Information Geometry (2025) 8:125–157
https://doi.org/10.1007/s41884-025-00165-4

RESEARCH PAPER

Dual stochastic natural gradient descent

and convergence of interior half-space gradient approximations

Borja Sánchez-López[1] · **Jesus Cerquides**[1]

Received: 1 April 2020 / Revised: 13 December 2024 / Accepted: 2 March 2025 /
Published online: 4 April 2025

Abstract

The multinomial logistic regression (MLR) model is widely used in statistics and machine learning. On the one hand, stochastic gradient descent (SGD) is the most common approach for determining the parameters of a such model in big data scenarios, due to its simplicity and low computational complexity property. Furthermore, SGD has proven convergence under reasonable conditions. However, SGD has slow sublinear rates of convergence and it often reduces convergence speed due to the plateau phenomenon. On the other hand, stochastic natural gradient descent (SNGD), proposed by Amari, is a manifold optimization method shown to be Fisher efficient when it converges, but its convergence properties remain unproven and it is often computationally prohibitive for models with a large number of parameters. Here, we propose dual stochastic natural gradient descent (DSNGD), a stochastic optimization method for MLR based on manifold optimization concepts. In the discrete scenario, DSNGD (i) has linear per-iteration computational complexity in the number of parameters, and (ii) is proven to converge. To achieve (i) we leverage the dual flatness of the family of joint distributions for MLR to simplify computations. To ensure (ii) DSNGD builds on the foundational ideas of convergent stochastic natural gradient descent (CSNGD), a variant of SNGD with guaranteed convergence, using an independent sequence to construct a bounded approximation of the natural gradient. By generalizing a result from Sunehag et al., we prove that DSNGD converges in the discrete case and maintains linear computational complexity per iteration. Beyond its convergence property and linear computational complexity, DSNGD empirically demonstrates fast convergence comparable to SNGD, improves upon SGD performance, and exhibits stability where SNGD does not.

Communicated by Nihat Ay.

✉ Borja Sánchez-López
bsanchlo9@gmail.com

Jesus Cerquides
cerquide@iiia.csic.es

[1] IIIA-CSIC, Campus UAB, 08193 Cerdanyola, Spain

Keywords Multinomial logistic regression · Stochastic gradient descent · Natural gradient · Convergence · Riemannian manifold · Computational complexity

1 Introduction

Multinomial Logistic Regression (MLR) is a widely used tool for classification. Its core assumption is that the log-odds ratio of the class posteriors is an affine function of the features [1]. Relevant examples include applications in image classification [2], video recommendation systems [3], and various health and life sciences scenarios analyzing nominal qualitative response variables [4–7].

MLR's theoretical underpinnings extend beyond practical use. In statistical decision theory, the MLR model can be derived under the assumptions that (i) random utilities are independently and identically distributed (i.i.d.) across alternatives and (ii) their common distribution is Gumbel [8]. Recent studies [9] demonstrate that the Gumbel distribution is not strictly necessary; any distribution asymptotically exponential in its tail suffices.

Cross-entropy, or log-loss, is the loss function most commonly employed in MLR, with solid justification from statistical decision theory. Once the form of the loss function is elicited [10, 11], and the inverse link function is understood as mapping scores to class probabilities, log-loss emerges as a proper composite loss for logistic regression [11–13]. This supports its adoption for classification from a theoretical standpoint.

Classification algorithms aim to predict a discrete variable (class) given feature variables. Denote the class variable by $\mathscr{Y}$ and the features by $\mathscr{X}$, with a finite set of classes $\mathscr{Y} \in \{1, \ldots, s\}$. The objective is to compute the unknown conditional probability distributions $P(\mathscr{Y} \mid \mathscr{X})$ by minimizing the expected risk function [14]. Stochastic gradient descent (SGD) is the most prevalent optimization method for this task due to its simplicity and speed. The strategy of SGD is very intuitive: Assume $\overline{P}$ is an unknown probability distribution on $\mathscr{X} \times \mathscr{Y}$, $\mathbb{L}$ is a differentiable function from $\mathbb{R}^k$ to $\mathbb{R}$, and l is a differentiable loss function such that $\mathbb{L}(\eta) = \mathbb{E}_{z \sim \overline{P}}[l(\eta, z)]$ where $z \in \mathscr{Y} \times \mathscr{X}$. In SGD, the update rule for parameters η is given by:

$$\eta_{t+1} \leftarrow \eta_t - \gamma_t \nabla l(\eta_t, s_t) \tag{1}$$

where $\nabla l(\eta_t, s_t) = \frac{1}{|s_t|} \sum_{z \in s_t} \nabla l(\eta_t, z)$ is an approximation to the gradient of $\mathbb{L}(\eta_t)$ according to a sample set $s_t \sim \overline{P}$, and γ_t is a positive number encoding the learning rate. Assuming certain regularities on the function and the learning rate, SGD converges to the minimum [15]. However, usually convergence speed suddenly drops after a moderate solution quality has been reached, making the minimum unreachable from a practical point of view [16]. Furthermore, SGD highly depends on the learning rate parameter, which can be difficult to tune, and it is vulnerable to the plateau phenomenon and to ill-conditioning. To face this issue, many SGD variants have arisen that basically modify the direction $\nabla l(\eta_t, s_t)$ using a positive semi-definite matrix M_t. Specifically,

such generalization of SGD can be described by equation

$$\eta_{t+1} \leftarrow \eta_t - \gamma_t M_t \nabla l(\eta_t, s_t) \tag{2}$$

where M_t is commonly a preconditioning matrix capturing the local curvature or related information such as the inverse of Hessian matrix in Newton's method [17] or the inverse of the Fisher Information Matrix in Stochastic Natural Gradient Descent (SNGD) [18]. Due to the increment of the computational complexity (Eq. (2) defines a second order method) a trade-off between quality of curvature information and computational cost is assumed. Some widely used examples are preconditioned SGD, diagonal approximations of the Hessian [19]; Adagrad [20], Adadelta [21], RMSProp [22] or Adam [23] that use the diagonal of the covariance matrix of the gradients.

Among existing preconditioning matrix algorithms, we focus our attention to SNGD and its variants in manifold [24–26]. This kind of algorithm runs over a smooth manifold $\mathscr{M}$ of dimension n [27] equipped with a metric g defined at every $p \in \mathscr{M}$. Metric g_p is the positive-definite tensor that expresses the local metric information at p. The pair $(\mathscr{M}, g)$ is a Riemannian manifold of dimension n [27]. It is possible to choose a system of coordinates $\eta \in U \subset \mathbb{R}^n$—or parametrization—to refer to points in $\mathscr{M}$. In this case the metric information of g at η is given by an n dimensional square matrix G_η, symmetric and positive-definite, in the base derived by the parametrization.

Assume that $\mathscr{M}$ is not the standard $\mathbb{R}^n$; for example $\mathscr{M}$ could be a sphere of dimension n or a Statistical manifold [28]. A statistical manifold is a manifold whose points represent probability distributions belonging to the same family, where the Fisher information metric (FIM) is commonly used. In such cases, working in a Riemannian manifold offers two key advantages. First,it provides a notion of angles and local lengths that account for the space's specific curvature, enabling more effective updates. This is particularly important in optimization algorithms that rely on gradients, as gradients are defined in terms of these local magnitudes. Second, it allows for the correct definition of directions within the space, addressing a limitation of stochastic gradient descent (SGD). In SGD, the descent direction is dependent on the parameterization; that is, the gradient varies with the chosen parameterization.

Roughly speaking, the gradient of a function f at $p \in \mathscr{M}$ represents the steepest direction of f at p. In a Riemannian manifold this is referred to as the natural gradient, as defined by [18], and is denoted as $\widetilde{\nabla} f(p)$. It is well-defined because it incorporates the metric of the manifold. If a parametrization η is fixed, Amari [18] proved that

$$\widetilde{\nabla} f(\eta) = (G_\eta)^{-1} \nabla f(\eta) \tag{3}$$

where $\widetilde{\nabla} f(\eta)$ is the natural gradient at η in the base derived from the parametrization. Furthermore, Amari shows that SNGD is Fisher efficient assuming convergence, which guarantees optimal performance by achieving the Cramer–Rao lower bound.

SNGD follows the natural gradient instead of the standard gradient. Specifically, it sets the preconditioning matrix $M_t = (G_{\eta_t})^{-1}$ in update Eq. (2), to align with the

natural gradient shown in Eq. (3)

$$\begin{aligned}\eta_{t+1} &\leftarrow \eta_t - \gamma_t \widetilde{\nabla} l(\eta_t, z_t) \\ &\leftarrow \eta_t - \gamma_t (G_{\eta_t})^{-1} \nabla l(\eta_t, z_t)\end{aligned} \tag{4}$$

This algorithm, or its approximation, often accelerates learning in many problems, helps avoid the plateau effect and it defines parametrization-independent directions [25]. However, it faces two main problems:

(i) High computational complexity due to the need to either inverting a matrix or solving a linear system, and
(ii) Lack of convergence guarantee or it needs stronger assumptions such as compactness to stabilize [29].

Issue (i) concerns the high computational complexity, which becomes a significant challenge for large-scale high-dimensional problems. Issue (ii) pertains to the lack of theoretical convergence and divergence proofs of SNGD. Some isolated experiments show instability of SNGD, such as in [26, 30]. Even in the die problem, a toy example discussed in [26], SNGD exhibits an erratic behaviour in low entropy scenarios, while the convergent natural gradient variant CSNGD stabilizes in all scenarios.

This paper presents a novel natural gradient optimization method [24] for MLR, named dual stochastic natural gradient descent (DSNGD). When $\mathscr{X}$ is discrete, DSNGD has:

(i) Linear computational complexity, comparable to the one of SGD, and
(ii) Theoretical convergence guarantee

Therefore the contribution of this paper is DSNGD, a natural gradient method that approximates SNGD without suffering issues (i) and (ii). To accomplish (i) we establish that the family of joint distributions for MLR is a dually flat manifold and we use that to speed up calculations. To reach (ii) our algorithm uses the fundamental idea from CSNGD [26], relying on an independent sequence to build a bounded approximation of the natural gradient.

Section 2 reviews related work and introduces essential concepts required to address issues (i) and (ii) for our natural gradient based algorithm. Section 3 defines DSNGD, while Sects. 4 and 5 tackle issues (i) and (ii), respectively, in the discrete case ($\mathscr{X} = \{1, \dots m\}$). These sections demonstrate that discrete DSNGD is both convergent and as efficient as SGD in terms of computational complexity. Finally, Sect. 6 presents empirical evidence showing that DSNGD outperforms SGD in convergence speed and exhibits greater stability than SNGD.

2 Related work

2.1 Dually flat manifolds

Issue (i) is addressed in this paper by restricting attention to dually flat manifolds (DFM) [31, 32]. The computational cost of natural gradient can be significantly reduced if the ambient space is a DFM, as in methods like mirror descent [33].

DFM are built using two dual connections—conjugate connections—that are flat, meaning that the Riemann–Christoffel curvature vanishes. As shown in [31], in such manifolds, there exist two dual parametrizations η and η^*, related by the Legendre transform of a convex function $F(\eta)$, such that

$$\begin{aligned} \eta^* &= \nabla F(\eta) \\ \nabla^2 F(\eta) &= G_\eta. \end{aligned} \tag{5}$$

Here, η and η^* refer to the same point. This leads to a key property of DFM. By applying the chain rule to Eq. (3), we obtain

$$\begin{aligned} \widetilde{\nabla} f(\eta) &= (G_\eta)^{-1} \nabla f(\eta) \\ &= (G_\eta)^{-1} \nabla \eta^*(\eta) \nabla f(\eta^*) \\ &= (G_\eta)^{-1} \nabla \nabla F(\eta) \nabla f(\eta^*) \\ &= (G_\eta)^{-1} G_\eta \nabla f(\eta^*) \\ &= \nabla f(\eta^*) \end{aligned} \tag{6}$$

for any differentiable function f defined on $\mathscr{M}$. As a notational convention, the ∇ operator is always taken with respect to the coordinates indicated by the function.

Equation (6) has been proved for linear exponential families, a well known DFM, in [34] and [35]. Thus, Eq. (6) shows that the natural gradient is equivalent to the gradient in its dual parametrization. From this, we can deduce a strategy to compute the natural gradient—or an approximation thereof—without the need for matrix inversion or solving a linear system.

