REMARKS ON GEOMETRIC OPTIMIZATION

DANIEL WESER

Abstract. Standard adaptive optimization methods (ADAM, ADAGRAD, AMSGRAD) rely on coordinate-wise scaling that, when applied to constrained settings like the sphere or ellipsoid (RMS-Norm), naturally distort the descent direction in nontrivial ways. The first goal of this note is to unify the Riemannian adaptive optimization methods by deriving general yet explicit update rules, providing a simple exposition of these methods. Then, we use this formulation to analyze the preservation of separation of variables in gradient accumulation, as well as the interaction between curvature and momentum.

Contents

1. Introduction

Optimization on manifolds is a foundational subject in applied mathematics [2], and its application to deep learning requires handling specific computational constraints. Libraries such as Geoopt [3] provide excellent general implementations, but many standard adaptive optimizers appear limited to the sphere, despite the prevalence of anisotropic constraints (RMS-Norm).

Moreover, from a theoretical perspective, even in the case of the sphere the geometry of the submanifold will introduce nontrivial "distortions" in the momentum and velocity arising from normal derivatives of the loss function, and these will in turn alter the effective learning rate.

Therefore, in an effort to better understand these underlying dynamics, this note unifies Riemannian adaptive optimization methods on the sphere $\mathbb{S}^n$ and the ellipsoid $\mathcal{E}$, the latter arising naturally from RMS-Normalization with affine scaling. Starting from the work of Bécigneul and Ganea [1], we derive general yet explicit update rules that highlight the way in which the separation of variables occurs in gradient accumulation. This facilitates our analysis of the interaction between the underlying geometry and the chosen optimization method, as well as allowing direct comparisons between the two geometric settings. Finally, we briefly note some further applications.

2. Optimization on the sphere

We begin with the simplest example: the sphere $\mathbb{S}^n \subset \mathbb{R}^{n+1}$. To quickly fix our notation, we will let D denote standard Euclidean differentiation and ∇ differentiation on $\mathbb{S}^n$. Additionally, for a point $x \in \mathbb{S}^n$ and a vector $v \in \mathbb{R}^{n+1}$, we will denote by $\mathrm{P}_x[v]$ the projection of v onto $\mathrm{T}_x\mathbb{S}^n$.

Date: February 5, 2026.

Our general methodology will repeatedly use the following facts: the sphere is a level set of the function $x \mapsto \frac{x}{|x|}$, any tangent vector on $\mathbb{S}^n$ can be written with respect to the standard basis in $\mathbb{R}^{n+1}$, and the exponential map on $\mathbb{S}^n$ is explicitly given.

2.1. **Preliminaries.** To begin, we recall that for a level set of a function $F : \mathbb{R}^{n+1} \to \mathbb{R}$, the normal vector to the level set at $F(x)$ is given by the gradient $DF(x)$. Since the sphere corresponds to the function $F(x) = |x|$, we have that $DF(x) = x$. Thus, the projection of any vector $w \in \mathbb{R}^{n+1}$ onto $\mathrm{T}_x\mathbb{S}^n$ is given by

$$\mathrm{P}_x[w] \;=\; w - \langle w, x\rangle\, x\,, \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product. In particular, if $f : \mathbb{R}^{n+1} \to \mathbb{R}$, then

$$\nabla f \;=\; Df - \langle Df, x\rangle\, x\,. \tag{2.2}$$

Remark. If $f : \mathbb{S}^n \to \mathbb{R}$ is only defined on the sphere, then one first defines the zero-homogeneous extension of f,

$$\hat{f}(x) \;:=\; f\Big(\frac{x}{|x|}\Big)\,, \quad x \in \mathbb{R}^{n+1}\,,$$

and then applies the identity

$$\nabla f \;=\; \big(D\hat{f}\,\big)\Big|_{\mathbb{S}^n}\,.$$

The last ingredient we will need is the exponential map on the sphere. It is conveniently given by the following formula: for $v \in \mathrm{T}_x\mathbb{S}^n$, one has

$$\exp_x(tv) \;=\; \cos\big(t|v|\big)\, x \;+\; \sin\big(t|v|\big)\,\frac{v}{|v|}\,. \tag{2.3}$$

