LEGENDRETRON: Uprising Proper Multiclass Loss Learning

Kevin H. Lam [1]   **Christian Walder** [2 3]   **Spiridon Penev** [1 4]   **Richard Nock** [2 3]

Abstract

Loss functions serve as the foundation of supervised learning and are often chosen prior to model development. To avoid potentially ad hoc choices of losses, statistical decision theory describes a desirable property for losses known as *properness*, which asserts that Bayes' rule is optimal. Recent works have sought to *learn losses* and models jointly. Existing methods do this by fitting an inverse canonical link function which monotonically maps $\mathbb{R}$ to $[0, 1]$ to estimate probabilities for binary problems. In this paper, we extend monotonicity to maps between $\mathbb{R}^{C-1}$ and the projected probability simplex $\tilde{\Delta}^{C-1}$ by using monotonicity of gradients of convex functions. We present LEGENDRETRON as a novel and practical method that jointly learns *proper canonical losses* and probabilities for multiclass problems. Tested on a benchmark of domains with up to 1,000 classes, our experimental results show that our method consistently outperforms the natural multiclass baseline under a t-test at 99% significance on all datasets with greater than 10 classes.

1. Introduction

Loss functions are a pillar of machine learning (ML). In supervised learning, a loss provides a measure of discrepancy between the underlying ground truth and a model's predictions. A learning algorithm attempts to minimise this discrepancy by adjusting the model. In other words, the loss governs how a model learns. The consequence of the bad choice of a loss is oblivious to the qualities of the learning pipeline: it means a poor model in the end. This brings forth the question: which loss is best for the problem at hand?

Statistical decision theory answers this by turning to admissible losses (Savage, 1971); also referred to as proper losses or proper scoring rules (Gneiting & Raftery, 2007). Proper losses are those for which the posterior expected loss value is minimised when probability predictions coincide with the true underlying probabilities. That is, a *proper* loss is one that can induce probability estimates that are *admissible* or optimal. Proper losses have been extensively studied in Shuford et al. (1966); Grünwald & Dawid (2004); Reid & Williamson (2010); Williamson et al. (2016), with the latter two works extending losses to proper composite forms in binary and multiclass settings. Only a handful of proper losses, such as the square and log losses, are commonly used in ML. This is not surprising: properness is an intensional property and does not provide any candidate function. While eliciting some members is possible, extending further requires tuning or adapting the loss as part of the ML task.

There has been a recent surge of interest in doing so for supervised learning, including Mei & Moura (2018); Grabocka et al. (2019); Streeter (2019); Liu et al. (2020); Siahkamari et al. (2020); Sypherd et al. (2022a). However, no connections are made to properness to formulate the losses in these works. On the other hand, several recent works have used properness to formulate losses including Nock & Nielsen (2008); Nock & Menon (2020); Walder & Nock (2020); Sypherd et al. (2022b). Notably, the works of Nock & Menon (2020); Walder & Nock (2020) have proposed algorithms to learn both the link function and linear predictor of logistic regression models by considering both functions to be unknown but learnable; thereby extending Single Index Models (Hardle et al., 1993; Mei & Moura, 2018) and algorithms to learn them (Kakade et al., 2011). Despite the impressive progress in these works, no references have been made to proper losses for multiclass problems.

Background To approach multiclass problems in a principled manner, we generalise logistic regression as follows. For a given invertible and monotonic (see Definition A.1) link function ψ that maps $[0, 1]$ to $\mathbb{R}$ and an input-label pair $(\mathbf{x}, y)$ with $\mathbf{x} \in \mathbb{R}^p$ and $y \in \{-1, 1\}$, logistic regression learns a model of the form $\Pr(Y = 1|\mathbf{x}) = \psi^{-1}(\mathbf{w}^\top \mathbf{x} + b)$ by fitting a coefficient vector $\mathbf{w} \in \mathbb{R}^p$ and an intercept $b \in \mathbb{R}$. A class prediction is then formed as $\hat{y} = \arg\max_{y \in \{-1, 1\}} \Pr(Y = y|\mathbf{x})$. The crucial element

[1] School of Mathematics & Statistics, UNSW Sydney, Australia [2] Google Research [3] ANU College of Engineering, Computing and Cybernetics, The Australian National University, Australia [4] UNSW Data Science Hub (uDASH), UNSW Sydney, Australia. Correspondence to: Kevin Lam <khflam@gmail.com>.

of logistic regression lies in the invertible and monotonic link function that connects probabilities to predictors. Invertibility of the link allows one to identify a unique probability estimate to associate with the predictor. Monotonicity of the link enforces an order to class predictions as elements of $\mathbf{x}$ either increase or decrease monotonically, so that the decision boundary between classes is unique. Loosely speaking, the generalisation of these ideas to multiclass problems with $C \geq 2$ classes is to form probability estimates by using a monotonic link function ψ such that $\psi^{-1}(\mathbf{x}) = (p_1, p_2, \ldots, p_{C-1})$ with $\sum_{k=1}^{C-1} p_k \leq 1$.

Motivation In this work, our interest lies in learning proper losses for multiclass problems. Two observations highlight why this is beneficial: properness directly enforces the same ranking of classes as probabilities without building multiple models, and learned losses can provide better models for related domains. We first note that to approach a multiclass problem with $C > 2$ classes, one would typically pose the problem as multiple 1-vs-rest or 1-vs-1 component problems. Each component problem consists of *positive* and *negative* labels where the former refers to a class of interest, and the latter refers to all other classes in 1-vs-rest or to a single other class of interest in 1-vs-1. An unfortunate consequence in the design of these reductions to binary problems is that they do not include the *admissibility* constraint that probability estimates should rank classes in the same way that true probabilities do. Without loss of generality to the 1-vs-1 approach, we observe this in the following theorem.

Theorem 1.1. *Suppose we use the 1-vs-rest approach to estimate probabilities for a multiclass problem with $C > 2$ classes. Then we learn models of the form*

$$\Pr(\tilde{Y} = c|\mathbf{x}) = \psi_k^{-1}(\mathbf{w}_k^\top \mathbf{x} + b_k)$$

where $c = \begin{cases} +1 & when \ y = k \\ -1 & otherwise \end{cases}$ *for* $k = 1, \ldots, C$. *Probability estimates for any class k is admissible if and only if* $\psi_k^{-1}(\mathbf{w}_k^\top \mathbf{x} + b_k) > \psi_i^{-1}(\mathbf{w}_i^\top \mathbf{x} + b_i)$ *for all $i \neq k$.*

To avoid solving C 1-vs-rest problems through constrained optimisation, we desire an approach that allows us to model multiclass probabilities *simultaneously*, while learning proper multiclass losses which can induce admissible probability estimates for all C classes directly. It has also been shown in the work of Nock & Menon (2020) that loss learning can provide better models for problems in domains related to the original problem where the loss was learned; compared to using uninformed losses such as cross-entropy or log loss. In general, linear models are known to be sensitive to training noise; with the notable result of Long & Servedio (2008) that such noise is sufficient to deteriorate any linear binary classification model to the point that it performs no better than an unbiased coin flip on the original

noise-free domain. The presence of *label noise* in a dataset can be interpreted as a domain that relates to an original noise-free domain, up to some classification noise process. The ideas of loss learning and loss transfer from Nock & Menon (2020) can then be seen as a mechanism that allows us to both overcome training noise and learn accurate models. We illustrate that learning proper multiclass losses can be done by modelling the *canonical link function* which connects probability estimates with a proper loss (see Definition 4.1 and remarks therein). In order to model canonical links flexibly, we form them as composite functions with a fixed component and a learnable component.

Contributions Our main contributions are as follows:

- We derive necessary and sufficient conditions for a composite function in $\mathbb{R}^{C-1}$ to be monotonic and the gradient of a twice-differentiable convex function;

- We derive sufficient conditions for a composite function in $\mathbb{R}^{C-1}$ to be monotonic and the gradient of a twice-differentiable strictly convex function;

- We present LEGENDRETRON as a novel and practical way of learning proper canonical losses and probabilities concurrently in the multiclass problem setting.

Organisation In Section 2, we review existing works which similarly aim to learn losses and models concurrently. In Section 3, we first describe properness and proper canonical losses. In Section 4, we design multiclass canonical link functions through Legendre functions and the (u, v)-geometric structure, and provide conditions for composite functions to be monotonic and gradients of convex functions. We then describe our method, LEGENDRETRON, in detail within Section 5. Lastly, numerical comparisons are provided in Section 6 before concluding in Section 7.

2. Related Work

TRON family of link-learning algorithms The notion of searching for proper losses was first established within Nock & Nielsen (2008). The SLISOTRON algorithm was later presented in Kakade et al. (2011), as the first algorithm designed to learn a model of the form $\Pr(Y = 1|\mathbf{x}) = u(\mathbf{w}^\top \mathbf{x})$ for binary problems, which involves learning the unknown link function $u : \mathbb{R} \to [0, 1]$ assumed to be 1-Lipschitz and non-decreasing, and the vector $\mathbf{w} \in \mathbb{R}^p$ used to form the linear predictor $\mathbf{w}^\top \mathbf{x}$. The algorithm iterates between *Lipschitz isotonic regression* to estimate u and gradient updates to estimate $\mathbf{w}$. A notable and practical shortcoming of SLISOTRON is that the isotonic regression steps to update u do not guarantee u to map to $[0, 1]$. The BREGMANTRON algorithm was later proposed in Nock & Menon (2020), to refine the SLISOTRON algorithm by addressing

this and providing convergence guarantees. By utilising the connection between proper losses and their canonical link functions outlined in Section 4, the BREGMANTRON replaced the link function u with the inverse canonical link $\tilde{\psi}^{-1}$ which guaranteed probability estimates to lie in $[0, 1]$.