In summary, the key idea that resolves issue (i) is Eq. (6). For instance, in the case of SNGD in a DFM, we can equivalently express SNGD as

$$\eta_{t+1} \leftarrow \eta_t - \gamma_t \nabla l(\eta_t^*, z_t), \tag{7}$$

which avoids the matrix inversion and linear system solving required in Eq. (4)

2.2 Mirror descent

Many algorithms benefit from the properties of dual spaces. The mirror descent algorithm [33] leverages dual parametrizations. As shown in [35], mirror descent in a dually flat manifold is essentially SNGD applied in the dual space.

According to [36], mirror descent follows the update rule

$$\begin{aligned} \eta_t &\leftarrow \nabla F^*(\eta_t^*) \\ \eta_{t+1}^* &\leftarrow \nabla F(\eta_t) - \gamma_t \nabla l(\eta_t, s_t) \end{aligned} \tag{8}$$

where F is a convex function and F^* is the Legendre transform of F.

Another algorithm making use of duality derived from exponential families can be found for instance in Bayesian Inference field named CVI [37]. For this method,

authors explain that parameters involved in the conjugate terms are updated using mirror descent, while the other parameters make stochastic gradient steps. In contrast to mirror descent, DSNGD utilizes both dual parameterizations, but it does not switch between spaces. Instead, it follows two separate sequences, one for each space, similarly to how CSNGD operates.

2.3 Multinomial logistic regression

As described above, our strategy to reduce the per-iteration computational complexity relies on the fact that the family of joint distributions for MLR forms a dually flat manifold.

The main assumption of MLR [1] is that the log-odds ratio of the class posteriors $P(\mathscr{Y} \mid \mathscr{X})$ is an affine function of the features $\mathscr{X}$. More precisely, this assumption states that for any two classes y_1, y_2, the ratio

$$\log \frac{P(y_1 \mid \mathscr{X})}{P(y_2 \mid \mathscr{X})} \tag{9}$$

is affine in $\mathscr{X}$. For further details on this assumption, refer to Banerjee [1]. Banerjee [1] proved (Theorem 2) that a class of distributions fulfills that assumption if and only if for each value of $\mathscr{Y}$, the class of conditional distributions $P(\mathscr{X} \mid \mathscr{Y})$ belongs to the same linear exponential family (LEF).[1] In Sect. 3.1 we use these results to prove that the class of joint distributions $P(\mathscr{Y}, \mathscr{X})$ also belongs to a LEF. It is well known that a LEF is a DFM [31]. Typically, finding the minimum expected risk MLR parameters is formulated as an optimization problem in $\mathbb{R}^k$ which is solved using SGD. Instead, we propose to formulate the problem as a manifold optimization problem [24], over the manifold of probability distributions $P(\mathscr{Y}, \mathscr{X})$ that satisfy the main assumption of MLR. The joint distribution is also considered in [39], for example, where dual parameterizations enable the use of a fast SNGD. Since we will demonstrate that this manifold is dually flat, this formulation of the problem will allow us to efficiently capture the curvature information of the manifold.

2.4 Convergent stochastic natural gradient descent

In [26], an convergent variant for SNGD, namely CSNGD, is presented. CSNGD becomes stable in every toy scenario presented, unlike SNGD which fails in those same situations. Moreover, from a practical point of view it inherits the convergence speed of SNGD. CSNGD is defined with the update rule

$$\eta_{t+1} \leftarrow \eta_t - \gamma_t (G_{\zeta_t})^{-1} \nabla l(\eta_t, z_t) \tag{10}$$

where $\{\zeta_t\}_{t \in \mathbb{N}}$ can be any convergent sequence in $\mathbb{R}^k$. Both SNGD and CSNGD work by progressively building a sequence $\{\eta_t\}_{t \in \mathbb{N}}$. However, CSNGD additionally maintains an independent sequence ζ_t, which is only required to be convergent. This difference

[1] For a definition of linear exponential family see [38]

allows CSNGD to converge to the unique minimum of a convex function after some reasonable conditions on the learning rate parameter. Precisely, SNGD does not converge due to the inverse matrix $(G_{\eta_t})^{-1}$ in Eq. (4), since eigenvalues of that matrix are unbounded. CSNGD forces convergence and eigenvalue confinement of sequence $\{(G_{\zeta_t})^{-1}\}_{t\in\mathbb{N}}$ because of the convergence of sequence $\{\zeta_t\}_{t\in\mathbb{N}}$ and continuity property, stabilizing the algorithm. This idea is used in Sect. 3 which allows us to prove convergence of our new algorithm in Sect. 5.

2.5 Convergence of interior half-space gradient approximations

In [40] Sunehag et al. provide a variable metric stochastic approximation theory. One of the key results in that paper is Theorem 3.2 which proves convergence given that we take a scaling matrix B_t at step t of the algorithm, provided that the spectrum of their (possibly non-convergent) scaling matrices is uniformly bounded from above by a finite constant and from below by a strictly positive constant. Moreover it assumes the step direction at iteration t is some Y_t modified by the scaling matrix. Vector Y_t is drawn from a family of random variables Y defined for all η, and $Y_t = Y(\eta_t)$. The result is stated here.

Theorem 1 (Theorem 3.2 in [40]) *Let $\mathbb{L} : \mathbb{R}^k \to \mathbb{R}$ be a twice differentiable function with a unique minimum $\overline{\eta}$ and $\eta_{t+1} = \eta_t - \gamma_t B_t Y_t$ where B_t is symmetric and only depends on information available at time t. Then η_t converges to $\overline{\eta}$ almost surely if the following conditions hold*

C.1 $(\forall t)\ \mathbb{E}_t Y_t = \nabla\mathbb{L}(\eta_t)$

C.2 $(\exists K)(\forall \eta)\ \left\|\nabla^2_\eta \mathbb{L}(\eta)\right\| \leq 2K$

C.3 $(\forall \delta > 0)\ \inf_{\mathbb{L}(\eta)-\mathbb{L}(\overline{\eta})>\delta} \|\nabla\mathbb{L}(\eta)\| > 0$

C.4 $(\exists A, B)(\forall t)\ \mathbb{E}_t \|Y_t\|^2 \leq A + B \cdot \mathbb{L}(\eta_t)$

C.5 $(\exists m, M : 0 < m < M < \infty)\ (\forall t) mI \prec B_t \prec MI$, *where I is the identity matrix;*

C.6 $\sum_t \gamma_t^2 < \infty,\ \sum_t \gamma_t = \infty$

$\mathbb{E}_t$ in conditions **C.1** and **C.4** notes the conditional expectation given observations until time t. That is, $\mathbb{E}_t X = \mathbb{E}[X \mid \mathscr{F}_t]$, where $\mathscr{F}_t = \{\eta_1, \ldots, \eta_t\}$ in this case. We recall Robbins–Siegmund Theorem [41] below, which is the key tool for proving both Theorem 1 and our generalization.

Theorem 2 (Robbins–Siegmund [41]) *Let $(\Omega, \mathscr{F}, P)$ be a probability space and $\mathscr{F}_1 \subseteq \mathscr{F}_2 \subseteq \cdots$ a sequence of sub-σ-fields of $\mathscr{F}$. Let U_t, β_t, ε_t and ζ_t, $t = 1, 2, \ldots$ be non-negative $\mathscr{F}_t$-measurable random variables such that*

$$\mathbb{E}(U_{t+1} \mid \mathscr{F}_t) \leq (1 + \beta_t)U_t + \varepsilon_t - \zeta_t, \quad t = 1, 2, \ldots \tag{11}$$

Then on the set $\left\{\sum_t \beta_t < \infty, \sum_t \varepsilon_t < \infty\right\}$, U_t *converges almost surely to a random variable, and* $\sum_t \zeta_t < \infty$ *almost surely.*

3 Dual stochastic natural gradient descent

Section 3.1 establishes that the family of joint distributions $P(\mathscr{Y}, \mathscr{X})$ satisfying the core MLR assumption (that is, $P(\mathscr{Y} \mid \mathscr{X})$ is an affine function of the features $\mathscr{X}$) is a LEF and hence a DFM. Then, we rely on duality to provide an efficient computation of the natural gradient of the log-loss function in Sect. 3.2. Finally, we provide the DSNGD algorithm in Sect. 3.3.

3.1 MLR generative model. The joint distribution

The following result proves that the family of joint distributions $P(\mathscr{Y}, \mathscr{X})$, which satisfy the core MLR assumption, is a LEF and hence a DFM.

Proposition 1 *The log-odds ratio of the class posteriors* $P(\mathscr{Y} \mid \mathscr{X})$ *is an affine function of the features* $\mathscr{X}$ *if and only if the joint distribution* $P(\mathscr{Y}, \mathscr{X})$ *belongs to the LEF family.*

Furthermore, there exists a LEF natural parametrization of the joint distribution

$$\begin{aligned} P_\eta(y, x) &= \frac{\exp\left(S(y)^\intercal \alpha + T(x)^\intercal \beta_y\right)}{\lambda(\eta)} \\ \lambda(\eta) &= \int_x \sum_y \exp\left(S(y)^\intercal \alpha + T(x)^\intercal \beta_y\right) \end{aligned} \tag{12}$$

where $\eta = (\alpha, \beta)$, $\alpha \in \mathbb{R}^{s-1}$, $\beta = [\beta_1 \cdots \beta_s] \in \mathbb{R}^{t \times s}$, β_y *is the* y*-th column of* β *and*

$$\begin{aligned} T &: \Omega \to \mathbb{R}^t \\ S &: [1, \ldots, s] \to \mathbb{R}^{s-1} \end{aligned} \tag{13}$$

are sufficient and minimal statistics of $\mathscr{X}$ *and* $\mathscr{Y}$*, respectively.*

The proof relies strongly on theorem 2 in [1] and can be found in Appendix *A*. This is convenient for our purposes. Recall from Sect. 2.1 that, in a DFM, the costs of natural gradient computations can be significantly reduced, based on the property shown by Eq. (6). Next, we provide the dually flat parametrization of $P(\mathscr{Y}, \mathscr{X})$.

3.1.1 Dually flat parametrization of the joint distribution

We have seen that $P(\mathscr{Y}, \mathscr{X})$ is a LEF and that we can choose the natural parametrization as given in Eq. (12). With a linear transformation, S can become a canonical statistic, that means, $S(i)_j = \delta_{i=j}$ for $1 \leq i < s$ and $S(s) = 0$. For simplicity, we fix

statistic S to be canonical from now on. The conditional probability distributions with η parametrization are

$$P_\eta(y \mid x) = \frac{\exp\left(S(y)^\intercal \alpha + T(x)^\intercal \beta_y\right)}{\sum_y \exp\left(S(y)^\intercal \alpha + T(x)^\intercal \beta_y\right)} \tag{14}$$

As [31] proves, the exponential family manifold is built after the convex function $F(\eta) = \log \lambda(\eta)$. The reference proves that this Riemannian manifold derived from $F(\eta)$, according to Eq. (5), has the Fisher information metric, as is usually considered for statistical manifolds, defined as

$$G_\eta = -\mathbb{E}_{x,y\sim P_\eta}\left[\nabla^2 \log P_\eta(y, x)\right] \tag{15}$$

Equation (5) also reveals the dual parametrization η^*. For LEF, it is called the expectation parametrization and it is shown below. For more properties of the dual parametrization see [31]. To simplify the notation, if $x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}$, we note $\nabla_x = \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix}^\intercal$. So for every $i \in \{1, \ldots, s\}$ write

$$\begin{aligned} \alpha^* = \nabla_\alpha F(\eta) &= \sum_y S(y) P_\eta(y) = \mathbb{E}_{\mathscr{Y}}[S(y)] \\ &= (P_\eta(\mathscr{Y} = 1), \ldots, (P_\eta(\mathscr{Y} = s-1))^\intercal \\ \beta_y^* = \nabla_{\beta_y} F(\eta) &= P_\eta(y) \int_{\mathscr{X}} T(x) P_\eta(x \mid y) = P_\eta(y) \mathbb{E}_{\mathscr{X}\mid y}[T(x)] \end{aligned} \tag{16}$$

Define $\eta^* = (\alpha^*, \beta^*)$ with $\beta^* = [\beta_1^* \cdots \beta_s^*]$ the dual parameterization, or equivalently, the expectation parameters.

Remark 1 $P(\mathscr{Y})$ is a categorical distribution (since $\mathscr{Y}$ is discrete and finite) and therefore it is a LEF, where α^* are actually the expectation parameters.

3.2 Fast natural gradient of the log-loss

This section allows to compute the natural gradient of the log-loss function without having to use the metric matrix directly, but using both dual parametrizations instead.