We are now ready to write our general framework for the optimization methods ADAGRAD, ADAM, and AMSGRAD. However, as the very first step, one must make a choice of which gradient to transform to obtain the tangent direction for the update step. Indeed, one may either transform

- the *Euclidean* gradient Df or
- the *tangential* gradient ∇f

and then project back onto the tangent space of $\mathbb{S}^n$. We will consider both cases.

2.2. **Euclidean gradient.** In this simpler case, the "separation of variables" is completely preserved at the very first step, since the Euclidean gradient Df is of course independent of sphere. This is certainly one argument in favor of this starting point. At the same time, this can be seen as a drawback, because this causes the underlying geometry to be completely absent until the very last step.

To begin, we will denote by $f_i = \partial_i f$ the components of Df, so that

$$Df = \sum_{i=1}^{n+1} f_i e_i\,,$$

where $\{e_i\}_{i=1}^{n+1}$ is the standard basis for $\mathbb{R}^{n+1}$. Next, for ADAGRAD, ADAM, and AMSGRAD, one simply applies an affine transformation to each component f_i, so that, if we denote this transformation by

$$\tilde{f}_i \;:=\; \beta_i f_i + \sigma_i\,,$$

then we yield the compact expression

$$\begin{aligned}\tilde{D}f \;&:=\; \sum_{i=1}^{n+1} \tilde{f}_i e_i \\ &=\; \beta \odot Df + \sigma\,.\end{aligned}$$

Here, $\beta = (\beta_1, ..., \beta_n)$ and $\sigma = (\sigma_1, ..., \sigma_n)$ denote the vectors of these coefficients and $\odot$ the Hadamard product, and we note that we are introducing the symbol $\tilde{D}$ to distinguish the fact that this a *transformed gradient*, not the gradient of a transformed function.

Next, we apply the projection (2.1) to obtain the transformed tangential gradient

$$\begin{aligned}\tilde{\nabla} f(x) \;&:=\; \mathrm{P}_x\big[\tilde{D}f(x)\big] \\ &=\; \beta \odot Df - \langle \beta \odot Df, x\rangle x \;+\; \sigma - \langle \sigma, x\rangle x\,. \end{aligned} \tag{2.4}$$

To conclude, we insert this expression into (2.3) to obtain the update step

$$x \;\leftarrow\; \exp_x\big(-\alpha\,\tilde{\nabla} f(x)\big) \;=\; \cos\big(\alpha|\tilde{\nabla} f(x)|\big)\,x \;-\; \sin\big(\alpha|\tilde{\nabla} f(x)|\big)\frac{\tilde{\nabla} f(x)}{|\tilde{\nabla} f(x)|}\,, \tag{2.5}$$

where we omit the full expression of $\tilde{\nabla} f$ for readability. The full algorithm is given in Algorithm 1.

Algorithm 1 General optimization algorithm on the sphere with Euclidean gradient

Require: $f : \mathbb{S}^n \to \mathbb{R}$; $x_0 \in \mathbb{S}^n$; $K \in \mathbb{N}$; $\{\beta_k\}_{k=1}^K, \{\sigma_k\}_{k=1}^K \subset \mathbb{R}^n$

$x \leftarrow x_0$

for $k = 1$ to K **do**

$Df \leftarrow Df(x)$

$\beta \leftarrow \beta_k$

$\sigma \leftarrow \sigma_k$

$\tilde{\nabla} f \leftarrow \beta \odot Df - \big\langle \beta \odot Df, x\big\rangle\, x \;+\; \sigma - \langle \sigma, x\rangle x$

$x \leftarrow \cos\big(\alpha|\tilde{\nabla} f(x)|\big)\, x - \sin\big(\alpha|\tilde{\nabla} f(x)|\big)\frac{\tilde{\nabla} f(x)}{|\tilde{\nabla} f(x)|}$

end for

With our initial comments in mind, looking back at the expression for $\tilde{\nabla} f$ in (2.4), the terms $\beta \odot Df$ and σ are completely independent of the sphere, so, while they both do indeed encode information about the change of f, it is only once we project that their relation to the sphere arises.