ISGP-LINKGISTIC algorithm The idea of using the (u, v)-geometric structure in combination with Legendre functions to learn canonical link functions has recently been explored in the work of Walder & Nock (2020) to propose the ISGP-LINKGISTIC algorithm to learn a model of the form $\Pr(Y = 1|\mathbf{x}) = (u \circ v^{-1})(\mathbf{w}^\top \mathbf{x})$. By the squaring and integration of a Gaussian Process (GP) to yield the Integrated Squared Gaussian Process (ISGP), monotonicity and invertibility of $v^{-1} : \mathbb{R} \to \mathbb{R}$ is guaranteed. The ISGP-LINKGISTIC algorithm exploits this property by choosing a fixed squashing function u separate from the *a priori* ISGP distributed v^{-1}. Inference is performed with a stochastic EM algorithm where the E-step fixes the linear predictor $\mathbf{w}^\top \mathbf{x}$ and applies a Laplace approximation to the latent GP to compute $\mathbb{E}_{q(v^{-1}|\mathbf{w})}[\log p(y|\mathbf{x}, v^{-1})]$, and the M-step maximises this expectation with respect to $\mathbf{w}$. The ISGP-LINKGISTIC algorithm takes a Bayesian approach to learning proper canonical losses jointly with a probability estimator by posterior sampling of inverse canonical links.

3. Definitions and Properties of Losses

In this section, we revisit the notions of proper losses to formulate proper canonical losses in the multiclass setting. We follow the definitions and notations of Williamson et al. (2016) and describe key properties therein, for our discussion of composite multiclass losses.

Let $C \geq 2$ be the total number of classes. Our setting is multiclass probability estimation. Denote the $(C - 1)$-dimensional probability simplex as

$$\Delta^{C-1} = \left\{ p \in \mathbb{R}_+^C : \sum_{i=1}^C p_i = 1 \right\},$$

and its relative interior as

$$\mathrm{ri}(\Delta^{C-1}) = \left\{ p \in \mathbb{R}_+^C : \sum_{i=1}^C p_i = 1, p_i \in (0, 1), \forall i \right\}.$$

Suppose we have a dataset $\mathcal{D}$ of N pairs $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ where each $\mathbf{x}_n \in \mathcal{X} = \mathbb{R}^p$ and $y_n \in \mathcal{Y} = \{1, \ldots, C\}$ denotes an input and a single label respectively. We aim to learn a function $h : \mathcal{X} \to \Delta^{C-1}$ such that $\hat{y}_n \in \arg\max_{c \in \{1, \ldots, C\}} \mathbb{P}(y_n = c|\mathbf{x}_n)$ closely matches y_n.

Consider the label as a random variable $Y \sim \text{Categorical}(p)$ with prior class probabilities $p \in \Delta^{C-1}$. We denote $q \in \Delta^{C-1}$ as the estimated probabilities in the following

definitions. To assess the quality of probability estimates, a loss function can be defined generally as

$$\ell : \Delta^{C-1} \to \mathbb{R}_+^C, \quad \ell(q) = (\ell_1(q), \ldots, \ell_C(q))^\top$$

where ℓ_i is the *partial loss* for predicting q when $y = i$. For a given label y, we can return to *scalar*-valued losses by referring to the y-th partial loss ℓ_y.

Definition 3.1 (conditional Bayes Risk). The conditional risk associated with ℓ is defined as $L(p, q) = \mathbb{E}_{Y \sim \text{Categorical}(p)}[\ell_Y(q)]$ for all $p, q \in \Delta^{C-1}$. The best achievable conditional risk associated with a loss is termed the *conditional Bayes risk* and is defined as

$$\underline{L} : \Delta^{C-1} \to \mathbb{R}_+,$$
$$\underline{L}(p) = \inf_{q \in \Delta^{C-1}} L(p, q) = \inf_{q \in \Delta^{C-1}} \mathbb{E}_{Y \sim \text{Categorical}(p)}[\ell_Y(q)].$$

It is well known that $\underline{L}$ is concave.

Definition 3.2 (Proper Losses). A loss ℓ is *proper* if and only if L is minimized when $q = p$. In other words, $\underline{L}(p) = L(p, p) \leq L(p, q)$ for all $p, q \in \Delta^{C-1}$. Losses where the inequality is strict when $p \neq q$, are termed *strictly proper*.

Remark Properness is an essential property of losses, as optimising a model with respect to a proper loss guides the model's probability estimates towards true posterior class probabilities. Examples of proper losses include the $0 - 1$, square, log, and Matsushita losses (Matusita, 1956).

To draw the connection between a proper loss and its conditional Bayes risk, we require definitions of subgradients and Bregman divergences. Subgradients are a generalisation of gradients and are particularly useful when analysing convex functions that may not be differentiable.

Subgradients For a convex set $S \subseteq \mathbb{R}^n$, the subdifferential of a convex function $f : S \to (-\infty, +\infty]$ at $\mathbf{x} \in S$ is defined as

$$\partial f(\mathbf{x}) = \{\phi \in \mathbb{R}^n : \langle \phi, \mathbf{y} - \mathbf{x} \rangle \leq f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \in \mathbb{R}^n\}$$

where an element $\phi \in \partial f(\mathbf{x})$ is called a *subgradient* of f at $\mathbf{x}$. By convention, we define $\partial f(\mathbf{x}) = \emptyset$ for all $\mathbf{x} \notin S$. Moreover, f is strictly convex if and only if $\partial f(\mathbf{x}) = \{\phi \in \mathbb{R}^n : \langle \phi, \mathbf{y} - \mathbf{x} \rangle < f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \in \mathbb{R}^n\}$.

Bregman divergence For a convex set $S \subseteq \mathbb{R}^n$, and a continuously-differentiable and strictly convex function $f : S \to (-\infty, +\infty]$, the Bregman divergence with generator f is defined for all $\mathbf{x}, \mathbf{y} \in S$ as

$$D_f(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle.$$

The following result is a rewritten characterisation of proper losses through their "Bregman representation", and explicates the connection between a proper loss and its conditional Bayes risk.

Proposition 3.3 ((Williamson et al., 2016, Proposition 7)). *Let $\ell : \Delta^{C-1} \to \mathbb{R}_+^C$ be a loss. ℓ is a (strictly) proper loss if and only if there exists a (strictly) convex function $f : \Delta^{C-1} \to \mathbb{R}$ such that for all $q \in \Delta^{C-1}$, there exists a subgradient $\phi \in \partial f(q)$ such that*

$$L(p,q) = -(p-q)^\top \phi - f(q) \text{ for all } p \in \Delta^{C-1}.$$

Moreover, if $\underline{L}$ is differentiable on $ri(\Delta^{C-1})$ then

$$L(p,q) = (p-q)^\top \ell(q) + \underline{L}(q)$$

where ℓ is the unique proper loss associated with $\underline{L}$ with the property $\nabla \underline{L}(p) = \ell(p), \forall p \in ri(\Delta^{C-1})$.

Remark 3.4. We note that $L(p,q)$ is a Bregman divergence if and only if ℓ is strictly proper due to the requirement of strict convexity of f.

In this work, we seek to learn strictly proper losses ℓ by exploiting the connection $\nabla \underline{L}(p) = \ell(p)$ described in Proposition 3.3. In Section 4, we extend this connection between probabilities and predictors in $\mathbb{R}^{C-1}$ through *canonical link functions*, and describe in detail how strictly proper losses can be learned through this extended connection.

4. Designing Multiclass Canonical Links

In this section, we provide definitions of canonical link functions, Legendre functions and the (u, v)-geometric structure. The latter two structures are essential for the design and learning of canonical link functions. We show that designing a canonical link amounts to designing a composite function that is the gradient of a twice-differentiable and convex function. To this end, we present our key theoretical contributions: conditions for composite functions to be gradients of convex functions.

Composite Form It is often desirable to link predictors with their probability estimates through an invertible link function $\psi : \Delta^{C-1} \to \mathbb{R}^C$. This allows one to uniquely identify probabilities while working with general predictors. It also allows one to define loss functions more generally as $\ell_\psi = \ell \circ \psi^{-1}$ which are referred to as *proper composite losses* when ℓ is proper. Williamson et al. (2016, Proposition 13) shows that a proper composite loss ℓ_ψ is uniquely represented by ℓ and ψ when ℓ_ψ is continuous and invertible.

Proper Canonical Form As elements of Δ^{C-1} are uniquely determined by the first $C - 1$ components, the above properties can be more naturally described by the *projected* probability simplex:

$$\tilde{\Delta}^{C-1} = \left\{ \tilde{p} \in \mathbb{R}_+^{C-1} : \sum_{i=1}^{C-1} \tilde{p}_i \leq 1 \right\}.$$

Define the projection map

$$\Pi : \Delta^{C-1} \to \tilde{\Delta}^{C-1},$$
$$\Pi(p) = (p_1, \ldots, p_{C-1}) \text{ for all } p \in \Delta^{C-1},$$

and its inverse

$$\Pi^{-1} : \tilde{\Delta}^{C-1} \to \Delta^{C-1},$$
$$\Pi^{-1}(\tilde{p}) = \left(\tilde{p}_1, \ldots, \tilde{p}_{C-1}, 1 - \sum_{i=1}^{C-1} \tilde{p}_i \right) \text{ for all } \tilde{p} \in \tilde{\Delta}^{C-1}.$$

Definition 4.1. The projected conditional Bayes risk is defined as $\tilde{\underline{L}} = \underline{L} \circ \Pi^{-1}$. Suppose $\tilde{\underline{L}}$ is differentiable. Then the *canonical link function* is defined as

$$\tilde{\psi} : \tilde{\Delta}^{C-1} \to \mathbb{R}^{C-1}, \quad \tilde{\psi}(\tilde{p}) = -\nabla \tilde{\underline{L}}(\tilde{p}).$$

Williamson et al. (2016, Corollary 32) shows that given a proper loss ℓ, the function $\ell \circ \Pi^{-1} \circ \tilde{\psi}^{-1}$ has components which are convex with respect to the input domain. We refer to such losses as *proper canonical losses* to distinguish them from proper composite losses. The connection between a differentiable conditional Bayes risk, a proper loss, and a canonical link, shown by Proposition 3.3 and Definition 4.1, is given by $\ell = \nabla \underline{L} = -((-\nabla \tilde{\underline{L}} \circ \Pi) \cdot J_\Pi) = -((\tilde{\psi} \circ \Pi) \cdot J_\Pi)$ where J_Π is the Jacobian of Π. This illustrates that one can learn proper canonical losses by modelling either the conditional Bayes risk or its associated canonical link.