Given $(y, x) \in \mathscr{Y} \times \mathscr{X}$ and $\eta \in \mathbb{R}^k$, the log-loss function is defined as

$$l(\eta, y, x) = -\log P_\eta(y \mid x) \tag{17}$$

Below result reveals $\widetilde{\nabla} l(\eta, x, y)$ using both dual parametrizations η and η^*.

Proposition 2 *Let l be the log-loss function and let $P(\mathscr{Y}, \mathscr{X})$ be a DFM. Then,*

$$\widetilde{\nabla} l(\eta, y, x) = \nabla h(x, \eta^*) \cdot q(y, x, P_\eta) \tag{18}$$

where

$$q(y, x, P) = \begin{pmatrix} P(\mathscr{Y} = 1|x) \\ \vdots \\ P(\mathscr{Y} = s|x) \end{pmatrix} - e_s(y), \tag{19}$$

$h(x, \eta^*) = (\log P_{\eta^*}(\mathscr{Y} = 1, x), \ldots \log P_{\eta^*}(\mathscr{Y} = s, x))$ *and* $e_s(k)$ *is the* k*-th canonical* s*-dimensional vector.*

Proof First, claim that

$$\nabla l(\eta, x, y) = \nabla h(x, \eta) \cdot q(y, x, P_\eta) \tag{20}$$

Indeed,

$$\begin{aligned} \nabla \log P_\eta(y \mid x) &= \nabla \log P_\eta(y, x) - \nabla \log \sum_y P_\eta(y, x) \\ &= \nabla \log P_\eta(y, x) - \frac{\sum_y \nabla P_\eta(y, x)}{\sum_y P_\eta(y, x)} \\ &= \nabla \log P_\eta(y, x) - \frac{\sum_y P_\eta(y, x) \nabla \log P_\eta(y, x)}{\sum_y P_\eta(y, x)} \\ &= \nabla \log P_\eta(y, x) - \sum_y P_\eta(y \mid x) \nabla \log P_\eta(y, x) \\ &= \nabla h(y, x, \eta) - \mathbb{E}_{\mathscr{Y}|x}[\nabla h(y, x, \eta)] \end{aligned} \tag{21}$$

where $h(y, x, \eta) = \log P_\eta(y, x)$. Observe we can rewrite Eq. (21) as;

$$\nabla \log P_\eta(y \mid x) = -\nabla h(x, \eta) \cdot q(y, x, \eta) \tag{22}$$

where $h(x, \eta) = (h(1, x, \eta), \ldots, h(s, x, \eta))$ implying the claim. From Eq. (20) observe that

$$\widetilde{\nabla} l(\eta, x, y) = \widetilde{\nabla} h(x, \eta) \cdot q(y, x, P_\eta) \tag{23}$$

Finally, since the log-loss is defined in a DFM, then use previous equation and Eq. (6) to finish the proof □

3.3 DSNGD definition

DSNGD aims to solve the MLR optimization problem using the natural parametrization η of the LEF on $\mathscr{Y} \times \mathscr{X}$. Specifically, given an unknown probability distribution, if $\overline{P}$ is an unknown probability distribution over $\mathscr{Y} \times \mathscr{X}$, the goal is to optimize $\mathbb{L}(\eta) = \mathbb{E}_{x,y\sim\overline{P}}[l(\eta, y, x)]$ for $\eta \in \mathbb{R}^k$ where $l(\eta, y, x)$ is the log-loss function. The solution $\overline{\eta} \in \mathbb{R}^k$ to this problem refers to the conditional distributions $P_{\overline{\eta}}(\mathscr{Y} \mid \mathscr{X})$ that better fits the hidden conditional distributions $\overline{P}(\mathscr{Y}|\mathscr{X})$. To that end, we define a stochastic natural gradient based algorithm.

Using Proposition 2, define DSNGD by the update equation

Definition 1 (*DSNGD update*)

$$\eta_{t+1} = \eta_t - \gamma_t \nabla h(x_t, \zeta_t^*) \cdot q(y_t, x_t, P_{\eta_t}) \tag{24}$$

where $\{\zeta_t^*\}_{t\in\mathbb{N}}$ is a sequence in the expectation parametrization such that $\{\zeta_t\}_{t\in\mathbb{N}}$ converges.

Note that $q(y_t, x_t, P_{\eta_t})$ is a stable term, as it only takes values between -1 and 1. Moreover, DSNGD ensures the stability of the term $\nabla h(x_t, \zeta_t^*)$, since ζ_t is a convergent sequence. This mirrors the strategy of CSNGD, and similarly, it guarantees the convergence of the algorithm, as shown in Theorem 5. It is also important to observe that Eq. (24) is well-defined even when the parameterization is not minimal—when T is not a minimal statistic. Therefore DSNGD can be applied in general cases where S and T are not minimal. The steps taken by DSNGD are outlined in Algorithm 1.

Algorithm 1: DSNGD

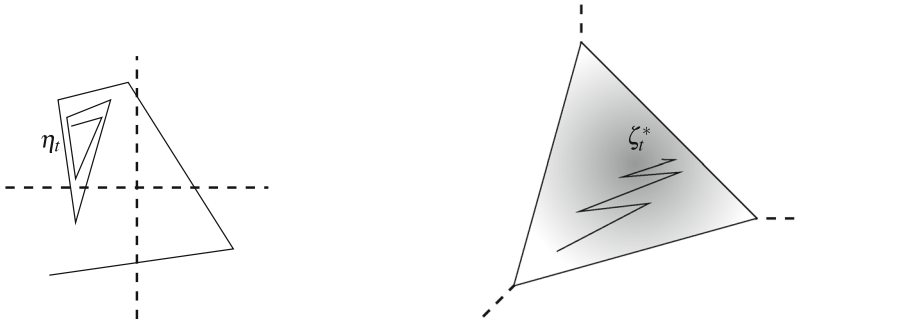
Result: η

1 $\eta \leftarrow \eta_0, \zeta^* \leftarrow \zeta_0^*, \gamma \leftarrow \gamma_0$;
2 **while** *observations* x, y *and stopping condition is false* **do**
3 $\quad q \leftarrow q(y, x, P_\eta)$;
4 $\quad gh \leftarrow \nabla h(x, \zeta^*)$;
5 $\quad d \leftarrow gh \cdot q$;
6 $\quad \eta \leftarrow \eta - \gamma \cdot d$;
7 $\quad$ update ζ^*;
8 $\quad$ update γ
9 **end**

The sequence $\{\zeta_t^*\}_{t\in\mathbb{N}}$, or simply ζ_t^* for brevity, can be any sequence in the dual space, as long as its dualized counterpart $\{\zeta_t\}_{t\in\mathbb{N}}$ converges. For instance, it can be a constant sequence. The proposed algorithm monitors two independent sequences; the primary sequence η_t which estimates the solution $\overline{\eta}$ to the problem, and the sequence ζ_t^*, which is defined in the dual space and chosen to satisfy the convergence condition.

Consider, for example, the trivial case where $\mathscr{X} = \{0\}$ and $\mathscr{Y} = \{0, 1, 2\}$. In this scenario, the sole conditional probability distribution of the problem is the Categorical distribution $P(\mathscr{Y} \mid \mathscr{X} = 0)$. This space is represented by $\mathbb{R}^2$ and its dual space is represented by the simplex S^2. In this context, the primary sequence η_t evolves within $\mathbb{R}^2$ while the auxiliary sequence ζ_t^* progresses through S^2. Figure 1 depicts the trajectories of η_t (instruction line 6 of the algorithm) and ζ_t^* (instruction line 7 of the algorithm) when applying DSNGD for this simple example.

It is crucial to note that the sequence ζ_t^* can be freely chosen as long as its dualized counterpart is convergent. Since DSNGD is a natural gradient-based algorithm, it takes effective natural gradient steps only when η_t and ζ_t^* correspond to the same point in the probability distribution, as specified in Eq. (24) and Proposition 2. The proof of DSNGD's convergence to the solution $\overline{\eta}$ is provided in Sect. 5. If ζ_t^* is chosen to converge to the solution as well, the two sequences will gradually approach each other during optimization, making DSNGD steps more closely approximate natural gradient



(a) η_t sequence in $\mathbb{R}^2$ (b) ζ_t^* sequence in the dual space S^2

Fig. 1 η_t and ζ_t^* sequences of DSNGD where $\mathscr{X} = \{0\}$ and $\mathscr{Y} = \{0, 1, 2\}$

steps. Hence, and since DSNGD is designed to approximate SNGD, it is advisable to ensure that the sequence ζ_t^* converges to the solution $\overline{\eta}^* = \nabla F(\overline{\eta})$. The theoretical convergence rate is not discussed here, as it strongly depends on the choice of the auxiliary sequence ζ_t^*, which requires further research.

The convergence of the auxiliary sequence can be achieved, for instance, by defining ζ_t^* as a maximum a posteriori estimate of the parameters $P(\mathscr{Y}, \mathscr{X})$ based on the data observed up to iteration t. Other options include SGD, CSNGD or even an independent version of DSNGD that converges. Observe that the convergence point of such algorithms is the solution to the problem, assuming standard regularities on the function to optimize. Recall that this situation is recommended since DSNGD estimates of the natural gradient are more accurate when both primal and auxiliary sequence are close.

Although both DSNGD and mirror descent rely on duality, there are key differences. The main distinction is that mirror descent maintains a sequence of points within the manifold, expressed in both primal and dual parametrizations, whereas DSNGD has two independent sequences evolving in the primal and dual spaces, which may not be directly connected. This is especially important because mirror descent requires the computation of dual coordinates, whereas DSNGD can be applied in spaces where the dual coordinates are not efficiently computable given the primal coordinates

Example 1 Let $\mathscr{Y} = \{1, 2\}$ and $\mathscr{X} = \{1, 2\}$ and minimal and canonical statistics S and T. Let $\eta = (\alpha, \beta)$ be the natural parameter and $\zeta^* = (\alpha^*, \beta^*)$ be the independent dual parameter. Observe that in this case, α and α^* are 1-element vectors and β and β^* are 2-element not squared matrices. In this example we complete an iteration of discrete DSNGD algorithm, following the instructions listed in Algorithm 1.

Let $(y, x) = (2, 1)$ be an observation. Statistics T and S are assumed to be canonical. Instruction line 3 consist on using Eq. (14) to compute

$$q(y = 2, x = 1, P_\eta) = \begin{pmatrix} P_\eta(\mathscr{Y} = 1 \mid x) \\ P_\eta(\mathscr{Y} = 2 \mid x) \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} = R \cdot \begin{pmatrix} \exp(\alpha_1 + \beta_1) \\ \exp \beta_2 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{25}$$

where $R = \frac{1}{\exp(\alpha_1+\beta_1)+\exp\beta_2}$. For instruction line 4, express function $h(x, \zeta^*)$ (use Eq. (16)), then apply the gradient.

$$h(x=1,\zeta^*) = (\log\beta_1^*, \log\beta_2^*) \rightarrow \nabla h(x=1,\zeta^*) = \begin{pmatrix} 0 & 0 \\ \frac{1}{\beta_1^*} & 0 \\ 0 & \frac{1}{\beta_2^*} \end{pmatrix} \tag{26}$$

Proceed now with instruction line 5. It computes the approximation of the natural gradient and the direction that DSNGD uses for the η update.

$$\begin{aligned} d &= \nabla h(x=1,\zeta^*) q(y=2, x=1, P_\eta) \\ &= \begin{pmatrix} 0 & 0 \\ \frac{1}{\beta_1^*} & 0 \\ 0 & \frac{1}{\beta_2^*} \end{pmatrix} \cdot \begin{pmatrix} R\cdot\exp(\alpha_1+\beta_1) \\ (R\cdot\exp\beta_2)-1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \frac{R\cdot\exp(\alpha_1+\beta_1)}{\beta_1^*} \\ \frac{(R\cdot\exp\beta_2)-1}{\beta_2^*} \end{pmatrix} \end{aligned} \tag{27}$$

Next instruction lines of the algorithm are standard to update the parameter vector $(\alpha, \beta_1, \beta_2)$ using direction d, so there is no need to go further. If for instance the observation is $(y, x) = (2, 2)$, then the approximation of the natural gradient is

$$d = \begin{pmatrix} \frac{R\exp\alpha_1}{\alpha_1^*-\beta_1^*} - \frac{R-1}{1-\alpha_1^*-\beta_2^*} \\ \frac{-R\cdot\exp\alpha_1}{\alpha_1^*-\beta_1^*} \\ \frac{R-1}{1-\alpha_1^*-\beta_2^*} \end{pmatrix} \tag{28}$$

where $R = \frac{1}{1+\exp\alpha_1}$.