Moreover, in this simpler setting it is easier to see a potential issue which can arise:

> **Issue:** *Why should $\beta \odot Df$ and σ* **not** *be normal to $\mathbb{S}^n$?*

Equivalently, why should $|\beta \odot Df|$ and $|\sigma|$ not equal 1? If this happens, then we will have $\tilde{\nabla} f = 0$, and the algorithm will terminate prematurely. This is the clearest example of the "distortions" which may arise in the geometric setting, as the momentum and velocity fundamentally interact with the geometry of the submanifold. Moreover, as we will see in Section 2.4 when we outline the choices of β and σ for the methods under consideration, there is no theoretical guarantee for why this should not happen in practice.

2.3. **Tangential gradient.** Starting from the identity (2.2) and following the notation of [1], we can write the components of ∇f with respect to the standard basis on $\mathbb{R}^{n+1}$

$$g_i := \langle \nabla f, e_i \rangle = f_i - \langle Df, x \rangle \, x_i \,, \tag{2.6}$$

so that

$$\nabla f = \sum_i g_i \, e_i \,.$$

Briefly examining (2.6), we have as before that each f_i encodes the change of f independent of $\mathbb{S}^n$, while the second term provides the correction to encode the spherical part of the change of f in x_i. By writing the tangential gradient using ambient coordinates, we may directly see the way in which the "separation of variables" is preserved.

Following the notation of the previous section, we now apply an affine transformation to each component g_i. Indeed, if we denote by

$$\tilde{g}_i \;=\; \beta_i g_i + \sigma_i,$$

then we obtain

$$\begin{aligned} \tilde{D}f \;&=\; \sum_i (\beta_i g_i + \sigma_i) \, e_i \\ &=\; \beta \odot \nabla f \;+\; \sigma \,, \end{aligned}$$

where we again use the symbol $\tilde{D}f$ to represent the fact that not only is this gradient transformed, but it is also no longer tangent.

Our next step is to apply (2.2) to obtain

$$\tilde{\nabla} f = \beta \odot \nabla f - \langle \beta \odot \nabla f, x \rangle x \;+\; \sigma - \langle \sigma, x \rangle x \,, \tag{2.7}$$

which is the same expression as (2.4) except with the tangential gradient ∇f instead of the Euclidean Df. Now, we can expand the terms involving ∇f as

$$\begin{aligned} &\beta \odot \nabla f - \langle \beta \odot \nabla f \,, x \rangle x \\ &=\; \beta \odot \Big(Df - \big\langle Df, x \big\rangle x \Big) - \Big\langle \beta \odot \Big(Df - \big\langle Df, x \big\rangle x \Big), \, x \Big\rangle \, x \\ &=\; \beta \odot Df - \big\langle \beta \odot Df, x \big\rangle \, x \;-\; \big\langle Df, x \big\rangle \Big(\beta \odot x - \big\langle \beta \odot x, x \big\rangle \, x \Big) \,, \end{aligned} \tag{2.8}$$

so that combining (2.7) and (2.8) yields the final expression

$$\tilde{\nabla} f \;=\; \beta \odot Df - \big\langle \beta \odot Df, x \big\rangle \, x \;+\; \sigma - \langle \sigma, x \rangle x \;-\; \big\langle Df, x \big\rangle \Big(\beta \odot x - \big\langle \beta \odot x, x \big\rangle \, x \Big) \,. \tag{2.9}$$

The final step is to again substitute $\tilde{\nabla} f$ into the equation for the exponential map (2.3), yielding the same expression (2.5). The complete algorithm is given in Algorithm 2.