Properties of Legendre functions Let $f : \mathbb{R}^{C-1} \to \mathbb{R}$ be continuously differentiable and strictly convex. We refer to f as a *Legendre function*. The *Legendre-Fenchel conjugate* of f, denoted by f^*, is defined as

$$f^* : S \to \mathbb{R},$$
$$f^*(\mathbf{x}^*) = \langle (\nabla f)^{-1}(\mathbf{x}^*), \mathbf{x}^* \rangle - f\big((\nabla f)^{-1}(\mathbf{x}^*)\big).$$

where $S = \{\nabla f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^{C-1}\}$, and f is Legendre if and only if f^* is Legendre. Rockafellar (1970, Theorem 26.5) shows that when the latter holds, $(f^*)^* = f$, and ∇f is continuous and invertible with $\nabla f^* = (\nabla f)^{-1}$. Moreover, if f is twice-differentiable with positive definite Hessian everywhere, then the inverse function theorem yields that f^* is twice-differentiable since $\nabla^2 f^*(\nabla f(\mathbf{x})) = (\nabla^2 f(\mathbf{x}))^{-1}$.

(u, v)**-geometric structure** Amari (2016); Nock et al. (2016); Walder & Nock (2020) state that a general dually flat structure on $\mathbb{R}^{C-1}$ can be defined in terms of an arbitrary strictly convex function ξ. Let u and v be differentiable invertible functions. The pair (u, v) give a dually flat structure on $\mathbb{R}^{C-1}$ if and only if $\nabla \xi = u \circ v^{-1}$. We consider the (u, v)-geometric structure of the Bregman divergence $D_{(-\tilde{\underline{L}})^*}$ which gives $\tilde{\psi}^{-1} = u \circ v^{-1}$.
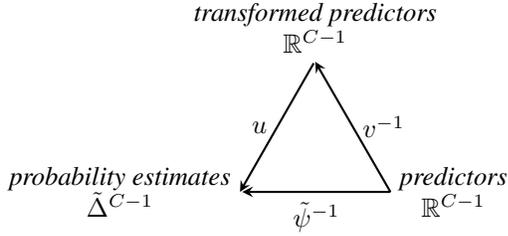
Figure 1. Relationship between predictors and probability estimates through the inverse of the canonical link function under the (u, v)-geometric structure.

Designing links In this work, we focus on the case when $-\tilde{L}$ is twice-differentiable. Note that $-\tilde{L}$ is convex since Π^{-1} is affine and $-\underline{L}$ is convex. Properties of Legendre functions allow us to move from $-\tilde{L}$ to its Legendre-Fenchel conjugate $(-\tilde{L})^*$, and similarly allow us to move from the canonical link $\tilde{\psi}$ to its inverse $\tilde{\psi}^{-1}$. The (u, v)-geometric structure then allows us to flexibly learn $\tilde{\psi}^{-1}$ by splitting it into a *learnable* component v^{-1} and a fixed component u. Fixing u to be a suitable *squashing* function ensures that $\tilde{\psi}^{-1}$ maps to $\tilde{\Delta}^{C-1}$; thereby allowing us to uniquely identify multiclass probabilities associated with predictors from $\mathbb{R}^{C-1}$. On the other hand, v^{-1} can be parameterised by an *invertible neural network* which allows $\tilde{\psi}^{-1}$ to adapt to the multiclass problem at hand. Legendre functions and the (u, v)-geometric structure together yield a more natural and practical design of the canonical link through its inverse since it is often much easier to map inputs from an unbounded space such as $\mathbb{R}^{C-1}$, to a bounded space such as $\tilde{\Delta}^{C-1}$. Figure 1 illustrates how the inverse of the canonical link is modelled using the (u, v)-geometric structure. Loosely speaking, v^{-1} allows one to find better logit representations before they are squashed to probabilities.

Under the (u, v)-geometric structure, if one can prove that $u \circ v^{-1}$ maps to $\tilde{\Delta}^{C-1}$ and is the gradient of a Legendre function f, then one can set $(-\tilde{L})^* = f$ and $\nabla(-\tilde{L})^* = u \circ v^{-1}$ as its corresponding inverse canonical link function by using properties of Legendre functions. This requires showing $u \circ v^{-1}$ is the gradient of a twice-differentiable and strictly convex function. In the following two theorems, we provide conditions where this assertion holds for general composite functions. We defer the background, supporting theorems and proofs of the following results to Sections A, E and F within the Appendices.

Theorem 4.2. *Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ and $g : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be differentiable. Then the following conditions are equivalent:*

1. *$f \circ g = \nabla F$ where F is a twice-differentiable convex function.*

2. *The Jacobian $J_{f \circ g}(\mathbf{x})$ is symmetric for all $\mathbf{x} \in \mathbb{R}^{C-1}$.*

3. *$J_{f \circ g}(\mathbf{x})$ is positive semi-definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$.*

4. *$f \circ g$ is monotone.*

Proof sketch of Theorem 4.2 To claim that a function $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ is the gradient of a convex function $g : \mathbb{R}^{C-1} \to \mathbb{R}$, requires f to satisfy *maximal cyclical monotonicity*. This is a more abstract notion of monotonicity within domains in higher dimensions, and encompasses two notions of monotonicity, namely *maximal monotonicity* and *cyclical monotonicity*. It turns out that it is sufficient to consider monotonicity as maximal monotonicity is automatically guaranteed as our domain is $\mathbb{R}^{C-1}$, and $f \circ g$ is differentiable and therefore continuous.

Theorem 4.2 characterises when a composite function is the gradient of a convex function. It also serves as a convenient and practical criteria to aid model design through a check of positive semi-definiteness for the Jacobian $J_{f \circ g}$. The implications of Theorem 4.2 are profound as it allows us to derive the following sufficient conditions under which the composition of gradients of convex functions is the gradient of a Legendre function.

Theorem 4.3. *Let $f : \mathbb{R}^{C-1} \to S$ and $g : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be differentiable where $S \subseteq \mathbb{R}^{C-1}$, and $J_f(\mathbf{x})$ and $J_g(\mathbf{x})$ are symmetric and positive definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$. Then $f \circ g$ is the gradient of a twice-differentiable Legendre function.*

Proof sketch of Theorem 4.3 Theorem 4.2 tells us it is sufficient to check for positive semi-definiteness of a composite function's Jacobian. Our proof involves a check that all eigenvalues of the Jacobian are positive. This asserts that the composite function is the gradient of a twice-differentiable and strictly convex function.

To use the (u, v)-geometric structure from Section 3 with Theorem 4.3, we can set $f = u$ and $g = v^{-1}$ within Theorem 4.3. This presents an additional requirement that the functions f and g are also invertible. In Section 5, we show how these requirements can be met with our proposed algorithm, LEGENDRETRON.

5. Learning Proper Canonical Multiclass Losses: LEGENDRETRON

In this section, we present LEGENDRETRON, our main algorithmic contribution for learning proper canonical losses for multiclass probability estimation. With the theory of Legendre functions, (u, v)-geometric structure and Theorem 4.3 in hand to support our approach, we now present LEGENDRETRON in detail, as an extension of generalised linear models and Single Index Models for multinomial logistic regression.

Model Given a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, we have the classification model

$$y_n | \mathbf{x}_n \sim \text{Categorical}(\hat{p}(\mathbf{z}_n)) \text{ where } \mathbf{z}_n = \mathbf{W}\mathbf{x}_n + \mathbf{b}$$

where $\mathbf{W} \in \mathbb{R}^{(C-1) \times p}$, $\mathbf{b} \in \mathbb{R}^{C-1}$ and $\hat{p}(\mathbf{z}_n) = (u \circ v^{-1})(\mathbf{z}_n)$ with u chosen as a squashing function that maps to $\tilde{\Delta}^{C-1}$ and $v^{-1} = \nabla g$ for a twice-differentiable and strictly convex function g. We leave the specification of a suitable squashing function u as a modelling choice and provide a natural choice at the end of this section.

For any $B \in \mathbb{Z}_+$, let $g_1, g_2, \ldots g_B$ be fully input convex neural networks (FICNN) investigated in Amos et al. (2017). We set $v^{-1} = \nabla g = (\nabla g_1) \circ (\nabla g_2) \circ \cdots \circ (\nabla g_B)$. For each g_i, we use the same architecture as Huang et al. (2021) which is described as

$$\mathbf{z}_{i,1} = l_{i,1}^+(\mathbf{x})$$
$$\mathbf{z}_{i,k} = l_{i,k}(\mathbf{x}) + l_{i,k}^+(s(\mathbf{z}_{i,k-1})) \text{ for } k = 2, \ldots, M+1,$$
$$h_i(\mathbf{x}) = s(\mathbf{z}_{i,M+1}),$$
$$g_i(\mathbf{x}) = s(w_{i,0}) h_i(\mathbf{x}) + s(w_{i,1}) \frac{\|\mathbf{x}\|^2}{2}$$

where we denote $l_{i,k}^+$ as a linear layer with *positive* weights, $l_{i,k}$ as a linear layer with unconstrained weights, $w_{i,0}, w_{i,1} \in \mathbb{R}$ are unconstrained parameters and $s(x) = \log(1 + e^x)$ is the softplus function with $s(\mathbf{x})$ denoting the softplus function applied elementwise on $\mathbf{x}$. In particular, $l_{i,M+1}$ and $l_{i,M+1}^+$ are linear layers that map to $\mathbb{R}$ while for each $k = 1, \ldots M$, $l_{i,k}$ and $l_{i,k}^+$ are hidden layers that map to $\mathbb{R}^H$ for a chosen dimension size $H \in \mathbb{Z}_+$. With this setup, each g_i is strongly convex (and therefore strictly convex) with an invertible gradient and positive definite Hessian for all $\mathbf{x} \in \mathbb{R}^{C-1}$ due to the quadratic term within each g_i.

We now show that, when equipped with a suitable squashing function u, any function learned by LEGENDRETRON is a valid inverse canonical link function. We turn to a modified version of the *LogSumExp* function previously studied in Nielsen & Hadjeres (2018) and describe its main properties within the following theorem.