4 Discrete DSNGD and computational complexity

In this section, assume that space $\mathscr{X}$ is discrete, specifically $\mathscr{X} = \{1, \ldots, m\}$ for some $m \in \mathbb{N}$. While $\mathscr{X}$ can be high-dimensional, it is assumed to have a finite number of distinct states. For simplicity, assume that T is also a canonical statistic, that is, $T(i)_j = \delta_{i=j} \in \mathbb{R}^{m-1}$ for $1 \leq i < m$ and $T(m) = 0$. Our Theorem 3 deduces and proves that the complexity order for discrete DSNGD of one iteration is linear on the dimension of the parameter η, assuming that the algorithm used for ζ^* is linear.

Before analyzing the computational complexity of DSNGD, it is necessary to determine the generator of ζ_t^* sequence. Sequence ζ_t^* belongs to the dual space of the LEF distributions on $\mathscr{Y} \times \mathscr{X}$, and if S and T are canonical statistics then it implies that ζ_t^* are directly the probabilities $P(y, x)$ after Eq. (16). It is possible to select the well known maximum a posteriori (MAP) estimator with parameter $a \in \mathbb{R}$. This estimator

is a simple counting of observations over the discrete space $\mathscr{Y} \times \mathscr{X}$ with a starting assumption of incidence of a for every event y, x. This estimator is linear and it clearly converges (to the solution).

To evaluate the computational complexity of DSNGD, we have to evaluate the costs derived from Eq. (18). We introduce the following notation:

$$K_i = \left(\begin{array}{c|c} & -1 \\ Id_{i-1} & \vdots \\ & -1 \end{array} \right), \tau(x) = \begin{cases} T(x) & x \neq m \\ -\mathbf{1}_t & x = m \end{cases}, D(x, \eta) \tag{29}$$

where Id_n is the n-th dimensional identity squared matrix, $\mathbf{1}_t$ is the vector of dimension t filled with ones and $D(x, \eta)$ is the diagonal matrix generated by the diagonal terms $\frac{1}{P_{\zeta^*}(x, \mathscr{Y}=1)}, \ldots, \frac{1}{P_{\zeta^*}(x, \mathscr{Y}=s)}$. Proposition 3 expresses the natural gradient approximation of DSNGD given discrete variable $\mathscr{X}$. The proof can be found in the Appendix B.

Proposition 3 *Let $\mathscr{X} = \{1, \ldots, m\}$ and let S and T be canonical statistics. Then, the natural gradient approximation of DSNGD is*

$$\begin{aligned} \nabla_{\alpha^*} h(x, \zeta^*) \cdot q(y, x, P_\eta) &= \mathbb{1}_{x=m} \cdot K_s \cdot D(x, \zeta^*) \cdot q(y, x, P_\eta) \\ \nabla_{\beta^*} h(x, \zeta^*) \cdot q(y, x, P_\eta) &= \tau(x) \cdot [D(x, \zeta^*) \cdot q(y, x, P_\eta)]^T \end{aligned} \tag{30}$$

Remark 2 The approximation of the natural gradient of Proposition 3 is presented maintaining the dimensions of $\eta = (\alpha, \beta)$. That is, the first component is a vector of the same dimension as α and the second component is a matrix with matching dimension with β.

Now it is possible to analyze the computational complexity of discrete DSNGD. Next theorem proves that DSNGD, just as SGD, is a linear algorithm on the manifold dimension, that is, the number of computations of DSNGD is affine in the manifold dimension.

Remark 3 By Proposition 1, the manifold dimension is $k = s - 1 + s \cdot t$ and therefore, an iteration of an algorithm is linear on the number of the variables of the model if its complexity order is $O(k) = O(s(1 + t)) = O(st)$

Theorem 3 *Let $\mathscr{X} = \{1, \ldots, m\}$ and let S and T be canonical statistics. Assume estimator ζ^* of DSNGD is linear. Then DSNGD iterations have linear complexity order on the manifold dimension.*

Proof Let $k = (s - 1) + s \cdot t$ be the dimension of η. Then $O(k) = O(st)$. Analyze the computational complexity of discrete DSNGD. That is, analyze the computational cost of instruction lines 3, 4, 5, 6, 7 and 8 shown in Algorithm 1.

Instruction line 3: Computing $q(y, x, P_\eta)$ is accomplished by computing $P_\eta(y \mid x)$ for all y. Using Eq. (14), and assuming that S and T are canonical statistics, $q(y, x, P_\eta)$ needs no operations for the products $T(x)^\intercal \beta_y$ and $S(y)^\intercal \alpha$. It requires 1 subtraction

in $S(y)^\intercal \alpha - T(x)^\intercal \beta_y$, then 1 exponentiation and finally 1 division. This is done for every $y \in \mathscr{Y}$. The denominator is the same for every y so it can be computed just once with $s-1$ sums. One more subtraction is done for the canonical vector $e_s(y)$. The total is $4s$ operations.

Instruction line 4 and 5: Instruction lines 4 and 5 are accomplished together by Proposition 3. Matrix $D(x, \zeta^*)$ requires to compute, for every y,

$$P_{\zeta^*}(y,x) = \begin{cases} (\beta_y^*)_x & x \neq m \\ (\alpha_y^* - |\beta_y^*|) & x = m,\ y \neq s \\ (1 - |\alpha^*| - |\beta_s^*|) & x = m,\ y = s \end{cases} \tag{31}$$

where $|v|$ with v a vector refers to the sum of all the vector components. When $x \neq m$ we are not required to perform any operation. Therefore, we will assume the worst case scenario were $x = m$. In this case, we can see that we need t operations when $y \neq s$ and $t+s$ when $y = s$. Additionally, we need to power those values to -1. This adds a total of $ts + 2s$ for computing $D(x, \zeta^*)$.

We can proceed to calculate the amount of operations of matrix products in Proposition 3. Notice that the only product that really consume operations is $D(x, \zeta^*) \cdot q(y, x, P_\eta)$, which requires s operations. The rest of matrix and vector products are just replacements and matrices dimension restructure operations.

Hence, instruction lines 4 and 5 require a total of $ts + 3s$ operations.

Instruction line 6: Updating every parameter requires requires $2k$ operations, since we have to multiply every parameter by a learning rate and then perform the update in every parameter.

Instruction line 7: This instruction is assumed in the proposition statement to have a linear complexity in the manifold dimension, hence $O(k)$ operations.

Instruction line 8: The learning rate γ is generally a decreasing sequence such as $\frac{1}{i}$ where i is the iteration, or similar sequences which are not carrying computational complexity dependent to the manifold dimension.

Total cost: The computational complexity of DSNGD is

$$O(4s + ts + 3ts + O(k)) = O(4(s + ts) + ts) = O(ts) \tag{32}$$

□

5 Discrete DSNGD convergence

In this section we prove the convergence of the discrete DSNGD. Discrete DSNGD refers to the case where $\mathscr{X} = \{1, \ldots, m\}$ for some $m \in \mathbb{R}$. We start by generalizing Theorem 3.2 in Sunehag's et al. [40] (introduced above and referred to from now on as Theorem 1) in Sect. 5.1. This generalization provides enough flexibility so as to be used later to prove the convergence of DSNGD in Sect. 5.2.

5.1 Generalization of convergence theorem

Theorem 1 is used to prove CSNGD convergence, however it can not be used to prove DSNGD convergence. First, because it demands the vector it follows to be factored as the product of a symmetric and positive definite matrix B_t and a vector Y_t that approximates the gradient (condition **C.1**). But DSNGD is defined to directly approximate the natural gradient, without the gradient as reference. And second, even if DSNGD is written as the product of a matrix and a vector, matrix $\nabla h(x_t, \zeta_t^*)$ is not square. So we need a more general convergence theorem, a result that directly contemplates the update direction assuming no factorization of such vector.

It is formally stated below. Proof is found in Appendix C.

Theorem 4 *Let $\mathbb{L} : \mathbb{R}^k \to \mathbb{R}$ be a twice differentiable function with a unique minimum $\overline{\eta}$ and $\eta_{t+1} = \eta_t - \gamma_t Y_t$. Then η_t converges to $\overline{\eta}$ almost surely if the following conditions hold*

$$
\begin{aligned}
&\textbf{\textit{C.2}}\ (\exists K)(\forall \eta)\ \|\nabla^2_\eta \mathbb{L}(\eta)\| \leq 2K \\
&\textbf{\textit{C.3}}\ (\forall \delta > 0) \inf_{\mathbb{L}(\eta_t) - \mathbb{L}(\overline{\eta}) > \delta} \nabla \mathbb{L}(\eta_t)^T \mathbb{E}\left[Y_t\right] > 0 \\
&\textbf{\textit{C.4}}\ (\exists A, B)(\forall t)\ \mathbb{E}\|Y_t\|^2 \leq A + B \cdot \mathbb{L}(\eta_t) \\
&\textbf{\textit{C.6}}\ \sum_t (\gamma_t)^2 < \infty,\ \sum_t \gamma_t = \infty
\end{aligned}
$$

Our result proves almost sure convergence of the sequence

$$\eta_{t+1} = \eta_t - \gamma_t Y(\eta_t, \mathscr{F}_t) \tag{33}$$

where $Y(\eta, \mathscr{F}_t)$ is a family of random vectors defined for every η and for every set

$$\mathscr{F}_t = \{(y_i, x_i) \mid i < t\} \tag{34}$$

As an abuse of notation write Y_t meaning the random variable $Y(\eta_t, \mathscr{F}_t) \in \mathbb{R}^n$. The main modification with respect to Theorem 1 is that we unify conditions C.1 and C.3

$$
\begin{aligned}
&\textbf{C.1}\ (\forall t)\ \mathbb{E}_t Y_t = \nabla \mathbb{L}(\eta_t) \\
&\textbf{C.3}\ (\forall \delta > 0) \inf_{\mathbb{L}(\eta) - \mathbb{L}(\overline{\eta}) > \delta} \|\nabla \mathbb{L}(\eta)\| > 0
\end{aligned}
\tag{35}
$$

to instead require

$$\textbf{C.3}\ (\forall \delta > 0) \inf_{\mathbb{L}(\eta_t) - \mathbb{L}(\overline{\eta}) > \delta} \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t\left[Y_t\right] > 0 \tag{36}$$

New condition **C.3** uses $\mathbb{E}_t$, referring to the conditional expectation given $\mathscr{F}_t$ of Eq. (34), which is a generalization of the definition of $\mathbb{E}_t$ in Sunehag [40].

Theorem 1 imposes that the expectation of the step taken must be the gradient and that the norm of the gradient must not approach to zero outside any environment of the minimum. Instead, we impose that the expectation of the step taken must not approach

to the border of the half-space which has the gradient as its normal vector, unless we are approaching the minimum simultaneously. This is a more general condition. Furthermore, condition **C.5** on the maximum and minimum eigenvalues of the matrix B_t can also be removed. In fact, our result proves the convergence of algorithms with scaling matrices B_t whose spectrum is not bounded from below by a strictly positive number, as long as new version of condition **C.3** holds.

5.2 Proof of convergence

Next, we show how Theorem 4 can be used to prove DSNGD convergence in the discrete case. That is, we use it to prove the next result:

Theorem 5 *Let $\mathscr{Y}$ and $\mathscr{X}$ be discrete variables, and S and T be canonical statistics. Then, DSNGD converges almost surely to the optimum.*

The proof consists on showing that conditions **C.2, C.3, C.4** and **C.6** of Theorem 4 hold. Condition **C.6** is assumed to hold, by just selecting an appropriate sequence of learning rates γ_t. Next, we prove conditions **C.2, C.3** and **C.4**.

Proof (**C.2** condition) Compute the hessian of

$$\mathbb{L}(\eta) = \sum_{x,y} l(\eta, y, x)\overline{P}(y, x) \tag{37}$$

where $\overline{P}$ stands for the true hidden probability distribution in $\mathscr{X} \times \mathscr{Y}$. The gradient of $l(\eta, y, x)$ is

$$\begin{aligned} \nabla_\alpha l(\eta, y, x) &= S \cdot q(y, x, P_\eta) \\ \nabla_{\beta_{y'}} l(\eta, y, x) &= q(y, x, P_\eta)_{y'} \cdot T(x) \end{aligned} \tag{38}$$

where S is the matrix having $S(i)$ as i-th column for $i \in \mathscr{Y}$. Therefore, the hessian is

$$\begin{aligned} \nabla^2_\alpha l(\eta, y, x) &= S \cdot (D(v(x)) - v(x) \cdot v(x)^\intercal) \cdot S^\intercal \\ \nabla_{\beta_{y_2}} \nabla_{\beta_{y_1}} l(\eta, y, x) &= \nabla_{\beta_{y_2}} q(y, x, P_\eta)_{y_1} \cdot T(x) \\ &= -T(x) \cdot T(x)^\intercal q(y, x, P_\eta)_{y_1} q(y, x, P_\eta)_{y_2} \\ \nabla_\alpha \nabla_{\beta_{y'}} l(\eta, y, x) &= \nabla_\alpha q(y, x, P_\eta)_{y'} \cdot T(x) \\ &= T(x) \cdot q(y', x, P_\eta)^\intercal \cdot S^\intercal \end{aligned} \tag{39}$$

where $v(x) = (P(\mathscr{Y} = 1 \mid x), \ldots, P(\mathscr{Y} = s \mid x))$ and $D(v(x))$ is the diagonal matrix containing $v(x)$ terms.