Algorithm 2 General optimization algorithm on the sphere with tangential gradient

Require: $f : \mathbb{S}^n \to \mathbb{R}$; $x_0 \in \mathbb{S}^n$; $K \in \mathbb{N}$; $\{\beta_k\}_{k=1}^K, \{\sigma_k\}_{k=1}^K \subset \mathbb{R}^n$

$x \leftarrow x_0$

for $k = 1$ to K **do**

$Df \;\leftarrow\; Df(x)$

$\beta \;\leftarrow\; \beta_k$

$\sigma \;\leftarrow\; \sigma_k$

$\tilde{\nabla} f \;\leftarrow\; \beta \odot Df - \big\langle \beta \odot Df, x \big\rangle \, x \;-\; \big\langle Df, x \big\rangle \big(\beta \odot x - \big\langle \beta \odot x, x \big\rangle \, x \big) \;+\; \sigma - \langle \sigma, x \rangle x$

$x \;\leftarrow\; \cos\big(\alpha |\tilde{\nabla} f(x)|\big) \, x - \sin\big(\alpha |\tilde{\nabla} f(x)|\big) \frac{\tilde{\nabla} f(x)}{|\tilde{\nabla} f(x)|}$

end for

Looking at (2.9), we note that the first four terms are identical to the Euclidean gradient case. What is new here is the last term, which is (minus) the projection of $\beta \odot x$, weighted by the radial derivative $\langle Df, x\rangle$. This term clearly gives an interaction between the coefficients β and the points on the sphere x, but its full meaning is not particularly clear. Looking ahead, each β will ultimately be a function of the derivatives of f, so we can perhaps think of $\beta \odot x$ as a sort of "projection" of the derivatives of f onto the normal vector x, almost like a nonlinear radial derivative, but which we then project back onto the tangent space.

2.4. **Application to ADAGRAD, ADAM, and AMSGRAD.** We now quickly expand these optimization methods in our above notation, following the exposition of [1, Sec. 2.3]. Here, we will be loose with our notation, letting g_i denote the components of the gradient of f for either the Euclidean or tangential gradient. We note that one may even mix-and-match: looking for example at ADAGRAD, one may use the Euclidean gradient $g_i(t_k) = f_i(t_k)$ in the numerator but the prior tangential gradients $g_i(t_l)$ in the denominator.

ADAGRAD. This is the simplest method, since one has $\sigma \equiv 0$. Indeed, one sets

$$\tilde{g}_i(t_k) \;=\; g_i(t_k) \Big/ \sqrt{\sum_{l=1}^{k} g_i(t_l)^2}\,,$$

which corresponds to

$$\beta_i(t_k) \;=\; \left(\sum_{l=1}^{k} g_i(t_l)^2\right)^{-1/2} \qquad \text{and} \qquad \sigma_i(t_k) \;=\; 0\,.$$

ADAM. For ADAM, one sets

$$\tilde{g}_i(t_k) \;=\; m_i(t_k)/\sqrt{v_i(t_k)}\,,$$

where

$$m_i(t_k) \;=\; \gamma_1\, m_i(t_{k-1}) + (1-\gamma_1)\, g_i(t_k) \qquad \text{and} \tag{2.10}$$

$$v_i(t_k) \;=\; \gamma_2\, v_i(t_{k-1}) + (1-\gamma_2)\, g_i(t_k)^2\,, \tag{2.11}$$

which yields the coefficients

$$\beta_i(t_k) \;=\; (1-\gamma_1) \Big/ \sqrt{\gamma_2\, v_i(t_{k-1}) + (1-\gamma_2)\, g_i(t_k)^2}$$

and

$$\sigma_i(t_k) \;=\; \gamma_1\, m_i(t_{k-1}) \Big/ \sqrt{\gamma_2\, v_i(t_{k-1}) + (1-\gamma_2)\, g_i(t_k)^2}\,.$$