Theorem 5.1. *Let $f(\mathbf{x}) = \log\left(1 + \sum_{k=1}^{C-1} \exp(x_k)\right)$. The key properties of f are:*

- *f is strictly convex with invertible gradient*

$$u : \mathbb{R}^{C-1} \to \tilde{\Delta}^{C-1},$$
$$u(\mathbf{x}) = \left(\frac{\exp(x_i)}{1 + \sum_{k=1}^{C-1} \exp(x_k)}\right)_{1 \leq i \leq C-1}.$$

- *the Hessian of f, given by $J_u(\mathbf{x})$, is positive definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$.*

We refer to f and u as LogSumExp$^+$ and softmax$^+$ respectively. Let $v^{-1} : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be defined as

$$v^{-1} = (\nabla g_1) \circ (\nabla g_2) \circ \cdots \circ (\nabla g_B)$$

where $g_1, g_2, \ldots g_B$ are FICNNs. Then any function $u \circ v^{-1}$ learned by LEGENDRETRON is the gradient of a twice-differentiable Legendre function and is therefore, the inverse of a canonical link function.

With this specification, any function $u \circ v^{-1}$ learned via LEGENDRETRON is the gradient of a twice-differentiable Legendre function which can serve as an inverse canonical link function. Moreover, we can deduce that any inverse canonical link or gradient of a twice-differentiable Legendre function, can be approximated by the architecture of $u \circ v^{-1}$ defined in Theorem 5.1.

Corollary 5.2. *Let $u : \mathbb{R}^{C-1} \to \tilde{\Delta}^{C-1}$ and $v^{-1} : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be defined as in Theorem 5.1 with v^{-1} parameterised by $\boldsymbol{\theta}$. Define $\mathcal{C}(\Omega)$ as the set of twice-differentiable convex functions with positive definite Hessian everywhere for a compact set $\Omega \subset \mathbb{R}^{C-1}$, and $\mathcal{F} = \{f : f : \mathbb{R}^{C-1} \to \tilde{\Delta}^{C-1}, \exists \boldsymbol{\theta} \text{ such that } u \circ v^{-1} = f\}$. Then $\mathcal{F}$ is dense in $\mathcal{C}(\Omega)$.*

Algorithm 1 describes LEGENDRETRON in detail. We conclude this section with remarks on our model design, and connections between LEGENDRETRON, multinomial logistic regression and the log loss.

Remark 5.3. The basis of our design comes from requiring the determinant of $J_{u \circ v^{-1}}(\mathbf{x})$ to satisfy $|J_{u \circ v^{-1}}(\mathbf{x})| = |J_u(v^{-1}(\mathbf{x}))||J_{v^{-1}}(\mathbf{x})| > 0$ for all $\mathbf{x} \in \mathbb{R}^{C-1}$. A direct way to guarantee this is by setting u and v^{-1} to be functions with known positive definite Hessians. To the best of our knowledge, only the CP-Flow architecture (Huang et al., 2021) satisfies this property among invertible networks in normalising flows literature; making it our choice for v^{-1}. While it is possible to set other functions as u, it is generally difficult to elicit *invertible* functions that squash inputs to $\tilde{\Delta}^{C-1}$ aside from variants of softmax. We have set $u = \text{softmax}^+$ as it simplifies to the well-known sigmoid when $C = 2$, which was used as the analogous squashing function in ISGP (Walder & Nock, 2020).

Remark 5.4. As LogSumExp$^+$ is twice-differentiable and Legendre, its gradient softmax$^+$ is a valid inverse canonical link function since it maps to $\tilde{\Delta}^{C-1}$. However, we note that setting $\tilde{\psi}^{-1} = \text{softmax}^+$ results in learning only the parameters $\mathbf{W}$ and $\mathbf{b}$ which is equivalent to formulating multinomial logistic regression as a generalised linear model with link $\tilde{\psi}(\tilde{p}) = \left(\log\left(\frac{\tilde{p}_i}{1 - \sum_{k=1}^{C-1} \tilde{p}_k}\right)\right)_{1 \leq i \leq C-1}$ with corresponding proper loss $\ell = -((\tilde{\psi} \circ \Pi) \cdot J_\Pi)$ which can be shown to be the log loss (or *cross-entropy*). That is, LEGENDRETRON with v^{-1} as the identity map is equivalent to multinomial logistic regression or logistic regression when

Algorithm 1 LEGENDRETRON
___
Input: sample $\mathcal{S} \subset \mathcal{D}$, number of iterations T, number of FICNNs B, hidden layer dimension size H, number of layers M, squashing function u.

Initialise $\mathbf{W}$ and $\mathbf{b}$.

Initialise $g_1, g_2, \ldots g_B$ each with M layers of dimension size H, and denote their joint set of parameters $\boldsymbol{\theta}$.

for $i = 1$ **to** T **do**
    Set $v^{-1} = (\nabla g_1) \circ (\nabla g_2) \circ \cdots \circ (\nabla g_B)$.
    for each $(\mathbf{x}_n, y_n) \in \mathcal{S}$ **do**
        Compute $\mathbf{z}_n = \mathbf{W}\mathbf{x}_n + \mathbf{b}$.
        Compute $\hat{p}(\mathbf{z}_n) = (u \circ v^{-1})(\mathbf{z}_n)$.
    end for
    Compute $\mathbb{E}_{\mathcal{S}}[\mathcal{L}(\hat{p}(\mathbf{z}), y)]$ by Monte Carlo where $\mathcal{L}$ is the log-likelihood of the Categorical distribution.
    Update $\mathbf{W}$, $\mathbf{b}$ and $\boldsymbol{\theta}$ by backpropagation.
end for
Output: $\mathbf{W}$, $\mathbf{b}$ and $g_1, g_2, \ldots g_B$.
___

$C = 2$. In this case, Algorithm 1 would only optimise parameters of a linear model without loss learning; by using the log loss. Comparisons between LEGENDRETRON against multinomial logistic regression or logistic regression illustrate performance differences between learning a proper loss for the dataset and optimising with respect to log loss (cross-entropy). These results are provided in Tables 1 and 3, and Figure 2.

6. Experiments

In this section, we provide numerical comparisons between LEGENDRETRON, multinomial logistic regression and other existing methods that also aim to jointly learn models and proper canonical losses. For our experiments, we set softmax$^+$ as the squashing function u for both LEGENDRETRON and multinomial logistic regression. For a practical and numerically stable implementation, we also map probability estimates to the log scale by deriving an alternate Log-Sum-Exp trick for softmax$^+$. We defer the full experimental details to Appendix I.

All experiments were performed using PyTorch (Paszke et al., 2019) and took roughly one CPU month to complete[1]. CPU run times for the aloi dataset, which had the largest number of classes $(1,000)$, were respectively 4 hours and 0.75 hours for LEGENDRETRON and multinomial logistic regression. We note that the difference in run times for this experiment are in part due to the larger number of epochs (360), larger number of blocks B, autograd and backpropagation operations to update v^{-1} for a much larger

___
[1]The total run time for our experiments is favourable relative to the reported two CPU months for the ISGP-LINKGISTIC algorithm from Walder & Nock (2020).

Table 1. Test AUC for generalised linear models with various link methods (ordering in decreasing average). See text for details.

	MNIST	FMNIST
LEGENDRETRON	99.9%	99.2%
ISGP-Linkgistic	99.9%	99.2%
GP-Linkgistic	99.9%	99.1%
Logistic regression	99.9%	98.5%
GLMTron	99.6%	98.1%
BREGMANTRON	99.7%	97.9%
BREGMANTRON$_{\texttt{label}}$	99.6%	97.7%
BREGMANTRON$_{\texttt{approx}}$	99.3%	94.6%
SLISOTRON	94.6%	90.7%

number of classes. Average GPU run times on a P100 for MNIST experiments in Table 2, were 2.32 and 2.12 hours for VGGTRON and VGG respectively. These run times demonstrate the relative efficiency and applicability of loss learning for most datasets.

MNIST Binary Problems   Binary problems are a special case of our setting where $C = 2$, so LEGENDRETRON is readily applicable. In Table 1, we compared LEGENDRETRON against ISGP-LINKGISTIC (Walder & Nock, 2020) and BREGMANTRON (Nock & Menon, 2020), as both algorithms also aim to learn proper canonical losses for binary problems. We also compared with other baselines in these two works including the SLISOTRON algorithm from Kakade et al. (2011). Experiment details can be found in Section 6 of Nock & Menon (2020). Our model successfully matches the (binary specific) ISGP-LINKGISTIC baseline, which was the strongest algorithm in test AUC performance from the experiments of Walder & Nock (2020).

MNIST Multiclass Problems using Linear Models   For the three MNIST-like datasets (LeCun et al., 2010; Xiao et al., 2017; Clanuwat et al., 2018), we compared LEGENDRETRON against multinomial logistic regression and ISGP-LINKGISTIC, since the latter is the strongest algorithm in ten-class classification test accuracy performance based on the experiments within Walder & Nock (2020). ISGP-LINKGISTIC approaches the multiclass problem by learning proper canonical losses for the 10 component 1-vs-rest problems. Our experimental results in Figure 2 show that LEGENDRETRON and multinomial logistic regression outperform the ISGP-LINKGISTIC baseline on all three datasets. These results illustrate our conjecture that properness with respect to losses and models in component problems in a multiclass setting, does not imply optimality of class predictions or probability estimates. By respecting the true problem structure, proper multiclass losses allow the model to learn probability estimates that are able to better distinguish between all the classes at hand. Our results also

Table 2. Test classification accuracies of VGGTron and VGG for the *MNIST*, *Kuzushiji-MNIST* and *Fashion-MNIST* datasets. See text for details.

	MNIST	FMNIST	KMNIST
VGGTRON	99.59%	92.88%	98.26%
VGG	99.40%	92.80%	98.12%

show that LEGENDRETRON either matches or outperforms multinomial logistic regression on all three datasets. This is most notable on the Kuzushiji-MNIST dataset where LEGENDRETRON outperforms multinomial logistic regression by a reasonable margin.