Observe how all matrices in Eq. (39) have their elements bounded once S and T statistics are fixed, since $\|q(y, x, P_\eta)\| \leq 1$. Therefore

$$\|\nabla^2 l(\eta, y, x)\| \leq 2K_{x,y} \tag{40}$$

for some positive numbers $K_{x,y}$. Define $K = \max_{x,y} K_{x,y}$. then finally

$$\begin{aligned}
\|\nabla^2_\eta \mathbb{L}(\eta)\| &= \left\| \nabla^2 \sum_{y,x} l(\eta, y, x) \cdot \overline{P}(x, y) \right\| \\
&= \left\| \sum_{y,x} \nabla^2 l(\eta, y, x) \cdot \overline{P}(x, y) \right\| \\
&\leq \sum_{y,x} \|\nabla^2 l(\eta, y, x)\| \cdot \overline{P}(x, y) \\
&\leq \sum_{y,x} 2 \cdot K_{x,y} \cdot \overline{P}(x, y) \\
&\leq 2 \cdot K \sum_{y,x} \overline{P}(x, y) \\
&= 2 \cdot K
\end{aligned} \tag{41}$$

□

Proof (**C.4** condition)

Observe that for any ε and t large enough there exists A_{x_t} such that

$$\begin{aligned}
\|Y_t\|^2 &= q(y_t, x_t, P_\eta)^\intercal \cdot h(x_t, \zeta_t^*)^\intercal h(x_t, \zeta_t^*) \cdot q(y_t, x_t, P_\eta) \\
&\leq A_{x_t} \|q(y_t, x_t, P_\eta)\|^2
\end{aligned} \tag{42}$$

where

$$A_{x_t} \geq \|h(x_t, \zeta_t^*)^\intercal h(x_t, \zeta_t^*)\| + \varepsilon \tag{43}$$

This is because ζ_t converges and because of Proposition 3. Now

$$\begin{aligned}
\|q(y_t, x_t, P_\eta)\|^2 &= 1 - 2P_{\eta_t}(y_t|x_t) + \sum_y P_{\eta_t}(y|x_t)^2 \\
&\leq s + 1
\end{aligned} \tag{44}$$

therefore $\|Y_t\|^2 \leq A_{x_t}(s+1)$ and

$$\begin{aligned}
\mathbb{E}\|Y_t\|^2 &\leq \mathbb{E}[A_{x_t}(s+1)] \\
&\leq A'(s+1) = A
\end{aligned} \tag{45}$$

where $A' = \max_x A_x$ and then condition **C.4** holds. □

Proof (**C.3** condition)

Compute the gradient of $\mathbb{L}(\eta)$ (see Eq. (20)) and use Proposition 2 to obtain $\mathbb{E}_t[Y_t]$ involved in condition **C.3**.

$$\begin{aligned}\nabla\mathbb{L}(\eta) &= \mathbb{E}_t\left[\nabla l(\eta, x, y)\right] \\ &= \sum_x \nabla h(x,\eta) \sum_y q(y, x, P_\eta)\overline{P}(x, y) \\ &= \sum_x \nabla h(x,\eta)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ \mathbb{E}_t\left[Y_t\right] &= \sum_x \nabla h(x,\zeta^*)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta)\end{aligned} \tag{46}$$

where

$$\mathrm{diff}(y, x, P_\eta) = (q(y, x, P_\eta) - q(y, x, \overline{P}))\overline{P}(x). \tag{47}$$

Further evolve Eq. (46) to finally multiply $\nabla\mathbb{L}(\eta)^\intercal\mathbb{E}_t\left[Y_t\right]$ and check condition **C.3**. Continue by developing $\nabla\mathbb{L}(\eta)$ first, precisely compute $\nabla h(x,\eta)$. To simplify the notation, decompose $\nabla = (\nabla_\alpha, \nabla_{\beta_1}, \ldots, \nabla_{\beta_s})$

$$\begin{aligned}\nabla_\alpha h(x,\eta) &= S + u(P_\eta)\cdot(1,\ldots,1)\;\; u(P) = -\sum_y S(y)P(y) \\ \nabla_{\beta_y} h(x,\eta) &= T\cdot e_m(x)\cdot e_s(y)^\intercal + v(y, P_\eta)\cdot(1,\ldots,1)\;\; v(y,P) \\ &= -\sum_x T(x)P(y,x)\end{aligned} \tag{48}$$

Since $(1,\ldots,1)\cdot\mathrm{diff}_{\mathscr{Y}}(x,\eta) = 0$ then

$$\begin{aligned}\nabla_\alpha\mathbb{L}(\eta) &= \sum_x \nabla_\alpha h(x,\eta)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ &= \sum_x S\cdot\mathrm{diff}_{\mathscr{Y}}(x,\eta) = S\cdot\sum_x \mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ \nabla_{\beta_y}\mathbb{L}(\eta) &= \sum_x \nabla_{\beta_y} h(x,\eta)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ &= T\cdot\sum_x e_m(x)e_s(y)^\intercal\,\mathrm{diff}_{\mathscr{Y}}(x,\eta)\end{aligned} \tag{49}$$

Now develop $\mathbb{E}_t Y_t$ further. Recall that the canonical parametrization is selected so plug in Proposition 3 into Eq. (46). Decompose $\mathbb{E}_t = (\mathbb{E}_{t,\alpha^*}, \mathbb{E}_{t,\beta_1^*}, \ldots, \mathbb{E}_{t,\beta_s^*})$

$$\begin{aligned}\mathbb{E}_{t,\alpha^*}\left[Y_t\right] &= \sum_x \nabla_{\alpha^*} h(x,\zeta^*)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ &= K_s\cdot D\left(\frac{1}{P_{\zeta^*}(\mathscr{X}=m,\mathscr{Y}=1)},\ldots,\frac{1}{P_{\zeta^*}(\mathscr{X}=m,\mathscr{Y}=s)}\right)\cdot\mathrm{diff}_{\mathscr{Y}}(m,\eta) \\ \mathbb{E}_{t,\beta_y^*}\left[Y_t\right] &= \sum_x \nabla_{\beta_y^*} h(x,\zeta^*)\,\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ &= K_m\cdot\sum_x \frac{1}{P_{\zeta^*}(x,y)} e_m(x)\cdot e_s(y)^T\cdot\mathrm{diff}_{\mathscr{Y}}(x,\eta) \\ &= K_m\cdot D\left(\frac{1}{P_{\zeta^*}(\mathscr{X}=1,y)},\ldots,\frac{1}{P_{\zeta^*}(\mathscr{X}=m,y)}\right)\cdot\mathrm{diff}_{\mathscr{X}}(y,\eta)\end{aligned} \tag{50}$$

where

$$\operatorname{diff}_{\mathscr{X}}(y,\eta) = \begin{pmatrix} (P_\eta(\mathscr{Y}=y \mid \mathscr{X}=1) - \overline{P}(\mathscr{Y}=y \mid \mathscr{X}=1))\overline{P}(\mathscr{X}=1) \\ \vdots \\ (P_\eta(\mathscr{Y}=y \mid \mathscr{X}=m) - \overline{P}(\mathscr{Y}=y \mid \mathscr{X}=m))\overline{P}(\mathscr{X}=m) \end{pmatrix} \tag{51}$$

Proceed now to check the condition. Develop the products until obtain

$$\begin{aligned} \nabla\mathbb{L}(\eta)^\intercal \mathbb{E}_t[Y_t] &= \nabla_\alpha \mathbb{L}(\eta)^\intercal \mathbb{E}_{t,\alpha^*}[Y_t] + \sum_y \nabla_{\beta_y} \mathbb{L}(\eta)^\intercal \mathbb{E}_{t,\beta_y^*}[Y_t] \\ &= \sum_{y,x} \frac{1}{P_{\zeta^*}(x,y)} (P_\eta(y|x) - \overline{P}(y|x))^2 \overline{P}(x)^2 \end{aligned} \tag{52}$$

Notice in Eq. (52) that $\nabla\mathbb{L}(\eta)^\intercal \mathbb{E}_t[Y_t]$ is a sum of positive numbers, and it vanishes only if $\eta = \overline{\eta}$. Also, since $\frac{1}{P_{\zeta^*}(x,y)} > 1$, observe that

$$\begin{aligned} \nabla\mathbb{L}(\eta)^\intercal \mathbb{E}_t[Y_t] &> \sum_{y,x} (P_\eta(y|x) - \overline{P}(y|x))^2 \overline{P}(x)^2 \\ &= \sum_y \| \operatorname{diff}_{\mathscr{X}}(y,\eta) \|^2 \end{aligned} \tag{53}$$

To finish proving the result, let $\{\eta_i\}_{i\in\mathbb{N}}$ be a sequence such that

$$\sum_y \| \operatorname{diff}_{\mathscr{X}}(y,\eta_i) \|^2 \xrightarrow[i\to\infty]{} 0 \tag{54}$$

since every term is positive, then for every $y \in \mathscr{Y}$

$$\| \operatorname{diff}_{\mathscr{X}}(y,\eta_i) \|^2 \xrightarrow[i\to\infty]{} 0 \tag{55}$$

implying that $P_{\eta_i}(y|x) - \overline{P}(y|x) \xrightarrow[i\to\infty]{} 0$ for all x, y and that

$$\mathbb{L}(\eta_i) - \mathbb{L}(\overline{\eta}) \xrightarrow[i\to\infty]{} 0 \tag{56}$$

Hence it's proven

$$(\forall \delta > 0) \quad \inf_{\mathbb{L}(\eta)-\mathbb{L}(\overline{\eta})>\delta} \sum_y \| \operatorname{diff}_{\mathscr{X}}(y,\eta_i) \|^2 > 0 \tag{57}$$

and therefore, after Eq. (53), condition **C.3** holds. □

6 Experiments

This section conducts experiments to assess the behavior of DSNGD. It explores the possible benefits and downsides of DSNGD when tested empirically against SGD and SNGD. We consider a more complex scenario inside the MLR manifold where we test the algorithms in different dimension and entropy set ups. Furthermore, we analyze the speed per iteration in the training of each method, to see how the theoretical low computational complexity of our algorithm really reflects in practical problems and how compares to classical SGD and SNGD.

The manifold considered for the experiments is presented in Sect. 6.1. Then, in Sect. 6.2, we prove that DSNGD has linear computational complexity assuming the Naive Bayes condition for the particular case where T is not canonical but canonical in every feature, also proving the convergence of DSNGD in this same case. Finally, in Sect. 6.3 we show and analyze the results of the experiments conducted. The code is available in [42].

6.1 The joint distribution and the discrete features vector

Consider a joint probability space with $\mathscr{X} = (\mathscr{X}_1, \dots, \mathscr{X}_r)$ where r is a positive number and $\mathscr{X}_i$ is a discrete variable for all $i \leq r$. This situation refers to many problems, where we want to predict the class of variable $\mathscr{Y}$ given a vector of discrete features. If $\mathscr{X}_i \in \{1, \dots, m_i\}$ we will denote $\mathscr{X} = (m_1, \dots, m_r)$ to simplify the notation.

For instance, the features vector $\mathscr{X}$ may consist of variables that describe a person's medical history, such as gender, age range, blood pressure status, and past diagnoses, all represented as discrete values. Hence every $x \in \mathscr{X}$ is a discrete features vector, that we will note $x = (x_1, \dots, x_r)$.

We define a statistic T for this case. The statistic T will not be a canonical statistic but a statistic canonical in every feature instead. That is, the vector

$$T(x) = (T_1(x_1) \cdots T_r(x_r))^\intercal, \tag{58}$$

where T_i is a canonical statistic of dimension $m_i - 1$. Observe that with this statistic T, it is

$$\begin{aligned} \beta^*_{y,j} &:= (P(\mathscr{X}_j = 1, \mathscr{Y} = y), \dots, P(\mathscr{X}_j = m_j - 1, \mathscr{Y} = y)) \\ \beta^*_y &= (\beta^*_{y,1} \cdots \beta^*_{y,r})^\intercal \end{aligned} \tag{59}$$

We will use these equations to determine the DSNGD update and code it. But before, we need to assume a condition in the space, that we introduce in next section.