AMSGRAD. Finally, for AMSGRAD, one slightly modifies ADAM by instead taking

$$\tilde{g}_i(t_k) \;=\; m_i(t_k)/\sqrt{\hat{v}_i(t_k)}\,,$$

where

$$\hat{v}_i(t_k) \;=\; \max\big\{v_i(t_k),\, \hat{v}_i(t_{k-1})\big\}$$

and $m_i(t_k)$ and $v_i(t_k)$ are defined exactly as in (2.10) and (2.11), respectively. Due to the presence of the maximum in the definition of $\hat{v}_i(t_k)$, we will simply write these coefficients to be

$$\beta_i(t_k) \;=\; (1-\gamma_1) \Big/ \sqrt{\hat{v}_i(t_k)} \qquad \text{and} \qquad \sigma_i(t_k) \;=\; \gamma_1\, m_i(t_{k-1}) \Big/ \sqrt{\hat{v}_i(t_k)}\,,$$

which we hope is clear to the reader.

3. Optimization on the ellipsoid

3.1. **Preliminaries.** We can easily extend our framework from the sphere to the ellipsoid. Our motivation for this extension is RMS-normalization [4], where one allows for weights in the computation of the ℓ^2-norm of a vector. To fix our notation, given $c = (c_1, ..., c_{n+1}) \in \mathbb{R}^{n+1}$ with $c_i > 0$, we define the anisotropic ℓ^2-norm induced by c through

$$\|x\|_c^2 \;:=\; \sum_{i=1}^{n+1} c_i x_i^2 \,,$$

which may equivalently be written as the inner product

$$\|x\|_c^2 \;=\; \langle c \odot x, x \rangle \,.$$

With this in hand, we will denote the ellipsoid induced by c as

$$\mathcal{E}_c := \Big\{ \|x\|_c = 1 \Big\} \,.$$

As in the previous section, we utilize the fact that $\mathcal{E}_c$ is a level set of the function $F(x) = \|x\|_c$, so that the normal to a point $x \in \mathcal{E}_c$ is given by

$$DF(x) \;=\; \frac{c \odot x}{\|x\|_c} \;=\; c \odot x \,.$$

The only difference in this case is that in order to write the projection we need to normalize this expression using the Euclidean norm. We therefore obtain the *unit* normal vector $n(x)$ to be

$$n(x) \;:=\; \frac{c \odot x}{|c \odot x|},$$

where $|c \odot x|$ is the Euclidean norm of the vector $c \odot x$. Note that $|c \odot x| \neq \|x\|_c$ unless $c = (1, ..., 1)$. Now that we have the unit normal, we find that the projection $\mathrm{P}_x[v]$ of a vector $v \in \mathbb{R}^{n+1}$ onto $\mathrm{T}_x\mathcal{E}_c$ is given by

$$\mathrm{P}_x[v] \;=\; v - \langle v, c \odot x \rangle \frac{c \odot x}{|c \odot x|^2} \,. \tag{3.1}$$

Unfortunately for the ellipsoid, there is not a closed form equation for the exponential map, as we had in the spherical case. One may easily derive the geodesic equation to be

$$\begin{aligned} \ddot{x} \;&=\; -\frac{\langle c \odot \dot{x}, \dot{x} \rangle}{\langle c \odot x, c \odot x \rangle} \, c \odot x \\ &=\; -\frac{\|\dot{x}\|_c^2}{|c \odot x|^2} \, c \odot x \,, \end{aligned}$$

but integrating this equation is computationally prohibitive in high-dimensional learning loops. One can instead use a retraction map $R_x : \mathrm{T}_x\mathcal{E}_c \to \mathcal{E}_c$, and the simplest choice is the metric projection

$$R_x(v) = \frac{x + v}{\|x + v\|_c} \,, \tag{3.2}$$

which provides a first-order approximation of the exponential map.