MNIST Multiclass Problems using Nonlinear Models
We note that the architecture of $u \circ v^{-1}$ does not restrict us to linear models. In Table 2, we provided experimental results of how loss learning can improve non-linear models. Specifically, we replace the linear model components of LEGENDRETRON and multinomial logistic regression with a VGG-5 architecture; which we refer to as VGGTRON and VGG respectively in Table 2. Our results show that learning proper losses can improve the performance of non-linear models. A more comprehensive survey of how loss learning can improve model performance for classification tasks is an avenue for future work, due to the variety of architectures and datasets.

Other Multiclass Problems and Label Noise We also compared LEGENDRETRON against multinomial logistic regression on 15 datasets that are publicly available from the LIBSVM library (Chang & Lin, 2011), the UCI machine learning repository (Asuncion & Newman, 2007; Dua & Graff, 2017), and the Statlog project (King et al., 1995). We note that we did not compare our proposed method with other multiclass classification methods such as kernel methods explored in Zien & Ong (2007) and Li et al. (2018), as these methods are centred on the task of classification, whereas our focus is on jointly learning multiclass probabilities and proper canonical losses through the canonical link function. To assess the robustness against label noise, we also compare the classification accuracy of LEGENDRETRON and multinomial logistic regression where labels in the training set are corrupted with probability η. That is, for any true label y_n, we instead train our models on the potentially corrupted label given by

$$\tilde{y}_n = \begin{cases} y_n & \text{with probability } 1 - \eta, \\ c & \text{with probability } \eta \text{ where } c \in \mathcal{Y} \setminus \{y_n\} \end{cases}.$$

We applied *symmetric* label noise in our experiments which is the case where the probability of $\tilde{y}_n = c$ for each $c \in \mathcal{Y} \setminus \{y_n\}$ is $\frac{\eta}{C-1}$. We run both LEGENDRETRON and

multinomial logistic regression for each dataset 20 times, where each run randomly splits the dataset into 80% training and 20% testing sets. Our results in Table 3 show that LEGENDRETRON outperforms multinomial logistic regression under a t-test at 99% significance for most datasets and label noise settings. The performance of LEGENDRETRON is on par with multinomial logistic regression on the *svmguide2*, *wine* and *iris* datasets. Multinomial logistic regression only statistically outperforms LEGENDRETRON on the *dna* dataset. LEGENDRETRON consistently outperforms multinomial logistic regression especially strongly on problems where the number of classes is greater than 10. The better performance of LEGENDRETRON can partially be attributed to greater model capacity afforded by v^{-1} which allows logit estimates to adapt to problems with more classes by adding more nonlinearities. We note that the Lipschitz and strongly monotone properties of $\nabla g_1, \nabla g_2, \ldots, \nabla g_B$ are dependent only on inputs which remain uncorrupted so probability estimates would respect the true rankings of classes by design. We conjecture that these properties allow for more adaptive shrinking or expanding of logit variations depending on the level of label noise present; offering a form of tolerance to label noise.

7. Conclusion and Broader Impact

In this work, we proposed a general approach which jointly learns proper canonical losses and multiclass probabilities. Our contributions advance the recent work on learning losses with probabilities based on the seminal work within Kakade et al. (2011); Nock & Menon (2020); Walder & Nock (2020) by providing a natural extension to the multiclass setting. The practical nature and generality of our model is owed to the general parameterisation of Fully Input Convex Neural Networks, with theoretical support from Legendre functions, structures from information geometry and hallmark results from convex analysis.

By grounding losses in properness for the multiclass setting, we have demonstrated that our model improves upon existing methods that aim to solve multiclass problems through binary reductions, and also outperforms the natural baseline of multinomial logistic regression. Separately, we have also provided conditions under which a composition of gradients of differentiable convex functions is the gradient of another differentiable convex function.

While it is possible for advances in machine learning to bring positive and negative societal impacts, the present work remains general and not specific to any application so it is unlikely to bring about any immediate negative societal impact. We anticipate that our results will find applications in multiclass classification and probability estimation, as well as variational inference.
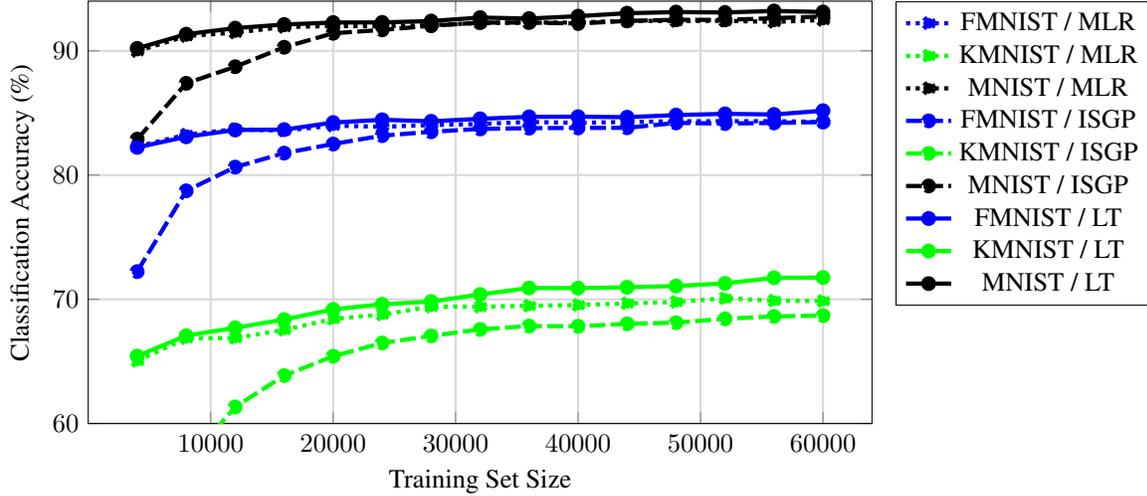
Figure 2. Test performance v.s. training set size for the *MNIST*, *Kuzushiji-MNIST* and *Fashion-MNIST* datasets. We compare the ten-class classification accuracy of LEGENDRETRON (LT), multinomial logistic regression (MLR) and ISGP-LINKGISTIC (ISGP) where the ISGP combines 10 one-vs-rest binary models while the former two algorithms model the probabilities of all 10 classes jointly.

Table 3. Average test classification accuracies (%) for LEGENDRETRON (LT) and multinomial logistic regression (MLR) on LIBSVM, UCI and Statlog datasets; at varying levels of label noise (η). Numbers of the method are bolded when it performs statistically better at a significance level of 99% under a t-test. Absence of bolding indicates both methods have statistically similar performance.

Dataset	# Features	# Classes	$\eta = 0\%$		$\eta = 20\%$		$\eta = 50\%$	
			LT	MLR	LT	MLR	LT	MLR
aloi	128	1,000	**88.11±0.03**	10.34±0.42	**83.03±0.06**	7.07±0.45	**75.23±0.07**	3.53±0.29
sector	55,197	105	**89.71±0.18**	8.77±0.73	**81.00±0.28**	4.12±0.44	**57.38±0.31**	3.17±0.47
letter	16	26	**79.82±0.30**	53.37±0.25	**74.17±0.21**	51.24±0.28	**64.28±0.26**	46.78±0.41
news20	62,061	20	**75.65±0.72**	63.09±0.58	**73.48±0.20**	50.49±1.16	**51.72±0.16**	31.54±1.83
Sensorless	48	11	**88.31±0.19**	34.42±0.46	**82.63±0.99**	32.70±0.50	**52.02±0.78**	29.35±0.84
vowel	10	11	**79.72±1.03**	44.58±1.08	**63.77±1.36**	43.44±1.17	**40.94±1.61**	35.42±1.45
usps	256	10	**95.23±0.16**	93.79±0.17	92.88±0.15	92.95±0.19	90.23±0.26	90.48±0.27
segment	19	7	**95.95±0.24**	87.86±0.40	**92.21±0.40**	87.28±0.40	**86.56±0.47**	82.75±0.46
satimage	36	6	**86.97±0.19**	83.93±0.28	**84.93±0.25**	81.16±0.28	77.44±0.29	77.39±0.29
glass	36	6	58.72±1.94	52.09±1.88	53.72±1.98	50.47±2.11	42.56±1.92	45.47±1.67
vehicle	18	4	**76.91±0.65**	64.94±0.43	**73.59±0.79**	63.06±0.53	**60.94±1.25**	55.18±1.20
dna	180	3	92.79±0.30	**94.43±0.19**	82.61±0.51	**89.55±0.31**	58.23±1.05	**64.18±0.81**
svmguide2	20	3	56.01±1.40	56.01±1.40	56.01±1.40	56.01±1.40	51.65±2.81	52.41±3.04
wine	13	3	96.94±1.14	97.78±0.59	90.97±1.92	96.25±0.99	69.44±2.89	77.36±2.46
iris	4	3	86.67±3.89	83.00±2.08	80.00±3.71	81.50±2.27	63.50±5.13	70.67±3.83

9

References

Amari, S. *Information geometry and its applications*, volume 194. Springer, 2016.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155. PMLR, 06–11 Aug 2017.

Asplund, E. A monotone convergence theorem for sequences of nonlinear mappings. In Browder, F. E. (ed.), *Proceedings of Symposia in Pure Mathematics*, volume 18, pp. 1–9, Chicago, IL, USA, 1968. American Mathematical Society.

Asuncion, A. and Newman, D. UCI repository of machine learning databases, 2007.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2011. ISBN 9781441994660.

Bhatia, R. *Matrix Analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer New York, 2013. ISBN 9781461206538.

Borwein, J. and Wiersma, H. Asplund decomposition of monotone operators. *SIAM Journal on Optimization*, 18 (3):946–960, 2007.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018.

Dua, D. and Graff, C. UCI machine learning repository, 2017.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Grabocka, J., Scholz, R., and Schmidt-Thieme, L. Learning surrogate losses. *arXiv preprint arXiv:1905.10108*, 2019.

Grünwald, P. D. and Dawid, A. P. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367 – 1433, 2004.

Hardle, W., Hall, P., and Ichimura, H. Optimal Smoothing in Single-Index Models. *The Annals of Statistics*, 21(1): 157 – 178, 1993.

Huang, C.-W., Chen, R. T. Q., Tsirigotis, C., and Courville, A. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021.

