_{\boldsymbol{v}} \sum_i \frac{v_i^2}{u_i^t} + \gamma \left(\sum_i v_i^{2-t} - 1\right) \Rightarrow \boldsymbol{v} = B \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_{2-t}} ,$$

where γ is the Lagrange multiplier. Plugging in the solution yields $\sum_i \frac{v_i^2}{u_i^t} \geqslant \frac{1}{B^t} \|\boldsymbol{v}\|_{2-t}^2$. □

Definition 3 (σ-Strong Smoothness)**.** A function differentiable function G is σ-strongly smooth w.r.t. the norm $\|\cdot\|$ if

$$\Delta_G(\boldsymbol{v}, \boldsymbol{u}) \leqslant \frac{\sigma}{2} \|\boldsymbol{v} - \boldsymbol{u}\|^2 .$$

Lemma 4. *Let F be a closed and convex function. Then F is σ-strongly convex w.r.t. the $\|\cdot\|$ if and only if F^*, the dual of F, is $\frac{1}{\sigma}$-strongly smooth w.r.t. the dual norm $\|\cdot\|_*$.*

Proof of Corollary 1. Note that using the duality of the Bregman divergences, we have

$$\Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \Delta_{F_t^*}(f_t(\hat{\boldsymbol{y}}), f_t(\boldsymbol{y})) = \Delta_{F_t^*}(\log_t(\hat{\boldsymbol{y}}), \log_t(\boldsymbol{y})) .$$

Using the strong convexity of F_t and strong smoothness of F_t^*, we have

$$\frac{1}{2B^t} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_{2-t}^2 \leqslant \Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \leqslant \frac{B^t}{2} \|\log_t \boldsymbol{y} - \log_t \hat{\boldsymbol{y}}\|_{\frac{2-t}{1-t}}^2 .$$

Note that $\|\cdot\|_{2-t}$ and $\|\cdot\|_{\frac{2-t}{1-t}}$ are dual norms. Substituting the definition of $\log_t$ to the right-hand-side, we have

$$\frac{B^t}{2} \|\log_t \boldsymbol{y} - \log_t \hat{\boldsymbol{y}}\|_{2-t}^2 = \frac{B^t}{2(1-t)^2} \|\boldsymbol{y}^{1-t} - \hat{\boldsymbol{y}}^{1-t}\|_{\frac{2-t}{1-t}}^2 \leqslant \frac{B^t}{2(1-t)^2} \left(2B^{1-t}\right)^2 = \frac{2B^{2-t}}{(1-t)^2} .$$

□

C Other Tempered Convex Functions

We begin with a list of interesting temperature choices for the convex function F_t and its induced divergence:

t	$F_t(\boldsymbol{y})$	$\Delta_{F_t}(\boldsymbol{y},\hat{\boldsymbol{y}})$	Name
0	$\frac{1}{2}\|\boldsymbol{y}\|_2^2$	$\frac{1}{2}\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2^2$	Euclidean
$\frac{1}{2}$	$\frac{1}{3}\sum_i(4\,y_i^{\frac{4}{3}} - 6\,y_i + 2)$	$\sum_i(\frac{4}{3}y_i^{\frac{3}{2}} - 2y_i\sqrt{\hat{y}_i} + \frac{3}{2}\hat{y}_i^{\frac{3}{2}})$	
1	$\sum_i(y_i \log y_i - y_i + 1)$	$\sum_i(y_i \log \frac{y_i}{\hat{y}_i} - y_i + \hat{y}_i)$	KL-divergence
$\frac{3}{2}$	$\sum_i(-4\,y_i^{\frac{3}{2}} + 2\,y_i + 2)$	$2\sum_i \frac{(\sqrt{y_i}-\sqrt{\hat{y}_i})^2}{\sqrt{\hat{y}_i}}$	Squared Xi on roots
2	$\sum_i(-\log y_i + y_i)$	$\sum_i(\frac{y_i}{\hat{y}_i} - \log \frac{y_i}{\hat{y}_i} - 1)$	Itakura-Saito
3	$\frac{1}{2}\sum_i(-\frac{1}{y_i} + y_i - 2)$	$\frac{1}{2}\sum_i(\frac{1}{y_i} - \frac{2}{\hat{y}_i} + \frac{y_i}{\hat{y}_i^2})$	Inverse

Table 3: Some special cases of the tempered Bregman divergence.