3.2. **General methods.** We may now combine (3.1) with our derivations of (2.4) and (2.9) to produce a general framework for optimization on the ellipsoid. Beginning with the simpler case of the **Euclidean gradient**, we obtain the following expression for the transformed gradient:

$$\tilde{\nabla} f(x) \;=\; \beta \odot Df - \big\langle \beta \odot Df, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \;+\; \sigma - \big\langle \sigma, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \,.$$

Here, we can explicitly see the way in which the geometry of the ellipsoid affects the resulting gradient, as it alters the spherical case (2.4) through its direct weighting of different directions (i.e. anisotropy).

To further expand upon this difference, consider the case when there exists a point $x \in \mathbb{S}^n \cap \mathcal{E}_c$, and let us denote at this point x the transformed gradient on the sphere (2.4) as $\tilde{\nabla}_{\mathbb{S}^n} f(x)$, while for the ellipsoid (2.4) we analogously write $\tilde{\nabla}_{\mathcal{E}_c} f(x)$. Then, subtracting these two expressions we obtain

$$\tilde{\nabla}_{\mathcal{E}_c} f(x) - \tilde{\nabla}_{\mathbb{S}^n} f(x) \;=\; \big\langle \beta \odot Df, x \big\rangle x - \big\langle \beta \odot Df, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} + \big\langle \sigma, x \big\rangle x - \big\langle \sigma, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \,,$$

which is the sum of the differences between the radial and anisotropic projections of $\beta \odot Df$ and σ. The magnitude of this quantity gives a measure of the dissimilarity between the two cases.

Finally, for completeness, we note that the equivalent expression of (2.9) on the ellipsoid is given by

$$\begin{aligned} \tilde{\nabla} f \;=\; & \beta \odot Df - \big\langle \beta \odot Df, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \;+\; \sigma - \big\langle \sigma, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \\ & - \frac{\langle Df, c \odot x \rangle}{|c \odot x|^2} \left(\beta \odot c \odot x - \big\langle \beta \odot c \odot x, c \odot x \big\rangle \frac{c \odot x}{|c \odot x|^2} \right) . \end{aligned}$$

4. Further applications

We briefly remark here that the methods we have used above may be extended to *any* submanifold which is the level-set of a smooth function.[1] Indeed, as we noted before, the normal vector is always given by the gradient of the function, thus yielding the normal vector which in turn yields the projection onto the tangent space. The only uncertain component will then be the choice of retraction map, but, under reasonable assumptions, one may use the first-order approximation as in (3.2).

Additionally, we note that the spherical case can be extended to a much broader class of geometries, for example cylinders and cones. More generally, any (warped-)product manifold having the sphere as one factor will only require one to analyze the other factors, as our above analysis will apply with only minor modifications (if any).

Finally, our interest in considering the above extensions is that they encode a non-linear "relationship" between the variables while still allowing for a "separation of variables" due to the ambient Euclidean space. This appears quite natural and rich, and we hope it may be of some future use.

[1]The only restriction needed in order to directly adapt our methods is that the level-set must be a hypersurface. If it is lower dimensional, as may arise for example with algebraic varieties, then the normal space will no longer be one dimensional, introducing more computational complexity.

References

[1] Bécigneul, Gary, and Octavian-Eugen Ganea. "Riemannian adaptive optimization methods." arXiv preprint *arXiv:1810.00760* (2018).

[2] Absil, P-A., Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, 2008.

[3] Kochurov, Max, et al. "Geoopt: Riemannian optimization in pytorch." arXiv preprint *arXiv:2005.12661* (2020).

[4] Zhang, Biao, and Rico Sennrich. "Root mean square layer normalization." *Advances in Neural Information Processing Systems* 32 (2019).