Kakade, S. M., Kanade, V., Shamir, O., and Kalai, A. Efficient learning of generalized linear and single index models with isotonic regression. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

King, R. D., Feng, C., and Sutherland, A. Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):289–333, 1995.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Li, J., Liu, Y., Yin, R., Zhang, H., Ding, L., and Wang, W. Multi-class learning: From theory to algorithm. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Liu, L., Wang, M., and Deng, J. A unified framework of surrogate loss by refactoring and interpolation. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12348 of *Lecture Notes in Computer Science*, pp. 278–293. Springer, 2020.

Long, P. M. and Servedio, R. A. Random classification noise defeats all convex potential boosters. In Cohen, W. W., McCallum, A., and Roweis, S. T. (eds.), *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pp. 608–615. ACM, 2008.

Matusita, K. Decision rule, based on the distance, for the classification problem. *Annals of the institute of statistical mathematics*, 8:67–77, 1956.

Meenakshi, A. and Rajian, C. On a product of positive semidefinite matrices. *Linear algebra and its applications*, 295(1):3–6, 1999. ISSN 00243795.

Mei, J. and Moura, J. M. F. SILVar: Single index latent variable models. *IEEE Transactions on Signal Processing*, 66(11):2790–2803, 2018.

Nielsen, F. and Hadjeres, G. Monte carlo information geometry: The dually flat case. *CoRR*, abs/1803.07225, 2018.

Nock, R. and Menon, A. Supervised learning: no loss no cry. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7370–7380. PMLR, 13–18 Jul 2020.

Nock, R. and Nielsen, F. On the efficient minimization of classification calibrated surrogates. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Nock, R., Nielsen, F., and Amari, S. On conformal divergences and their population minimizers. *IEEE Transactions on Information Theory*, 62(1):527–538, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Reid, M. D. and Williamson, R. C. Composite binary losses. *Journal of Machine Learning Research*, 11(83):2387–2422, 2010.

Rockafellar, R. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, 1970. ISBN 0691080690.

Rockafellar, R., Wets, M., and Wets, R. *Variational Analysis*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2009. ISBN 9783540627722.

Savage, L. J. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971. ISSN 01621459.

Shuford, E. H., Albert, A., and Edward Massengill, H. Admissible probability measurement procedures. *Psychometrika*, 31(2):125–145, 1966.

Siahkamari, A., Xia, X., Saligrama, V., Castañón, D., and Kulis, B. Learning to approximate a bregman divergence. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3603–3612. Curran Associates, Inc., 2020.

Streeter, M. Learning effective loss functions efficiently. *CoRR*, abs/1907.00103, 2019.

Sypherd, T., Diaz, M., Cava, J. K., Dasarathy, G., Kairouz, P., and Sankar, L. A tunable loss function for robust classification: Calibration, landscape, and generalization. *IEEE Transactions on Information Theory*, 68(9):6021–6051, 2022a.

Sypherd, T., Nock, R., and Sankar, L. Being properly improper. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20891–20932. PMLR, 17–23 Jul 2022b.

Walder, C. and Nock, R. All your loss are belong to bayes. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18505–18517. Curran Associates, Inc., 2020.

Williamson, R. C., Vernet, E., and Reid, M. D. Composite multiclass losses. *Journal of Machine Learning Research*, 17(222):1–52, 2016.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, 2017.

Zien, A. and Ong, C. S. Multiclass multiple kernel learning. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 1191–1198. Association for Computing Machinery, 2007. ISBN 9781595937933.

A. Convex Analysis: Relevant Background and List of Theorems

A.1. Background

To motivate the results studied in this section, we first note that in general, the composition of two monotone functions in $\mathbb{R}^{C-1}$ is not necessarily another monotone function in $\mathbb{R}^{C-1}$. This means that methods to design monotonic functions in $\mathbb{R}$ cannot be applied to functions defined on $\mathbb{R}^{C-1}$, leaving the methods discussed in Section 2 unsuitable for the general multiclass setting. Separately, we note that the composition of two gradients of differentiable convex functions is *not* necessarily the gradient of another convex function. In general, to claim that a function $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ is the gradient of a convex function $g : \mathbb{R}^{C-1} \to \mathbb{R}$, requires f to satisfy a notion of monotonicity generalised to higher dimensions. The connection between convex functions and their gradients is well known in convex analysis via the notion of *maximal cyclically monotone* functions. This is a combination of two notions of monotonicity: *maximal monotonicity* and *cyclical monotonicity*. These are defined within the following list of definitions and theorems.

A.2. List of Theorems

Definition A.1 ((Rockafellar et al., 2009, Definition 12.1))**.** A function $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ is monotone if $\langle f(\mathbf{x}) - f(\mathbf{z}), \mathbf{x} - \mathbf{z} \rangle \geq 0$ for all $\mathbf{x}, \mathbf{z} \in \mathbb{R}^{C-1}$. Moreover, it is strictly monotone when the inequality is strict whenever $\mathbf{x} \neq \mathbf{z}$.

The following two definitions require the notion of the graph of a function $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ which is defined as $\mathrm{gph}(f) = \{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathbb{R}^{C-1}, \mathbf{y} \in f(\mathbf{x})\}$.

Definition A.2 ((Bauschke & Combettes, 2011, Definition 20.20))**.** Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be a monotone function. Then f is maximally monotone if there exists no monotone function $g : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ such that $\mathrm{gph}(f) \subsetneq \mathrm{gph}(g)$.

Definition A.3 ((Bauschke & Combettes, 2011, Definition 22.10))**.** Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$. For an arbitrary integer $n \geq 2$, f is n-cyclically monotone if for any $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,n} \subset \mathrm{gph}(f)$ it follows that

$$\sum_{i=1}^{n} \langle \mathbf{y}_i, \mathbf{x}_{i+1} - \mathbf{x}_i \rangle \leq 0 \text{ where } \mathbf{x}_{n+1} = \mathbf{x}_1.$$

f is cyclically monotone if it is n-cyclically monotone for any integer $n \geq 2$. In addition, if $\mathrm{gph}(f) \not\subset \mathrm{gph}(g)$ for any cyclically monotone function $g \neq f$ then f is maximal cyclically monotone.

Theorem A.4 ((Rockafellar et al., 2009, Theorems 12.17 & 12.25))**.** *Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$. Then $f = \nabla h$ for a differentiable convex function $h : \mathbb{R}^{C-1} \to \mathbb{R}$ if and only if f is maximal cyclically monotone. That is, f is maximally monotone and cyclically monotone.*

Theorem A.5 ((Rockafellar et al., 2009, Proposition 12.3))**.** *Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be a differentiable function. Then f is monotone if and only if $\nabla f(\mathbf{x})$ is positive semi-definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$. Moreover, if $\nabla f(\mathbf{x})$ is positive definite for all $\mathbf{x} \in \mathbb{R}^{C-1} \setminus \{0\}$ then f is strictly monotone.*

Theorem A.6 ((Bauschke & Combettes, 2011, Corollary 20.25))**.** *Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be a monotone and continuous function. Then f is maximally monotone.*

Theorem A.7 ((Borwein & Wiersma, 2007, Theorem 3))**.** *Let $f : \mathbb{R}^{C-1} \to \mathbb{R}^{C-1}$ be maximally monotone and continuously differentiable. Then $f(\mathbf{x}) = \nabla F(\mathbf{x}) + L\mathbf{x}$ where F is a differentiable convex function, and L is a skew symmetric matrix.*

Theorem A.8 ((Meenakshi & Rajian, 1999, Theorem 3))**.** *Let $A, B \in \mathbb{R}^{(C-1) \times (C-1)}$ be symmetric and positive semi-definite matrices. Then AB is positive semi-definite if and only if it is symmetric.*

Theorem A.9 ((Bhatia, 2013, Theorem VIII.4.6))**.** *Let $V \subset \mathbb{R}^{(C-1) \times (C-1)}$ be a real vector space whose elements are matrices with real eigenvalues. Denote $\lambda_i(M)$ as the i-th smallest eigenvalue for any matrix $M \in V$. Let $A, B \in V$ then*

$$\lambda_i(A) + \lambda_1(B) \leq \lambda_i(A + B) \leq \lambda_i(A) + \lambda_{C-1}(B).$$

A.3. Remarks

Theorem A.4 serves as a criterion and characterisation of differentiable convex functions through their gradients. Theorems A.6 to A.8 are hallmark results from the rich literature of convex analysis and monotone operators that tie together conditions under which a differentiable composite function is the gradient of a convex function. Notably, Theorem A.7 is a rewritten
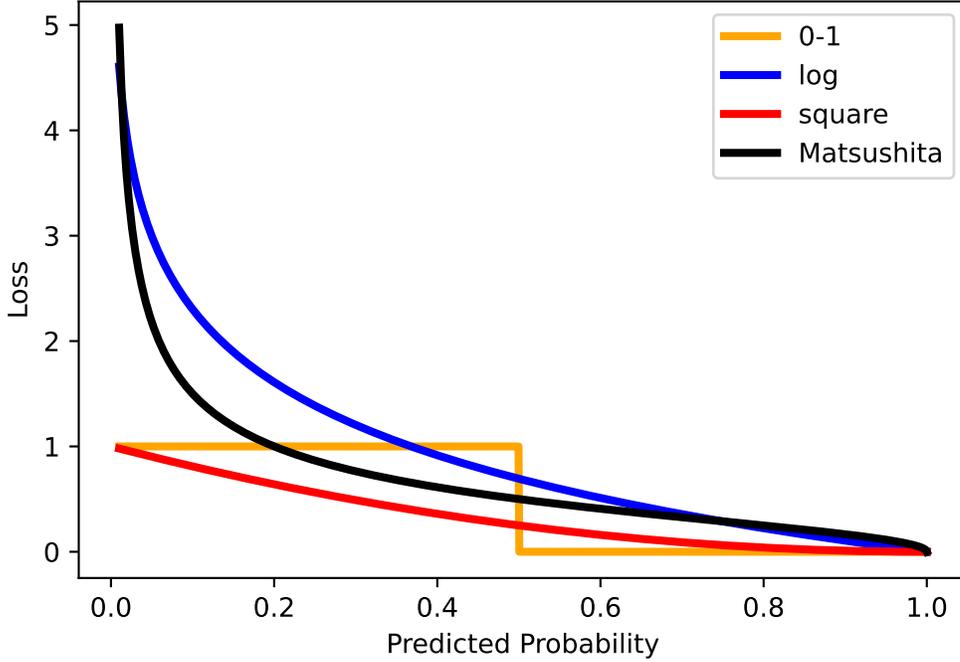
Figure 3. Plots of partial losses corresponding various proper losses when true probability is 1.

version of the Asplund decomposition of maximal monotone operators (Asplund, 1968) which tells us it suffices to focus on maximal monotonicity. We refer the reader to Appendix E for the usage of Theorems A.5 to A.8 in the proof of Theorem 4.2.