In the construction of the convex function family F_t we used $F_t(x) = \int \log_t(x)$ exploiting the fact that $\log_t(x)$ is strictly increasing. We can also define an alternative convex function family $\widetilde{F}_t$ by utilizing the convexity (respectively, concavity) of the $\log_t$ function for values of $t \geqslant 0$ (respectively, $t \leqslant 0$):

$$\widetilde{F}_t(\boldsymbol{y}) = -\frac{1}{t}\sum_i(\log_t y_i - y_i + 1) = -\frac{1}{t\,(1-t)}\sum_i(y_i^{1-t} - y_i)\,.$$

Note that $\tilde{f}_t(\boldsymbol{y}) \coloneqq \nabla\widetilde{F}(\boldsymbol{y}) = \frac{1-\boldsymbol{y}^{-t}}{t}$ and $\nabla^2\widetilde{F}_t(\boldsymbol{y}) = \mathrm{diag}(\boldsymbol{y}^{-(1+t)})$, thus $\widetilde{F}_t$ is indeed a strictly convex function. The following proposition shows that the Bregman divergence induced by the original and the alternate convex function are related by a temperature shift:

Proposition 3. *For the Bregman divergence induced by the convex function $\widetilde{F}_t$, we have*

$$\forall \boldsymbol{y},\,\hat{\boldsymbol{y}} \in \mathbb{R}_+^k:\quad \Delta_{\widetilde{F}_t}(\boldsymbol{y},\hat{\boldsymbol{y}}) \;=\; \frac{1}{t}\sum_i(\log_t \hat{y}_i - \log_t y_i + (y_i - \hat{y}_i)\,\hat{y}_i^{-t}) \;=\; \Delta_{F_{t+1}}(\boldsymbol{y},\hat{\boldsymbol{y}})\,.$$

The $\widetilde{F}_t$ function is also related to the negative Tsallis entropy over the probability measures $\boldsymbol{y} \in \Delta_+^k$ defined as

$$-H_t^{\text{Tsallis}}(\boldsymbol{y}) = \frac{1}{1-t}\big(1 - \sum_i y_i^t\big) = -\sum_i y_i \log_t \frac{1}{y_i}\,.$$

Note that $(-H_t^{\text{Tsallis}} - (1-t)\,\widetilde{F}_{1-t})$ is an affine function. Thus, the Bregman Divergence induced by $\widetilde{F}_t$, and the one induced by $-H_t^{\text{Tsallis}}$ are also equivalent up to a scaling and a temperature shift. Thus, both functions F_t and $\widetilde{F}_t$ can be viewed as some generalized negative entropy functions. Note that the Bregman divergence induced by $-H_t^{\text{Tsallis}}$ is different from the Tsallis divergence over the simplex, defined as

$$\Delta_t^{\text{Tsallis}}(\boldsymbol{y},\hat{\boldsymbol{y}}) = -\sum_i y_i \log_t \frac{\hat{y}_i}{y_i} = \sum_i y_i^t\,(\log_t y_i - \hat{y}_i)\,.$$

D Convexity of the Tempered Matching Loss

The convexity of the loss function $\Delta_{F_t}\big(\boldsymbol{y}, \exp_t(\hat{\boldsymbol{a}} - \lambda_t(\hat{\boldsymbol{a}})\big)$ with $t \geqslant 1$ for $\hat{\boldsymbol{a}} \in \mathbb{R}^k$ immediately follows from the definition of the matching loss. A more subtle case occurs when $0 \leqslant t < 1$. Note that the range of the combined function $\log_t \circ \exp_t$ does not cover all $\mathbb{R}^k$ as the $\log_t$ function is bounded from below by $-\frac{1}{1-t}$. Therefore, $\textbf{range}(\log_t \circ \exp_t) = \{\boldsymbol{a}' \in \mathbb{R}^k \mid -\frac{1}{1-t} \leqslant \boldsymbol{a}'\}$.

Remark 1. *The normalization function $\lambda_t(\boldsymbol{a})$ satisfies: $\lambda_t(\boldsymbol{a} + b\,\mathbf{1}) = \lambda_t(\boldsymbol{a}) + b$ for $b \in \mathbb{R}$.*

Proof. Note that

$$\sum_i \exp_t((a_i + b) - \lambda_t(\boldsymbol{a} + b\,\mathbf{1})) = \sum_i \exp_t\big(a_i - \underbrace{(\lambda_t(\boldsymbol{a} + b\,\mathbf{1}) - b)}_{=\lambda_t(\boldsymbol{a})}\big) = 1 \quad \text{for} \quad \forall \boldsymbol{a} \in \mathbb{R}^k\,.$$

The claim follows immediately. □

Proposition 4. *The loss function* $\Delta_{F_t}\big(\boldsymbol{y}, \exp_t(\hat{\boldsymbol{a}} - \lambda_t(\hat{\boldsymbol{a}}))$ *for* $0 \leqslant t < 1$ *is convex for*

$$\hat{\boldsymbol{a}} \in \{\boldsymbol{a}' + \mathbb{R}\,\mathbf{1} \mid -\frac{1}{1-t} \leqslant \boldsymbol{a}'\}\,.$$

