

Derivative of the first order SVD

Randolf Scholz

May 15, 2023

Consider computing the first order SVD expansion. By the Eckart–Young–Mirsky theorem, this is equivalent to solving

$$\underset{\sigma, u, v}{\text{minimize}} \frac{1}{2} \|A - \sigma uv^\top\|_F^2 \quad \text{s.t.} \quad \|u\| = 1 \quad \text{and} \quad \|v\| = 1 \quad \text{and} \quad \sigma \geq 0$$

Equivalently this may be formalized as

$$\sigma = \max_{u, v} u^\top A v \quad \text{s.t.} \quad \|u\| = 1 \quad \text{and} \quad \|v\| = 1$$

Which is a non-convex quadratically constrained quadratic program (QCQP)

$$\sigma = \max_{u, v} \frac{1}{2} \begin{bmatrix} u \\ v \end{bmatrix}^\top \cdot \begin{bmatrix} \mathbf{0}_{m \times m} & A \\ A^\top & \mathbf{0}_{n \times n} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix} u \\ v \end{bmatrix}^\top \cdot \begin{bmatrix} \mathbb{I}_m & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times m} & \mathbb{I}_n \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = 1$$

$$\begin{bmatrix} u \\ v \end{bmatrix}^\top \cdot \begin{bmatrix} \mathbf{0}_{m \times m} & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times m} & \mathbb{I}_n \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = 1$$

The Jacobian and Lagrangian The derivative of the objective function is

$$\mathbf{J}_f(A, \begin{bmatrix} \sigma \\ u \\ v \end{bmatrix}) = \begin{bmatrix} A - \sigma uv^\top & \sigma - u^\top A v \\ \sigma^2 u - \sigma A v & \\ \sigma^2 v - \sigma A^\top u & \end{bmatrix} \implies \mathbf{H}_f\left(\begin{bmatrix} \sigma \\ u \\ v \end{bmatrix}\right) = \begin{bmatrix} 1 & 2\sigma u - A v & 2\sigma v - A^\top u \\ -A v & \sigma^2 \mathbb{I}_m & -\sigma A \\ -A^\top u & -\sigma A^\top & \sigma^2 \mathbb{I}_n \end{bmatrix}$$

Consider the function

$$f(A, \begin{bmatrix} \sigma \\ u \\ v \end{bmatrix}) = \begin{pmatrix} \sigma - u^\top A v \\ \sigma^2 u - \sigma A v \\ \sigma^2 v - \sigma A^\top u \end{pmatrix} \equiv \mathbf{0} \implies \mathbf{J}_f(A, \begin{bmatrix} \sigma \\ u \\ v \end{bmatrix}) = \begin{bmatrix} -\xi v u^\top & 1 & 2\sigma u - A v & 2\sigma v - A^\top u \\ -\sigma v \phi^\top & -A v & \sigma^2 \mathbb{I}_m & -\sigma A \\ -\sigma u \psi^\top & -A^\top u & -\sigma A^\top & \sigma^2 \mathbb{I}_n \end{bmatrix}$$

Thus, gradient descent schema is

$$\begin{aligned} \sigma' &= \sigma - \eta_\sigma (\sigma - u^\top A v) \\ u' &= u - \eta_u (\sigma^2 u - \sigma A v) \\ v' &= v - \eta_v (\sigma^2 v - \sigma A^\top u) \end{aligned}$$

And the newton step with diagonal approximation of the hessian:

$$\begin{aligned}\sigma' &= \sigma - 1(\sigma - u^\top Av) &= u^\top Av \\ u' &= u - \frac{1}{\sigma^2}(\sigma^2 u - \sigma Av) &= \frac{1}{\sigma} Av \\ v' &= v - \frac{1}{\sigma^2}(\sigma^2 v - \sigma A^\top u) &= \frac{1}{\sigma} A^\top u\end{aligned}$$

1 Analysis of the backward

At the equilibrium point, we have:

$$\sigma = u^\top Av \quad Av = \sigma u \quad A^\top u = \sigma v$$

Note that this states that σ is an eigenvalue:

$$\begin{bmatrix} 0 & A \\ A^\top & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \sigma \begin{bmatrix} u \\ v \end{bmatrix}$$

In particular, Rayleigh iteration could be useful. from this we can derive

$$\Delta\sigma = \Delta u^\top Av + u^\top \Delta Av + u^\top A \Delta v = \Delta u^\top u + u^\top \Delta Av + v^\top \Delta v = u^\top \Delta Av$$

Where in the last step we used $\Delta u \perp u$ and $\Delta v \perp v$, which follows from the side condition. Further we have:

$$\begin{aligned}\Delta\sigma u + \sigma\Delta u &= \Delta Av + A\Delta v \\ \Delta\sigma v + \sigma\Delta v &= \Delta A^\top u + A^\top \Delta u\end{aligned} \iff \underbrace{\begin{bmatrix} \sigma \mathbb{I}_m & -A \\ -A^\top & \sigma \mathbb{I}_n \end{bmatrix}}_{=:K} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \Delta Av - \Delta\sigma u \\ \Delta A^\top u - \Delta\sigma v \end{bmatrix}$$

which allows us to express Δu and Δv in terms of ΔA .