Theorem A.9 allows us to obtain a lower bound on the smallest eigenvalue of the sum of two real-valued matrices with real eigenvalues. This is particularly useful to prove positive definiteness in Theorem 4.3. We refer the reader to Appendix F for its usage in the proof of Theorem 4.3.

B. Examples of Binary Proper Losses

Denote $y \in \{-1, 1\}$ a label and $p = \Pr(Y = 1|\mathbf{x})$ be the true probability that $Y = 1|\mathbf{x}$. Let $\hat{y}$ and $\hat{p}$ be the predicted class and probability estimate of $Y = 1$ given input $\mathbf{x}$. Below are the analytical formulas of partial losses for various examples of binary proper losses where it is assumed that $p = 1$. Figure 3 shows the plots of the below partial losses.

	$\ell_1(\hat{p})$
$0 - 1$	$[\![\hat{y}(\hat{p}) = 1]\!]$
square	$(1 - \hat{p})^2$
log	$-\log(\hat{p})$
Matsushita	$\frac{1}{2}\sqrt{\frac{1-\hat{p}}{\hat{p}}}$

C. Proof of equivalent conditions on subdifferentials for strictly convex functions

($\Rightarrow$) Suppose f is strictly convex and assume for a proof by contradiction that there exists some $\mathbf{x}, \mathbf{y} \in \mathrm{dom} f$ such that $\mathbf{x} \neq \mathbf{y}$ with $f(\mathbf{x}) + \langle \phi, \mathbf{y} - \mathbf{x} \rangle \geq f(\mathbf{y})$ for some $\phi \in \partial f(\mathbf{x})$.

Fix $\lambda \in (0, 1)$. Then we have

$$
\begin{aligned}
f(\mathbf{x}) + \langle \phi, (\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) - \mathbf{x} \rangle &= f(\mathbf{x}) + (1 - \lambda)\langle \phi, \mathbf{y} - \mathbf{x} \rangle \\
&\leq f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \text{ by definition of a subgradient} \\
&< \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}) \text{ by strict convexity of } f \\
&\leq f(\mathbf{x}) + (1 - \lambda)\langle \phi, \mathbf{y} - \mathbf{x} \rangle \text{ by the above assumption.}
\end{aligned}
$$

Thus, we have a contradiction so we must have the subdifferential of f for all $\mathbf{x} \in \text{dom} f$ is given by

$$
\partial f(\mathbf{x}) = \{\phi \in \mathbb{R}^n : \langle \phi, \mathbf{y} - \mathbf{x} \rangle < f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \in \mathbb{R}^n\}.
$$

($\Leftarrow$) Suppose the subdifferential of f for any $\mathbf{x} \in \text{dom} f$ is given by

$$
\partial f(\mathbf{x}) = \{\phi \in \mathbb{R}^n : \langle \phi, \mathbf{y} - \mathbf{x} \rangle < f(\mathbf{y}) - f(\mathbf{x}), \forall \mathbf{y} \in \mathbb{R}^n\}.
$$

Fix $\mathbf{x}, \mathbf{y} \in \text{dom} f$ and $\lambda \in (0, 1)$. Consider $\phi \in \partial f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y})$. Then we have

$$
\begin{aligned}
f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) + (1 - \lambda)\langle \phi, \mathbf{x} - \mathbf{y} \rangle &< f(\mathbf{x}), \\
f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) + \lambda \langle \phi, \mathbf{y} - \mathbf{x} \rangle &< f(\mathbf{y}).
\end{aligned}
$$

Multiplying the first inequality by λ and the second by $(1 - \lambda)$, summing them gives us $f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$. This holds for arbitrary $\mathbf{x}, \mathbf{y} \in \text{dom} f$ and $\lambda \in (0, 1)$ so it follows that f is strictly convex.

D. Proof of Proposition 3.3

($\Rightarrow$) Fix $q \in \Delta^{C-1}$. Suppose ℓ is proper. Then we have

$$
L(p, q) = p^\top \ell(q) = q^\top \ell(q) + (p - q)^\top \ell(q) = \underline{L}(q) + (p - q)^\top \ell(q)
$$

and also,

$$
\begin{aligned}
0 \leq L(p, q) - L(p, p) &= \underline{L}(q) + (p - q)^\top \ell(q) - \underline{L}(p) \\
\implies -(p - q)^\top \ell(q) &\leq -\underline{L}(p) - (-\underline{L}(q)).
\end{aligned}
$$

Recall that $\underline{L}$ is concave so it follows that $-\underline{L}$ is convex. Hence, $-\ell(q) \in \partial(-\underline{L})(q)$ which means $-\ell(q)$ is a subgradient of $-\underline{L}$ at q and $L(p, q) = -(-\underline{L}(q)) - (p - q)^\top(-\ell(q))$.

($\Leftarrow$) Suppose there exists a convex function $f : \Delta^{C-1} \to \mathbb{R}$ such that for all $q \in \Delta^{C-1}$, there exists a subgradient $\phi \in \partial f(q)$ and $L(p, q) = -f(q) - (p - q)^\top \phi$.

For all $p \in \Delta^{C-1}$, we have

$$
\begin{aligned}
L(p, q) - L(p, p) &= f(p) - f(q) - (p - q)^\top \phi \\
&\geq 0 \text{ since } \phi \text{ is a subgradient of } f \text{ at } q \\
\implies L(p, p) &\leq L(p, q).
\end{aligned}
$$

Hence, ℓ is a proper loss.

To prove that ℓ is strictly proper if and only if there exists a strictly convex function $f : \Delta^{C-1} \to \mathbb{R}$ such that for all $q \in \Delta^{C-1}$, there exists a subgradient $\phi \in \partial f(q)$ such that $L(p, q) = -(p - q)^\top \phi - f(q)$ for all $p \in \Delta^{C-1}$. This follows immediately by definitions of strictly proper losses and subdifferentials from which the above inequalities become strict.

We are left to prove that $L(p, q) = (p - q)^\top \ell(q) + \underline{L}(q)$ when $\underline{L}$ is differentiable. We first note that $\underline{L}$ is concave so $-\underline{L}$ is convex. Recall that $L(p, q) = p^\top \ell(q) = \underline{L}(q) + (p - q)^\top \ell(q)$ from the workings within Appendix D. Setting $f = -\underline{L}$ for Proposition 3.3, we can deduce $-\ell(q) = -\nabla \underline{L}(q) \forall q \in \text{ri}(\Delta^{C-1})$ for a proper loss ℓ which follows by the uniqueness of subgradients for differentiable functions. That is, $\nabla \underline{L}(q) = \ell(q), \forall q \in \text{ri}(\Delta^{C-1})$.

E. Proof of Theorem 4.2

(1) $\implies$ (2). This follows from Schwarz's theorem on the equality of mixed partial derivatives. (2) $\implies$ (3). This follows from Theorem A.8 since the Jacobian of a composite function is a product of matrices. (3) $\implies$ (4). This follows from Theorem A.5. We are left to prove (4) $\implies$ (1).

(4) $\implies$ (1). Since $f \circ g$ is monotone and continuous, it follows from Theorem A.6 that $f \circ g$ is maximally monotone. From Theorem A.7, $(f \circ g)(\mathbf{x}) = \nabla F(\mathbf{x}) + L\mathbf{x}$ for a differentiable convex function F and a skew-symmetric matrix L. Since $f \circ g$ is differentiable then it follows that F is twice-differentiable. This gives us $J_{f \circ g} = \nabla^2 F + L^\top$ where $\nabla^2 F$ is symmetric by Schwarz's theorem on the equality of mixed partial derivatives. Theorems A.5 and A.8 tell us that $J_{f \circ g}$ is also symmetric. As $J_{f \circ g}$ and $\nabla^2 F$ are both symmetric, then we must have $L^\top = 0 = L$. That is, $f \circ g = \nabla F$ where F is twice-differentiable and convex.

F. Proof of Theorem 4.3

Fix $\mathbf{x} \in \mathbb{R}^{C-1}$. The Jacobian of $f \circ g$ is given by

$$J_{f \circ g}(\mathbf{x}) = J_f(g(\mathbf{x}))J_g(\mathbf{x}).$$

Here we aim to prove that $J_{f \circ g}(\mathbf{x})$ is positive definite. We first note that $J_g(\mathbf{x})$ is invertible since $|J_g(\mathbf{x})| > 0$. Now, note that $J_{f \circ g}(\mathbf{x})$ is similar to the matrix

$$(J_g(\mathbf{x}))^{\frac{1}{2}} J_f(g(\mathbf{x})) J_g(\mathbf{x}) (J_g(\mathbf{x}))^{-\frac{1}{2}} = (J_g(\mathbf{x}))^{\frac{1}{2}} J_f(g(\mathbf{x})) (J_g(\mathbf{x}))^{\frac{1}{2}}$$

where the square root of the matrix $J_g(\mathbf{x})$ is given by $(J_g(\mathbf{x}))^{\frac{1}{2}}$ which is known to be symmetric since $J_g(\mathbf{x})$ is symmetric and positive definite. Since $J_f(g(\mathbf{x}))$ is also symmetric, it follows that $(J_g(\mathbf{x}))^{\frac{1}{2}} J_f(g(\mathbf{x})) (J_g(\mathbf{x}))^{\frac{1}{2}}$ is symmetric. As $J_g(\mathbf{x})$ and $J_f(g(\mathbf{x}))$ are positive definite, we have

$$\left| (J_g(\mathbf{x}))^{\frac{1}{2}} J_f(g(\mathbf{x})) (J_g(\mathbf{x}))^{\frac{1}{2}} \right| = |J_f(g(\mathbf{x}))| \, |(J_g(\mathbf{x}))| > 0.$$

It follows that $(J_g(\mathbf{x}))^{\frac{1}{2}} J_f(g(\mathbf{x})) (J_g(\mathbf{x}))^{\frac{1}{2}}$ is positive definite, meaning it has positive eigenvalues $\lambda_1, \ldots, \lambda_{C-1} \in \mathbb{R}$ that can be denoted such that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{C-1}$. Since similar matrices have the same eigenvalues, it follows that $\lambda_1, \ldots, \lambda_{C-1}$ are also the eigenvalues of $J_{f \circ g}(\mathbf{x})$.