Proof. Using the definition of the dual function $\check{F}^*$ and its derivative $\check{f}^*$, we can write

$$\begin{aligned}
\Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= F_t(\boldsymbol{y}) - F_t(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot f_t(\hat{\boldsymbol{y}}) && \big(\hat{\boldsymbol{y}} = \exp_t(\hat{\boldsymbol{a}} - \lambda_t(\hat{\boldsymbol{a}})\,\mathbf{1})\big) \\
&= F_t(\boldsymbol{y}) - F_t(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot \log_t \circ \exp_t(\hat{\boldsymbol{a}} - \lambda \mathbf{1}) \\
&= F_t(\boldsymbol{y}) - F_t(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot (\hat{\boldsymbol{a}} - \lambda_t(\hat{\boldsymbol{a}})\,\mathbf{1}) && \big((\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot \mathbf{1} = 1 - 1 = 0\big) \\
&= \underbrace{F_t(\boldsymbol{y}) - \boldsymbol{y} \cdot (\check{f}_t^*)^{-1}(\boldsymbol{y})}_{-\check{F}_t^*((\check{f}_t^*)^{-1}(\boldsymbol{y}))} + \boldsymbol{y} \cdot (\check{f}_t^*)^{-1}(\boldsymbol{y}) \underbrace{- F_t(\hat{\boldsymbol{y}}) + \hat{\boldsymbol{y}} \cdot \hat{\boldsymbol{a}}}_{\check{F}_t^*(\hat{\boldsymbol{a}})} - \boldsymbol{y} \cdot \hat{\boldsymbol{a}} \\
&= \check{F}_t^*(\hat{\boldsymbol{a}}) - \check{F}_t^*((\check{f}_t^*)^{-1}(\boldsymbol{y})) - (\hat{\boldsymbol{a}} - (\check{f}_t^*)^{-1}(\boldsymbol{y})) \cdot \boldsymbol{y} \\
&= \Delta_{\check{F}_t^*}(\hat{\boldsymbol{a}}, (\check{f}_t^*)^{-1}(\boldsymbol{y}))\,.
\end{aligned}$$

Note that the transition from the second line to the third line requires that the assumption $-\frac{1}{1-t} \leqslant \hat{\boldsymbol{a}}$ holds. The dual function $\check{F}_t^*$ satisfies

$$\check{F}_t^*(\boldsymbol{a} + b\,\mathbf{1}) = \lambda_t(\boldsymbol{a} + b\,\mathbf{1}) + \frac{1}{2-t} \sum_i \exp_t\big((a_i + b) - \lambda_t(\boldsymbol{a} + b\,\mathbf{1})\big)^{2-t} = \check{F}_t^*(\boldsymbol{a}) + b\,.$$

Additionally,

$$\Delta_{\check{F}_t^*}(\hat{\boldsymbol{a}} + b\,\mathbf{1}, (\check{f}_t^*)^{-1}(\boldsymbol{y})) = \check{F}_t^*(\hat{\boldsymbol{a}} + b\,\mathbf{1}) - \check{F}_t^*((\check{f}_t^*)^{-1}(\boldsymbol{y})) - (\hat{\boldsymbol{a}} + b\,\mathbf{1} - (\check{f}_t^*)^{-1}(\boldsymbol{y})) \cdot \boldsymbol{y} = \Delta_{\check{F}_t^*}(\hat{\boldsymbol{a}}, (\check{f}_t^*)^{-1}(\boldsymbol{y}))\,.$$

The claim follows by considering the range of $\log_t \circ \exp_t$ and the invariance of the Bregman divergence induced by $\check{F}_t^*$ along $\mathbb{R}\,\mathbf{1}$. □

E Derivatives of Lagrangian and the Bi-tempered Matching Loss

The gradient of $\lambda_t(\boldsymbol{a})$ w.r.t. $\boldsymbol{a}$ can be calculated by taking the partial derivative of both sides of the equality $1 = \sum_j \exp_t(a_j - \lambda_t(\boldsymbol{a}))$ w.r.t. a_i:

$$\begin{aligned}
0 &= \sum_j \exp_t(a_j - \lambda_t(\boldsymbol{a}))^t \left(\delta_{ij} - \frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_i}\right) \\
&= \exp_t\big(a_i - \lambda_t(\boldsymbol{a})\big)^t - \frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_i} \sum_j \exp_t(a_j - \lambda_t(\boldsymbol{a}))^t, \quad \text{where } \delta_{ii} = 1 \text{ and } \delta_{ij} = 0 \text{ for } i \neq j\,. \qquad (11)
\end{aligned}$$

Therefore $\frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_i} = \frac{\exp_t\left(a_i - \lambda_t(\boldsymbol{a})\right)^t}{Z_t}$, where $Z_t = \sum_j \exp_t(a_j - \lambda_t(\boldsymbol{a}))^t$. We conclude that $\frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_i}$ is the "t-escort distribution" of the distribution $\frac{\exp(a_i - \lambda_t(\boldsymbol{a}))}{Z_1}$.