6.2 Naive Bayes assumption

The Naive Bayes assumption is considered with the following equation

$$P(\mathscr{X} \mid \mathscr{Y}) = \prod_i P(\mathscr{X}_i \mid \mathscr{Y}) \tag{60}$$

Therefore, this assumption is showing a feature independence given the class variable.

To specify the update computations of DSNGD assuming a discrete features vector $\mathscr{X}$ and considering the Naive Bayes assumption, we introduce the notation:

$$\tau_i(x) \begin{cases} T_i(x) & x \neq m_i \\ -\mathbf{1}_{m_i-1} & x = m_i \end{cases}, D_{\mathscr{X}_i}(x, \zeta^*)i \leq r, \ D_{\mathscr{Y}}(\zeta^*) \tag{61}$$

where $D_{\mathscr{X}_i}(x, \zeta^*)$ is the diagonal matrix generated by terms $\frac{1}{P_{\zeta^*}(\mathscr{X}_i=x,\mathscr{Y}=1)}, \ldots, \frac{1}{P_{\zeta^*}(\mathscr{X}_i=x,\mathscr{Y}=s)}$ and $D_{\mathscr{Y}}(\zeta^*)$ is the diagonal matrix generated by $\frac{1}{P_{\zeta^*}(\mathscr{Y}=1)}, \ldots, \frac{1}{P_{\zeta^*}(\mathscr{Y}=s)}$.

Proposition 4 *Let $\mathscr{Y}$ and $\mathscr{X}$ be discrete where $\mathscr{X} = (m_1, \ldots, m_r)$ for some $r > 0$. Let S be the canonical statistic and T be a statistic canonical in every feature. Then*

$$\begin{aligned} &\nabla_{\alpha^*} h(x, \zeta^*) \cdot q(y, x, P_\eta) \\ &= K_s \cdot \left[(1-r) D_{\mathscr{Y}}(\zeta^*) + \sum_j D_{\mathscr{X}_j}(x_j, \zeta^*) \cdot \mathbb{1}_{x_j = m_j} \right] \cdot q(y, x, P_\eta) \\ &\nabla_{\beta^*} h(x, \zeta^*) \cdot q(y, x, P_\eta) \\ &= \begin{bmatrix} \tau_1(x_1) \cdot \left[D_{\mathscr{X}_1}(x_1, \zeta^*) \cdot q(y, x, P_\eta) \right]^\intercal \\ \vdots \\ \tau_r(x_r) \cdot \left[D_{\mathscr{X}_r}(x_r, \zeta^*) \cdot q(y, x, P_\eta) \right]^\intercal \end{bmatrix} \end{aligned} \tag{62}$$

where $\mathbb{1}_A$ is the indicator function at set A.

Proof To prove Proposition 4, we only need to express function h in terms of dual parameters α^* and β^* and then apply the derivatives, similarly as we did to prove Proposition 3. To express the function h in terms of dual parameters, it suffices to consider Eq. (16) for α^* expression and Eq. (59) for the β^* representation, and apply them in h

$$\begin{aligned} h(y, x, \zeta^*) &= \log P(y) + \sum_i \log P(\mathscr{X}_i = x_i \mid y) \\ &= (1-r) \log P(y) + \sum_i^r \log P(\mathscr{X}_i = x_i, y) \\ &= \begin{cases} (1-r) \log \alpha_y^* & y \neq s \\ (1-r) \log(1 - |\alpha^*|) & y = s \end{cases} \\ &\quad + \sum_i^r \begin{cases} \log \left[(\beta_{y,i}^*)_{x_i} \right] & x_i \neq m_i - 1 \\ \log(\alpha_i^* - |\beta_{y,i}^*|) & x_i = m_i - 1, y \neq s \\ \log(1 - |\alpha^*| - |\beta_{y,i}^*|) & x_i = m_i - 1, y = s \end{cases} \end{aligned} \tag{63}$$

where $|v|$ stands for the sum of all coordinates of any vector v. Finally, apply the derivatives to h and proceed as in proof of Proposition 3 to prove the result. □

From Proposition 4 we can deduce the next results.

Theorem 6 *Let $\mathscr{Y}$ and $\mathscr{X}$ be discrete where $\mathscr{X} = (m_1, \ldots, m_r)$ for some $r > 0$. Let S be the canonical statistic and T be a statistic canonical in every feature. Assume estimator ζ^* of DSNGD is linear. Then discrete DSNGD iterations have linear complexity order on the manifold dimension.*

Theorem 7 *Let $\mathscr{Y}$ and $\mathscr{X}$ be discrete where $\mathscr{X} = (m_1, \ldots, m_r)$ for some $r > 0$. Let S be the canonical statistic and T be a statistic canonical in every feature. Then, DSNGD converges almost surely to the optimum.*

Theorems 6 and 7 can be easily proven by using Proposition 4, following the same strategy as when we used Proposition 3 to prove Theorems 3 and 5.

6.3 Experiment results

This section tests DSNGD in the discrete MLR space where $\mathscr{X}$ is a discrete features vector. Algorithms SGD, SNGD and adagrad are added por comparison purposes. The code is available in [42].

We consider 3 different manifolds $\mathscr{M}_1$, $\mathscr{M}_2$ and $\mathscr{M}_3$ defined as

$$\begin{aligned}
\mathscr{M}_1 &= \begin{cases} \mathscr{Y} = \{1, \ldots, 10\} \\ \mathscr{X} = (10, 5) \end{cases} \\
\mathscr{M}_2 &= \begin{cases} \mathscr{Y} = \{1, \ldots, 20\} \\ \mathscr{X} = (10, 5, 10, 5) \end{cases} \\
\mathscr{M}_3 &= \begin{cases} \mathscr{Y} = \{1, \ldots, 30\} \\ \mathscr{X} = (10, 5, 10, 5, 10, 5) \end{cases}
\end{aligned} \tag{64}$$

where the dimensions are 139, 539 and 1199 for manifolds $\mathscr{M}_1$, $\mathscr{M}_2$ and $\mathscr{M}_3$ respectively. We examine the performance of DSNGD under various scenarios characterized by differing levels of entropy: low, medium, and high. These scenarios are distinguished by the parameter σ in the normal distribution $N(0, \sigma^2)$, where σ is set to 0.1, 0.7, and 1 for high, medium, and low entropy, respectively. This setup allows for a comprehensive evaluation of DSNGD's capabilities and potential limitations. The experimental procedure involves creating a set of problems under each scenario, generating synthetic datasets of length 10^7, and applying several optimization algorithms to estimate the parameters that generated the data. The algorithms evaluated include SGD, AdaGrad [20], SNGD, and DSNGD. The learning rate is $\gamma_t = \frac{a}{1+bt}$ for some combination of parameters a and b from the set $\{10^i \mid -4 \leq i < 2\}$. We select the learning rate yielding the highest performance on approximately 2.5% of the data. The error is quantified using the Kullback–Leibler (KL) divergence [32] between the distributions of the true parameters and the estimated parameters. Each experiment is

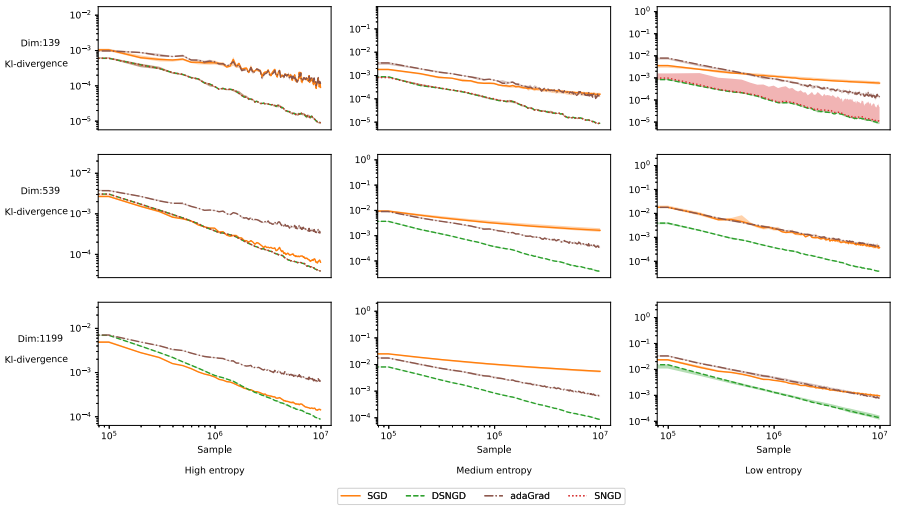


Fig. 2 KL-Divergence of SGD, DSNGD and adaGrad estimates along training processes for every scenario

repeated 10 times for every scenario, with the results summarized using the median and interquartile range.

In Fig. 2, the results of the experiments are presented, showing the performance of the optimization algorithms across three entropy scenarios (low, medium, and high) combined with three manifold dimensions (139, 539, and 1199). The y-axis represents the Kullback–Leibler divergence (KL-error) [32], while the x-axis indicates the training progression in terms of samples processed.

Across all configurations, it is evident that DSNGD consistently outperforms the other algorithms, including SGD, AdaGrad, and SNGD. Notably, DSNGD and SNGD achieve the lowest KL-error and best convergence speed in $\mathscr{M}_1$ for the 3 entropy scenarios, highlighting their effectiveness in estimating the true parameters with a higher convergence rate, as observed in its rapid reduction of KL-error compared to SGD and AdaGrad. This improved performance may underscore the advantages of leveraging a natural gradient-based approach in the optimization process. However, SNGD in the low entropy scenario has a high variance, suggesting that in this scenario it encountered some instabilities. For manifold $\mathscr{M}_2$ and $\mathscr{M}_3$ (second and third rows), DSNGD keeps yielding best estimates with the best performance. SNGD fails to estimate parameters for both medium and low entropy scenarios due to numerical instabilities. SNGD is directly excluded from experiments in $\mathscr{M}_3$ (third row) due to its computational demands, showing its limited applicability.

The combined results suggest that DSNGD not only offers better parameter estimation but also converges faster than traditional stochastic gradient-based methods, making it a robust choice for complex optimization tasks. The dominance of natural gradient-based algorithms—DSNGD and SNGD—is evident in terms of convergence speed across various configurations. DSNGD's superior performance in stability

Table 1 Computational time in seconds of every training process for algorithms SGD, AdaGrad, DSNGD and SNGD in manifolds $\mathscr{M}_1$, $\mathscr{M}_2$ and $\mathscr{M}_3$

	$\mathscr{M}_1$	$\mathscr{M}_2$	$\mathscr{M}_3$
SGD	~ 5	~ 10	~ 37
AdaGrad	~ 5	~ 11	~ 38
DSNGD	~ 11	~ 22	~ 72
SNGD	~ 18	~ 267	> 37800

and reliability further establishes its advantage for high-dimensional and complex optimization tasks.

Table 1 presents the computational time of the training process required by each algorithm-SGD, AdaGrad, DSNGD, and SNGD-across three manifolds ($\mathscr{M}_1$, $\mathscr{M}_2$, and $\mathscr{M}_3$). The results highlight the computational efficiency and scalability of the algorithms under varying dimensional complexities.

In the simplest manifold, $\mathscr{M}_1$, all algorithms exhibit low computational times, with SGD and AdaGrad achieving the fastest execution. As the dimensionality increases in $\mathscr{M}_2$, most algorithms double their computational time, except for SNGD that shows a dramatic increase in computational time, requiring $\sim$ 267 s. This emphasizes the scaling challenges of SNGD as dimensions grow, contrasting sharply with DSNGD's more manageable overhead. In the highest dimensional manifold, $\mathscr{M}_3$, the disparity between algorithms becomes pronounced. SGD and AdaGrad maintain efficiency, requiring $\sim 37-38$ s. DSNGD, while more computationally intensive, remains feasible at $\sim$ 72 s evidencing its linear computational complexity. In stark contrast, SNGD becomes computationally prohibitive, exceeding 37,800 s (over 10 h). Recall that we run 10 experiments per entropy scenario, which would add up to 30 experiments of more than 10 h each. This highlights the severe scalability issues associated with SNGD for high-dimensional problems.

7 Conclusion and future work

This study explores the potential of DSNGD as an efficient and robust optimization algorithm. DSNGD approximates efficiently the natural gradient at each step, with theoretical guarantees of convergence in the discrete case. To support this, we proved a general result on the convergence of interior half-space gradient approximations, providing a foundation that could extend to broader algorithmic frameworks.