Denote $S = \frac{1}{2}(J_{f \circ g}(\mathbf{x}) + (J_{f \circ g}(\mathbf{x}))^\top)$ and $A = \frac{1}{2}(J_{f \circ g}(\mathbf{x}) - (J_{f \circ g}(\mathbf{x}))^\top)$ as the symmetric and skew-symmetric parts of $J_{f \circ g}(\mathbf{x})$ respectively. It is well known that for any skew-symmetric matrix A and any $\mathbf{z} \in \mathbb{R}^{C-1}$, we have $\mathbf{z}^\top A \mathbf{z} = 0$. To prove that $J_{f \circ g}(\mathbf{x})$ is positive definite, it suffices to prove that $\mathbf{z}^\top J_{f \circ g}(\mathbf{x}) \mathbf{z} = \mathbf{z}^\top S \mathbf{z} > 0$ for any $\mathbf{z} \in \mathbb{R}^{C-1} \setminus \{0\}$.

Firstly, recall that all eigenvalues of $J_{f \circ g}(\mathbf{x})$ are real and positive, and the fact that the transpose of $J_{f \circ g}(\mathbf{x})$, $(J_{f \circ g}(\mathbf{x}))^\top$, has the same eigenvalues as $J_{f \circ g}(\mathbf{x})$. That is, all eigenvalues of $(J_{f \circ g}(\mathbf{x}))^\top$ are real and positive. Secondly, $S = \frac{1}{2}(J_{f \circ g}(\mathbf{x}) + (J_{f \circ g}(\mathbf{x}))^\top)$ is symmetric so all of its eigenvalues must be real. Hence, Theorem A.9 gives us the following bound for the smallest eigenvalue $\lambda_1(S)$

$$\lambda_1(S) \geq \lambda_1\left(\frac{1}{2}J_{f \circ g}(\mathbf{x})\right) + \lambda_1\left(\frac{1}{2}(J_{f \circ g}(\mathbf{x}))^\top\right)$$
$$= \frac{1}{2}\left(\lambda_1(J_{f \circ g}(\mathbf{x})) + \lambda_1((J_{f \circ g}(\mathbf{x}))^\top)\right)$$
$$> 0.$$

The Rayleigh quotient for S and any $\mathbf{z} \in \mathbb{R}^{C-1} \setminus \{0\}$, is given by $\frac{\mathbf{z}^\top S \mathbf{z}}{\|\mathbf{z}\|^2}$, and satisfies the inequality

$$\lambda_1(S) \leq \frac{\mathbf{z}^\top S \mathbf{z}}{\|\mathbf{z}\|^2} \leq \lambda_{C-1}(S).$$

Hence, we have

$$\frac{\mathbf{z}^\top S \mathbf{z}}{\|\mathbf{z}\|^2} \geq \lambda_1(S) > 0 \text{ for all } \mathbf{z} \in \mathbb{R}^{C-1} \setminus \{0\}.$$

Thus, $\mathbf{z}^\top J_{f \circ g}(\mathbf{x})\mathbf{z} = \mathbf{z}^\top S\mathbf{z} > 0, \forall \mathbf{z} \in \mathbb{R}^{C-1} \setminus \{0\}$ and so, $J_{f \circ g}(\mathbf{x})$ is positive definite. This holds for arbitrary $\mathbf{x} \in \mathbb{R}^{C-1}$ so it follows that $f \circ g$ is the gradient of a twice-differentiable convex function F by Theorem 4.2 with F being strictly convex since $J_{f \circ g}(\mathbf{x})$ is positive definite. In other words, $f \circ g$ is the gradient of a twice-differentiable Legendre function.

G. Proof of Theorem 5.1

Proof of Properties of LogSumExp$^+$ and softmax$^+$ Since positive definiteness of $J_u(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^{C-1}$ implies strict convexity of f and strict convexity of f implies invertibility of u, it suffices to prove that $J_u(\mathbf{x})$ is positive definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$.

Fix $\mathbf{x} \in \mathbb{R}^{C-1}$. For ease of notation, we denote M as $J_u(\mathbf{x})$ where M_{ij} refers to the entry within the i-th row and j-th column of $J_u(\mathbf{x})$. Consider any row $i \in \{1, \ldots, C-1\}$. We have

$$
\begin{aligned}
M_{ii} &= \frac{\exp(x_i)}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \left(1 - \frac{\exp(x_i)}{1 + \sum_{k=1}^{C-1} \exp(x_k)}\right), \\
M_{ij} &= -\frac{\exp(x_i)}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \frac{\exp(x_j)}{1 + \sum_{k=1}^{C-1} \exp(x_k)}.
\end{aligned}
$$

Observe that $M_{ii} - \sum_{j \neq i} |M_{ij}| = \frac{\exp(x_i)}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \left(1 - \frac{\sum_{k=1}^{C-1} \exp(x_k)}{1 + \sum_{k=1}^{C-1} \exp(x_k)}\right) > 0$. This holds for arbitrary $i \in \{1, \ldots, C-1\}$ so it follows that $J_u(\mathbf{x})$ is strictly diagonally dominant. This implies that $J_u(\mathbf{x})$ is positive definite so it follows that f is strictly convex. This completes the proof of the key properties of the LogSumExp$^+$ function and its gradient softmax$^+$.

Proof of functions learned by LEGENDRETRON are inverse canonical links   We first note that $v^{-1} = (\nabla g_1) \circ (\nabla g_2) \circ \cdots \circ (\nabla g_B)$ is indeed invertible since the RHS is invertible by the strong convexity of $g_1, g_2, \ldots, g_B$. Since each ∇g_i is symmetric and positive definite, it follows that v^{-1} is the gradient of a twice-differentiable Legendre function by applying Theorem 4.3 recursively. It follows from Theorem 4.2 that $J_{v^{-1}}(\mathbf{x})$ is symmetric. We also have that $|J_{v^{-1}}(\mathbf{x})| > 0$ so $J_{v^{-1}}(\mathbf{x})$ is positive definite for all $\mathbf{x} \in \mathbb{R}^{C-1}$.

Recall that LogSumExp$^+$ is twice-differentiable with gradient $u = $ softmax$^+$ and Hessian $J_u(\mathbf{x})$ being strictly diagonally dominant. That is, $J_u(\mathbf{x})$ is symmetric and positive definite. Applying Theorem 4.3 on $u \circ v^{-1}$ allows us to deduce that $u \circ v^{-1}$ is the gradient of a twice-differentiable Legendre function that maps to $\tilde{\Delta}^{C-1}$ so $u \circ v^{-1}$ can be set as the inverse of an implicit canonical link function.

H. Proof of Corollary 5.2

Let $u = $ softmax$^+$ and fix g to be the gradient of a twice-differentiable Legendre function with positive Hessian everywhere. Note that LogSumExp$^+$ is twice-differentiable and Legendre so u^{-1} and g satisfy the sufficient conditions of Theorem 4.3. It follows that $u^{-1} \circ g$ is the gradient of a twice-differentiable Legendre function defined on a compact set Ω. The result then follows from using Proposition 3 of Huang et al. (2021).

I. Experimental Details

I.1. Network Architecture and Optimisation Details

Experiment details on architecture and optimisation parameters for LEGENDRETRON (LT) and multinomial logistic regression (MLR). Here we denote α as the learning rate, λ as weight decay, γ as the multiplicative rate of decay applied to α every S epochs through a step-wise learning rate scheduler. We used the Adam optimiser for all experiments.

Dataset(s)	Model	B	H	M	α	γ	S	Epochs	Batch Size
MNIST/FMNIST/KMNIST	LT	1	4	4	0.001	0.7	4	200	128
MNIST/FMNIST/KMNIST	MLR	\	\	\	0.001	0.7	4	200	128
aloi	LT	2	2	4	0.01	0.95	4	360	64
aloi	MLR	\	\	\	0.01	0.95	4	360	64
LIBSVM/UCI/Statlog (other datasets)	LT	2	2	4	0.01	0.95	4	240	64
LIBSVM/UCI/Statlog (other datasets)	MLR	\	\	\	0.01	0.95	4	240	64

I.2. LogSumExp trick for softmax$^+$

Let $u = \mathrm{softmax}^+$ and consider $\mathbf{x} \in \mathbb{R}^{C-1}$. We have

$$\log(\Pi^{-1}(u(\mathbf{x}))) = \left(\log \left(\frac{\exp(x_1)}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \right), \ldots, \log \left(\frac{\exp(x_{C-1})}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \right), \log \left(\frac{1}{1 + \sum_{k=1}^{C-1} \exp(x_k)} \right) \right)^\top$$

where $\log$ on the LHS is applied elementwise. We seek an alternate expression for $\log(\Pi^{-1}(u(\mathbf{x})))$ that is numerically stable.

Let $x^* = \max(x_1, \ldots, x_{C-1})$ and $S = \exp(-x^*) + \sum_{k=1}^{C-1} \exp(x_k - x^*)$. We can write

$$\log(\Pi^{-1}(u(\mathbf{x}))) = \left(\log \left(\frac{\exp(x_1 - x^*)}{S} \right), \ldots, \log \left(\frac{\exp(x_{C-1} - x^*)}{S} \right), \log \left(\frac{\exp(-x^*)}{S} \right) \right)^\top$$
$$= (x_1 - x^* - \log(S), \ldots, x_{C-1} - x^* - \log(S), -x^* - \log(S))^\top .$$

It can be observed that this expression is numerically stable for all large values of x^*.