Similarly, the second derivative of $\lambda_t(\boldsymbol{a})$ can be calculated by repeating the derivation on (11):

$$\frac{\partial^2 \lambda_t(\boldsymbol{a})}{\partial a_i \partial a_j} = \frac{1}{Z_t} \sum_{j'} t\, \exp_t\big(a_{j'} - \lambda_t(\boldsymbol{a})\big)^{2t-1} \left(\delta_{ij'} - \frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_i}\right)\left(\delta_{jj'} - \frac{\partial \lambda_t(\boldsymbol{a})}{\partial a_j}\right).$$

Although not immediately obvious from the second derivative, it is easy to show that $\lambda_t(\boldsymbol{a})$ is in fact a convex $\boldsymbol{a}$. Also the derivative of the loss $L_{t_1}^{t_2}(\hat{\boldsymbol{a}}|\boldsymbol{y})$ w.r.t. $\hat{a}_i$ (expressed in terms of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}} = \exp_{t_2}(\hat{\boldsymbol{a}} - \lambda_{t_2}(\hat{\boldsymbol{a}}))$) becomes

$$\begin{aligned}
\frac{\partial L_{t_1}^{t_2}}{\partial \hat{a}_i} &= \sum_j \frac{\partial}{\partial \hat{y}_j}\Big(y_j \log_{t_1} y_j - y_j \log_{t_1} \hat{y}_j - \frac{1}{2-t_1}\, y_j^{2-t_1} + \frac{1}{2-t_1}\, \hat{y}_j^{2-t_1}\Big)\, \frac{\partial \hat{y}_j}{\partial \hat{a}_i} \\
&= \sum_j (\hat{y}_j - y_j)\, \hat{y}_j^{t_2 - t_1} \left(\delta_{ij} - \frac{\hat{y}_i^{t_2}}{\sum_{j'} \hat{y}_{j'}^{t_2}}\right).
\end{aligned}$$

F Proof of Bayes-risk Consistency

The conditional risk of the multiclass loss $\boldsymbol{l}(\hat{\boldsymbol{a}})$ with $l_i \coloneqq \ell(\hat{\boldsymbol{a}}|\, y = i)$, $i \in [k]$ is defined as

$$R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}})) = \sum_i \eta_i \, l_i \,,$$

where $\eta_i \coloneqq P_{\text{UK}}(y = i|\, \boldsymbol{x})$.

Definition 4 (Bayes-risk Consistency)**.** A Bayes-risk consistent loss for multiclass classification is the class of loss functions ℓ for which $\hat{\boldsymbol{a}}^\star$, the minimizer of $R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}}))$, satisfies

$$\arg\min_i \; \ell(\hat{\boldsymbol{a}}^\star|\, y = i) \subseteq \text{argmax}_i \, \eta_i \,.$$

Proof of Proposition 2. For the bi-tempered loss, we have

$$l_i = -\log_{t_1} \exp_{t_2}(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}})) + \frac{1}{2 - t_1} \sum_j \exp_{t_2}(\hat{a}_j - \lambda_{t_2}(\hat{\boldsymbol{a}}))^{2-t_1} \,.$$

Note that the second term is repeated for all classes $i \in [k]$. Also,

$$R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}})) = -\sum_i \eta_i \; \log_{t_1} \exp_{t_2}(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}})) + \frac{1}{2 - t_1} \sum_i \exp_{t_2}(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}}))^{2-t_1} \,.$$

The minimizer of $R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}}))$ satisfies

$$\eta_i = \exp_{t_2}(\hat{a}_i^\star - \lambda_{t_2}(\hat{\boldsymbol{a}}^\star)) \,.$$

Since $-\log_{t_1}$ is a monotonically decreasing function for $0 \leqslant t_1 < 1$, we have

$$\arg\min_i \; \ell(\hat{\boldsymbol{a}}^\star|\, y = i) = \arg\min_i -\log_{t_1} \exp_{t_2}(\hat{a}_i^\star - \lambda_{t_2}(\hat{\boldsymbol{a}}^\star)) = \arg\max_i \hat{a}_i^\star \subseteq \arg\max_i \eta_i \,.$$

□