The experimental results presented in this paper validate DSNGD's practical applicability. DSNGD consistently outperforms traditional algorithms such as SGD and AdaGrad. Natural gradient variants—DSNGD and SNGD—achieve better estimates with improved convergence speed. In contrast to DSNGD, SNGD, while competitive in convergence rates, faces prohibitive computational costs in high-dimensional problems and instability in low-entropy scenarios. These findings highlight the scalability and reliability of DSNGD across diverse problem settings.

For the more theoretical part, in the future we plan to study the convergence of continuous and mixed DSNGD, that is when $\mathscr{X}$ is continuous, and in cases where

$\mathscr{X} = (\mathscr{X}_d, \mathscr{X}_c)$ is divided into a discrete and a continuous part. Looking ahead, we are developing a flexible implementation of DSNGD that can be adapted to different LEFs linked to the conditional distributions $P(\mathscr{X} \mid \mathscr{Y})$. These include commonly used continuous distributions (e.g., normal, Poisson, exponential), as well as discrete and mixed distributions consistent with naive Bayes independence assumptions.

Research to expand DSNGD to nonlinear exponential families remains open. The algorithm strongly relies on two dual parametrizations, which may complicate the task.

In conclusion, DSNGD emerges as a promising solution for scalable and efficient optimization, particularly in high-dimensional and complex parameter spaces. Its theoretical underpinnings and empirical performance position it as a strong candidate for tackling the challenges of modern optimization tasks.

Appendices

A Proof of Proposition 1

Proof Prove first that if the log-odds ratio of $P(\mathscr{Y} \mid \mathscr{X})$ is an affine function of $\mathscr{X}$ then the joint distribution $P(\mathscr{Y}, \mathscr{X})$ belongs to LEF.

According to theorem 2 in [1], assume that $P(\mathscr{X} \mid \mathscr{Y} = i)$ belongs to the same LEF for all $i \in \mathscr{Y}$. Also, since $\mathscr{Y}$ is discrete and finite, $P(\mathscr{Y})$ is a categorical distribution and hence, it belongs to LEF. This means that there exist parameters $\overline{\alpha} \in \mathbb{R}^{s-1}$ and $\overline{\theta_i} \in \mathbb{R}^t$ for all $i \in \mathscr{Y}$ such that

$$\begin{aligned} P_{\overline{\alpha}}(\mathscr{Y} = i) &= \frac{\exp S(i)^\intercal \overline{\alpha}}{\sum_y \exp S(y)^\intercal \overline{\alpha}} \\ P_{\overline{\theta_i}}(x \mid \mathscr{Y} = i) &= \frac{\exp T(x)^\intercal \overline{\theta_i}}{\int_x \exp T(x)^\intercal \overline{\theta_i}} \end{aligned} \tag{65}$$

where S and T are sufficient statistics of $\mathscr{Y}$ and $\mathscr{X}$ respectively. If $\overline{\theta}$ is the matrix having $\overline{\theta_i}$ as i-th row, name $\overline{\eta} = (\overline{\alpha}, \overline{\theta})$ and write

$$\begin{aligned} P_{\overline{\eta}}(\mathscr{Y} = i, x) = P_{\overline{\alpha}}(\mathscr{Y} = i) P_{\overline{\theta_i}}(x \mid \mathscr{Y} = i) &= \frac{\exp S(i)^\intercal \overline{\alpha}}{\sum_y \exp S(y)^\intercal \overline{\alpha}} \frac{\exp T(x)^\intercal \overline{\theta_i}}{\int_x \exp T(x)^\intercal \overline{\theta_i}} \\ &= \frac{\exp S(i)^\intercal \overline{\alpha} + T(x)^\intercal \overline{\theta_i}}{\sum_y \exp S(y)^\intercal \overline{\alpha} \int_x \exp T(x)^\intercal \overline{\theta_i}} \end{aligned} \tag{66}$$

To prove the result, it is enough to find a change of variables from $\overline{\eta} = (\overline{\alpha}, \overline{\theta})$ to $\eta = (\alpha, \beta)$ satisfying $P_{\overline{\eta}}(y, x) = P_\eta(y, x)$ where

$$P_\eta(\mathscr{Y} = i, x) = \frac{\exp S(i)^\intercal \alpha + T(x)^\intercal \beta_i}{\int_x \sum_y \exp S(y)^\intercal \alpha + T(x)^\intercal \beta_y} \tag{67}$$

since η is the natural parametrization of a LEF.

In particular, the change of variables has to satisfy that $P_{\overline{\eta}}(x \mid \mathscr{Y} = i) = P_\eta(x \mid \mathscr{Y} = i)$ and $P_{\overline{\eta}}(y) = P_\eta(y)$. Start with the conditional probability and observe that

$$\begin{aligned} P_\eta(x \mid \mathscr{Y} = i) = \frac{P_\eta(\mathscr{Y} = i, x)}{\int_x P_\eta(\mathscr{Y} = i, x)} &= \frac{\exp S(i)^\intercal \alpha + T(x)^\intercal \beta_i}{\int_x \exp S(i)^\intercal \alpha + T(x)^\intercal \beta_i} \\ &= \frac{\exp T(x)^\intercal \beta_i}{\int_x \exp T(x)^\intercal \beta_i} \end{aligned} \tag{68}$$

Last equation matches exactly with Eq. (65) by just setting $\beta = \overline{\theta}$. To complete the change of variables continue by matching $P_{\overline{\eta}}(y) = P_\eta(y)$.

$$\begin{aligned} P_\eta(\mathscr{Y} = i) &= \frac{\int_x \exp S(i)^\intercal \alpha + T(x)^\intercal \beta_i}{\sum_j \int_x \exp S(j)^\intercal \alpha + T(x)^\intercal \beta_j} \\ &= \frac{\exp\left(S(i)^\intercal \alpha\right) \int_x \exp T(x)^\intercal \beta_i}{\sum_j \exp\left(S(j)^\intercal \alpha\right) \int_x \exp T(x)^\intercal \beta_j} \\ &= \frac{\exp S(i)^\intercal \alpha + \log A_i}{\sum_j \exp S(j)^\intercal \alpha + \log A_j} \end{aligned} \tag{69}$$

where $A_i = \int_x \exp T(x)^\intercal \beta_i$. Last equation must coincide with Eq. (65). That is

$$P_\eta(\mathscr{Y} = i) = P_{\overline{\eta}}(\mathscr{Y} = i) \iff \frac{\exp S(i)^\intercal \alpha + \log A_i}{\sum_j \exp S(j)^\intercal \alpha + \log A_j} = \frac{\exp S(i)^\intercal \overline{\alpha}}{\sum_y \exp S(y)^\intercal \overline{\alpha}} \tag{70}$$

To simplify, assume S is canonical. That is $S(i) = e_i$ is the i-th canonical vector for all $i \neq s$ and $S(s) = 0 \in \mathbb{R}^{s-1}$. Note that it is enough to prove that there exists a $\mu \in \mathbb{R}$ such that

$$S(i)^\intercal \alpha + \log A_i - \mu = S(i)^\intercal \overline{\alpha}, \ \forall i \in \mathscr{Y} \tag{71}$$

because as a consequence, Eq. (70) clearly holds. In our case, it is $S(i)^\intercal \alpha = \alpha_i$ when $i \neq s$ and $S(i)^\intercal \alpha = 0$, and therefore the solution is

$$\begin{aligned} \alpha + \begin{pmatrix} \log A_1 \\ \vdots \\ \log A_{s-1} \end{pmatrix} - \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \mu &= \overline{\alpha} \\ \mu &= \log A_s \end{aligned} \tag{72}$$

and the proof is completed when S is canonical.

Prove now the result for a general minimal sufficient statistic S. Equation (71) describes the below linear equations system

$$\begin{aligned} \mathbf{S}\alpha + \begin{pmatrix} \log A_1 \\ \vdots \\ \log A_{s-1} \end{pmatrix} - \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \mu &= \mathbf{S}\overline{\alpha} \\ S(s)^\intercal \alpha + \log A_s - \mu &= S(s)^\intercal \overline{\alpha} \end{aligned} \tag{73}$$

where $\mathbf{S}$ is the matrix having $S(1), \ldots, S(s-1)$ as rows. Since S is a minimal sufficient statistic, assume without loss of generality that $S(1), \ldots, S(s-1)$ are linearly independent vectors, and then $\mathbf{S}$ is invertible. Finally, it is easy to check that the change of variables is

$$\begin{aligned} \alpha + \mathbf{S}^{-1} \left(\begin{pmatrix} \log A_1 \\ \vdots \\ \log A_{s-1} \end{pmatrix} - \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \mu \right) &= \overline{\alpha} \\ \mu = \frac{S(s)^\intercal \mathbf{S}^{-1} \begin{pmatrix} \log A_1 \\ \vdots \\ \log A_{s-1} \end{pmatrix} - \log A_s}{S(s)^\intercal \mathbf{S}^{-1} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - 1} & \end{aligned} \tag{74}$$

The converse implication is straightforward. Assuming that $P(\mathscr{X}, \mathscr{Y})$ belongs to LEF, and therefore assuming Eq. (67), start by expressing the conditional probability distributions of $\mathscr{Y}$ given $\mathscr{X}$ in η.

$$\begin{aligned} P_\eta(y \mid x) &= \frac{P_\eta(y, x)}{\sum_y P_\eta(y, x)} \\ &= \frac{\exp S(y)^\intercal \alpha + T(x)^\intercal \beta_y}{\sum_y \exp S(y)^\intercal \alpha + T(x)^\intercal \beta_y} \end{aligned} \tag{75}$$

and compute the log-odds ratio

$$\begin{aligned} \log \frac{P_\eta(x \mid \mathscr{Y} = k)}{P_\eta(x \mid \mathscr{Y} = h)} &= S(k)^\intercal \alpha + T(x)^\intercal \beta_k - (S(h)^\intercal \alpha + T(x)^\intercal \beta_h) \\ &= (S(k) - S(h))^\intercal \alpha + T(x)^\intercal (\beta_k - \beta_h) \end{aligned} \tag{76}$$

which is clearly an affine function of features $\mathscr{X}$. □

B Proof of Proposition 3

Proof To proof the result we have to effectively express

$$h(x,\zeta) = (\log P_{\zeta^*}(\mathscr{Y}=1,x), \dots \log P_{\zeta^*}(\mathscr{Y}=s,x)) \tag{77}$$

in the dual parameterization $\zeta^* = (\alpha^*, \beta^*)$. According to Eq. (16), and assuming that S and T are canonical statistics, deduce that

$$\log P_{\zeta^*}(y,x) = \begin{cases} \log(\beta^*_y)_x & x \neq m \\ \log(\alpha^*_y - |\beta^*_y|) & x = m,\ y \neq s \\ \log(1 - |\alpha^*| - |\beta^*_s|) & x = m,\ y = s \end{cases} \tag{78}$$

where $|v|$ with v a vector refers to the sum of all the vector components.

Continue by proving first term of the proposition, that is

$$\nabla_{\alpha^*} h(x,\zeta) \cdot q(y,x,P_\eta) = \mathbb{1}_{x=m} \cdot K_s \cdot D(x,\zeta^*) \cdot q(y,x,P_\eta) \tag{79}$$

Now, apply the derivative to each term of h with respect to α^*

$$\begin{aligned} \nabla_{\alpha^*} \log P_{\zeta^*}(y,x) &= \begin{cases} 0 & x \neq m \\ \frac{e_{s-1}(y)}{\alpha^*_y - |\beta^*_y|} & x = m,\ y \neq s \\ \frac{-\mathbf{1}_{s-1}}{1-|\alpha^*|-|\beta^*_s|} & x = m,\ y = s \end{cases} \\ &= \begin{cases} 0 & x \neq m \\ \frac{e_{s-1}(y)}{P_{\zeta^*}(y,m)} & x = m,\ y \neq s \\ \frac{-\mathbf{1}_{s-1}}{P_{\zeta^*}(s,m)} & x = m,\ y = s \end{cases} \end{aligned} \tag{80}$$

with $\mathbf{1}_n$ being the n dimensional vector filled with ones. Observe that last equation already proves what we wanted.

Proceed now to prove the second term

$$\nabla_{\beta^*} h(x,\zeta^*) \cdot q(y,x,P_\eta) = K_m \cdot e_m(x) \cdot [D(x,\zeta^*) \cdot q(y,x,P_\eta)]^T \tag{81}$$

Apply the derivative to each term of h with respect to β^*

$$\nabla_{\beta^*} \log P_{\zeta^*}(y,x) = \begin{cases} \frac{e_{m-1}(x) \cdot e_s(y)^\intercal}{P_{\zeta^*}(y,x)} & x \neq m \\ \frac{-\mathbf{1}_m \cdot e_s(y)^\intercal}{P_{\zeta^*}(y,m)} & x = m \end{cases} \tag{82}$$

Observe $\nabla_{\beta^*} h$ is expressed respecting the dimension of de parameters matrix β^*. However, the product of $\nabla_{\beta^*} h \cdot q$ demands for $\nabla_{\beta^*} h$ to be expressed as a column vector. Hence we have

$$\nabla_{\beta^*} h(x, \zeta^*) = \begin{pmatrix} [K_m \cdot e_m(x)] & & 0 \\ & \ddots & \\ 0 & & [K_m \cdot e_m(x)] \end{pmatrix} \cdot D(x, \zeta^*) \tag{83}$$

Finally, multiply the matrix by $q(y, x, P_\eta)$ and re-arrange terms into a matrix with same shape of β^*. This matches exactly with what we aim at proving and the proof is finished. □

C Proof of Theorem 4

Proof The proof uses Robbins–Siegmund theorem as key tool. Steps taken are closely inspired by those taken in the proof of Theorem 3.2 in [40].

Compute Taylor' second order approximation of $l(\eta_{t+1})$, and after condition **C.2** apply Taylor's inequality

$$\mathbb{L}(\eta_{t+1}) \leq \mathbb{L}(\eta_t) - \gamma_t \nabla \mathbb{L}(\eta_t)^T Y_t + \gamma_t^2 K \|Y_t\|^2 \tag{84}$$

Therefore, applying the expectation conditioned to information at time t obtain

$$\mathbb{E}_t[\mathbb{L}(\eta_{t+1})] \leq \mathbb{L}(\eta_t) - \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] + \gamma_t^2 K \mathbb{E}_t \|Y_t\|^2 \tag{85}$$

Use bound of **C.4** to third term of right hand side

$$\mathbb{E}_t[\mathbb{L}(\eta_{t+1})] \leq \mathbb{L}(\eta_t) - \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] + \gamma_t^2 K (A + Bl(\eta_t)) \tag{86}$$

Finally, substitute $U_t = \mathbb{L}(\eta_t)$ and arrange terms to match with Eq. (11)

$$\mathbb{E}_t[U_{t+1}] \leq (1 + B\gamma_t^2 K) U_t - \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] + \gamma_t^2 K A \tag{87}$$

Note that Theorem 2 conditions are satisfied, since condition **C.6** implies $\sum_t \beta_t = \sum_t BK\gamma^2 = BK \sum_t \gamma^2 < \infty$ and $\sum_t \varepsilon_t = \sum_t KA\gamma^2 < \infty$. Hence, Robbins–Siegmund theorem ensures that $U_t = \mathbb{L}(\eta_t)$ converges almost surely to a random variable and

$$\sum_t \zeta_t = \sum_t \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] < \infty \tag{88}$$

Now prove that $\lim_t \mathbb{L}(\eta_t) = \mathbb{L}(\overline{\eta})$. If $\mathbb{L}(\eta_t)$ converges to some different random variable, condition **C.3**, second condition of **C.6** and Eq. (88) lead to a contradiction. Indeed, if $\lim_t \mathbb{L}(\eta_t) = v \neq \mathbb{L}(\overline{\eta})$, use condition **C.3** and deduce that for a fixed $0 < \delta < v - \mathbb{L}(\overline{\eta})$ there exists an N large enough and $\varepsilon > 0$ such that

$$\nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] \geq \varepsilon \tag{89}$$

for all $t > N$. Therefore, Eq. (88) becomes

$$\begin{aligned}\sum_t \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] &= \sum_t^N \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] + \sum_{t>N} \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] \\ &\geq \sum_t^N \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] + \sum_{t>N} \varepsilon \gamma_t \\ &\geq \varepsilon \sum_{t>N} \gamma_t\end{aligned} \tag{90}$$

Second condition in **C.6** applied to right hand side of above equation assures that

$$\sum_t \gamma_t \nabla \mathbb{L}(\eta_t)^T \mathbb{E}_t[Y_t] = \infty \tag{91}$$

which contradicts Eq. (88).

Finally, it is only possible that $\lim_t \mathbb{L}(\eta_t) = \mathbb{L}(\overline{\eta})$ almost surely as we wanted to prove. □

Author Contributions The authors have contributed equally to this work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The work has been funded by EU Horizon 2020 under grant agreements 872944 (Crowd4SDG) and 825619 (Humane-AI-net), and by the Spanish Ministry of Science and Innovation through the CI-SUSTAIN project (PID2019-104156GB-I00).

Data availability Not applicable.

Code availability https://github.com/Kissyfur/dsngd [42].

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Banerjee, A.: An analysis of logistic models: exponential family connections and online performance. In: Proceedings of the 2007 SIAM International Conference on Data Mining, Proceedings, pp. 204–215.

Society for Industrial and Applied Mathematics (2007). https://doi.org/10.1137/1.9781611972771.19. https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.19
2. Li, J., Bioucas-Dias, J.M., Plaza, A.: Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. IEEE Trans. Geosci. Remote Sens. **50**(3), 809–823 (2012). https://doi.org/10.1109/TGRS.2011.2162649
3. Covington, P., Adams, J., Sargin, E.: Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, pp. 191–198. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2959100.2959190
4. Daniels, M.J., Gatsonis, C.: Hierarchical polytomous regression models with applications to health services research. Stat. Med. **16**(20), 2311–2325 (1997)
5. Bull, S.B., Lewinger, J.P., Lee, S.S.: Confidence intervals for multinomial logistic regression in sparse data. Stat. Med. **26**(4), 903–918 (2007)
6. Biesheuvel, C., Vergouwe, Y., Steyerberg, E., Grobbee, D., Moons, K.: Polytomous logistic regression analysis could be applied more often in diagnostic research. J. Clin. Epidemiol. **61**(2), 125–134 (2008)
7. Leppink, J.: Multicategory nominal choices. In: The Art of Modelling the Learning Process, pp. 103–110. Springer (2020)
8. Ben-Akiva, M.E., Lerman, S.R., Lerman, S.R.: Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press (1985). Google-Books-ID: oLC6ZYPs9UoC
9. Tadei, R., Perboli, G., Manerba, D.: A recent approach to derive the multinomial logit model for choice probability. In: Daniele, P., Scrimali, L. (eds.) New Trends in Emerging Complex Real Life Problems: ODS, Taormina, Italy, September 10–13, 2018, AIRO Springer Series, pp. 473–481. Springer International Publishing, Cham (2018)
10. Nock, R., Nielsen, F.: On the efficient minimization of classification calibrated surrogates. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 21, pp. 1201–1208. Curran Associates, Inc. (2009)
11. Reid, M.D., Williamson, R.C.: Composite Binary Losses. J. Mach. Learn. Res. **11**(83), 2387–2422 (2010)
12. Vernet, E., Reid, M.D., Williamson, R.C.: Composite multiclass losses. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 24, pp. 1224–1232. Curran Associates, Inc. (2011)
13. Nock, R., Menon, A.K.: Supervised Learning: No Loss No Cry. arXiv:2002.03555 [cs, stat] (2020)
14. Vapnik, V.: Principles of risk minimization for learning theory. In: Proceedings of the 4th International Conference on Neural Information Processing Systems, NIPS'91, pp. 831–838. Morgan Kaufmann Publishers Inc., San Francisco (1991)
15. Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) Online Learning and Neural Networks. Cambridge University Press, Cambridge (1998). http://leon.bottou.org/papers/bottou-98x. Revised, Oct 2012
16. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**(4), 1574–1609 (2009). https://doi.org/10.1137/070704277. https://epubs.siam.org/doi/abs/10.1137/070704277
17. Dennis, J.E., Schnabel, R.B.: Numerical methods for unconstrained optimization and nonlinear equations. Classics in Applied Mathematics, vol. 16. SIAM, Philadelphia (1996). OCLC: 845110213
18. Natural Gradient Works Efficiently in Learning: Amari, S.i. Neural Comput. **276**, 251–276 (1998)
19. Becker, S., Lecun, Y.: Improving the convergence of back-propagation learning with second-order methods. In: Touretzky, D., Hinton, G., Sejnowski, T. (eds.) Proceedings of the 1988 Connectionist Models Summer School, San Mateo, pp. 29–37. Morgan Kaufmann (1989)
20. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**(null), 2121–2159 (2011)
21. Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method. arXiv:1212.5701 [cs]. http://arxiv.org/abs/1212.5701 (2012)
22. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
23. Kingma, D.P., Ba, L.J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015). https://dare.uva.nl/search?identifier=a20791d3-1aff-464a-8544-268383c33a75
24. Hu, J., Liu, X., Wen, Z.W., Yuan, Y.X.: A brief introduction to manifold optimization. J. Oper. Res. Soc. China **8**(2), 199–248 (2020). https://doi.org/10.1007/s40305-020-00295-9

25. Boumal, N.: An Introduction to Optimization on Smooth Manifolds. Cambridge University Press, Cambridge (2023). https://doi.org/10.1017/9781009166164. https://www.nicolasboumal.net/book
26. Sánchez-López, B., Cerquides, J.: Convergent stochastic almost natural gradient descent. In: Artificial Intelligence Research and Development—Proceedings of the 22nd International Conference of the Catalan Association for Artificial Intelligence, vol. 319, pp. 54–63 (2019)
27. Carmo, M.P.D.: Riemannian geometry, corrected at 14th printing$h2013 edn. Mathematics: Theory and Applications. Birkhäuser, Boston (2013)
28. Murray, M.K., Rice, J.W.: Differential Geometry and Statistics, vol. 48. CRC Press, Boca Raton (1993)
29. Bonnabel, S.: Stochastic gradient descent on Riemannian manifolds. IEEE Trans. Autom. Control **58**(9), 2217–2229 (2013). https://doi.org/10.1109/TAC.2013.2254619. http://arxiv.org/abs/1111.5280. arXiv:1111.5280
30. Thomas, P.S.: Genga: a generalization of natural gradient ascent with positive and negative convergence results. In: 31st International Conference on Machine Learning, ICML 2014, vol. 5, pp. 3533–3541 (2014)
31. Amari, S.I.: Information Geometry and Its Applications, vol. 5416. Springer, Berlin (2016)
32. Nielsen, F.: An elementary introduction to information geometry. arXiv:1808.08271 [cs, math, stat]. http://arxiv.org/abs/1808.08271 (2018)
33. Nemirovskiĭ, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. A Wiley-Interscience publication. Wiley (1983). https://books.google.es/books?id=6ULvAAAAMAAJ
34. Masegosa, A.R.: Stochastic discriminative EM. In: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI'14, pp. 573–582. AUAI Press, Arlington (2014). http://dl.acm.org/citation.cfm?id=3020751.3020811. Event-place: Quebec City, Quebec, Canada
35. Raskutti, G., Mukherjee, S.: The information geometry of mirror descent. IEEE Trans. Inf. Theory **61**(3), 1451–1457 (2015). https://doi.org/10.1109/TIT.2015.2388583
36. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. **31**(3), 167–175 (2003). https://doi.org/10.1016/S0167-6377(02)00231-6. http://www.sciencedirect.com/science/article/pii/S0167637702002316
37. Khan, M.E., Lin, W.: Conjugate-computation variational inference: converting variational inference in non-conjugate models to inferences in conjugate models. CoRR arXiv:1703.04265. http://arxiv.org/abs/1703.04265 (2017)
38. Wani, J.K.: On the linear exponential family. Math. Proc. Camb. Philos. Soc. **64**(2), 481–483 (1968). https://doi.org/10.1017/S0305004100043097
39. Lin, W., Khan, M.E., Schmidt, M.W.: Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In: ICML (2019)
40. Sunehag, P., Trumpf, J., Vishwanathan, S.V.N., Schraudolph, N.: Variable Metric Stochastic Approximation Theory. In: Artificial Intelligence and Statistics, pp. 560–566 (2009). http://proceedings.mlr.press/v5/sunehag09a.html
41. Robbins, H., Siegmund, D.: A convergence theorem for non negative almost supermartingales and some applications. In: Rustagi, J.S. (ed.) Optimizing Methods in Statistics, pp. 233–257. Academic Press, London (1971)
42. Sanchez Lopez, B., Cerquides, J.: dsngd (2022). https://doi.org/10.5281/zenodo.1234. https://github.com/Kissyfur/dsngd