2 The VJP

The last equation allows us to compute the VJP at ease:

$$\begin{aligned}\left\langle \begin{bmatrix} \phi \\ \psi \end{bmatrix} \middle| \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \right\rangle &= \left\langle \begin{bmatrix} \phi \\ \psi \end{bmatrix} \middle| K^{-1} \begin{bmatrix} \Delta Av - \Delta\sigma u \\ \Delta A^\top u - \Delta\sigma v \end{bmatrix} \right\rangle \\ &= \left\langle K^{-\top} \begin{bmatrix} \phi \\ \psi \end{bmatrix} \middle| \begin{bmatrix} \Delta Av - \Delta\sigma u \\ \Delta A^\top u - \Delta\sigma v \end{bmatrix} \right\rangle \\ &= \left\langle \begin{bmatrix} \tilde{\phi} \\ \tilde{\psi} \end{bmatrix} \middle| \begin{bmatrix} \Delta Av - \Delta\sigma u \\ \Delta A^\top u - \Delta\sigma v \end{bmatrix} \right\rangle\end{aligned}$$

Now, we compute the terms individually:

$$\begin{aligned}\langle \tilde{\phi} \mid \Delta Av - \Delta\sigma u \rangle &= \langle \tilde{\phi} v^\top \mid \Delta A \rangle - \langle u^\top \tilde{\phi} \mid \Delta\sigma \rangle \\ &= \langle \tilde{\phi} v^\top \mid \Delta A \rangle - \langle u^\top \tilde{\phi} \mid u^\top \Delta Av \rangle \\ &= \langle (\mathbb{I}_m - uu^\top) \tilde{\phi} v^\top \mid \Delta A \rangle\end{aligned}$$

And for the second term we get

$$\begin{aligned}
\langle \tilde{\psi} \mid \Delta A^\top u - \Delta \sigma v \rangle &= \langle \tilde{\psi} u^\top \mid \Delta A^\top \rangle - \langle v^\top \tilde{\psi} \mid \Delta \sigma \rangle \\
&= \langle u \tilde{\psi}^\top \mid \Delta A \rangle - \langle \tilde{\psi}^\top v \mid u^\top \Delta A v \rangle \\
&= \langle u \tilde{\psi} (\mathbb{I}_n - v v^\top) \mid \Delta A \rangle
\end{aligned}$$

Using the formula for inverting a 2×2 block-matrix, we can give an explicit solution to $K^{-\top} \begin{bmatrix} \phi \\ \psi \end{bmatrix}$:

$$\begin{aligned}
K^{-\top} &= \begin{bmatrix} \sigma \mathbb{I}_m & -A \\ -A^\top & \sigma \mathbb{I}_n \end{bmatrix}^{-1} = \begin{bmatrix} (\sigma \mathbb{I}_m - \frac{1}{\sigma} A A^\top)^{-1} & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times m} & (\sigma \mathbb{I}_n - \frac{1}{\sigma} A^\top A)^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbb{I}_m & \frac{1}{\sigma} A \\ \frac{1}{\sigma} A^\top & \mathbb{I}_n \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{\sigma} (\mathbb{I}_m - \frac{1}{\sigma^2} A A^\top)^{-1} & \frac{1}{\sigma^2} (\mathbb{I}_m - \frac{1}{\sigma^2} A A^\top)^{-1} A \\ \frac{1}{\sigma^2} (\mathbb{I}_n - \frac{1}{\sigma^2} A^\top A)^{-1} A^\top & \frac{1}{\sigma} (\mathbb{I}_n - \frac{1}{\sigma^2} A^\top A)^{-1} \end{bmatrix} \\
&= \frac{1}{\sigma} \begin{bmatrix} (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} & (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} \tilde{A} \\ (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top & (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \end{bmatrix} \\
&= \frac{1}{\sigma} \begin{bmatrix} (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} & (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} \tilde{A} \\ (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top & (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \end{bmatrix}
\end{aligned}$$

And we see it's basically projection operators with respect to the image/kernel of $\tilde{A} = \frac{1}{\sigma} A$. In summary, we obtain the following formula for the VJP:

$$K \begin{bmatrix} \tilde{\phi} \\ \tilde{\psi} \end{bmatrix} = \begin{bmatrix} \phi \\ \psi \end{bmatrix} \iff \frac{1}{\sigma} \begin{bmatrix} (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} & (\mathbb{I}_m - \tilde{A} \tilde{A}^\top)^{-1} \tilde{A} \\ (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top & (\mathbb{I}_n - \tilde{A}^\top \tilde{A})^{-1} \end{bmatrix} \begin{bmatrix} \phi \\ \psi \end{bmatrix}$$

In particular, we can find the solution by solving 4 smaller linear systems:

$$\begin{aligned}
(\mathbb{I}_m - \tilde{A} \tilde{A}^\top) x &= \frac{1}{\sigma} \phi & (\mathbb{I}_m - \tilde{A} \tilde{A}^\top) y &= \frac{1}{\sigma} \tilde{A} \psi \\
(\mathbb{I}_n - \tilde{A}^\top \tilde{A}) w &= \frac{1}{\sigma} \tilde{A}^\top \phi & (\mathbb{I}_n - \tilde{A}^\top \tilde{A}) z &= \frac{1}{\sigma} \psi
\end{aligned}$$

Or, equivalently

$$\begin{aligned}
(\sigma^2 \mathbb{I}_m - A A^\top) x &= \sigma \phi & (\sigma^2 \mathbb{I}_m - A A^\top) y &= A \psi \\
(\sigma^2 \mathbb{I}_n - A^\top A) w &= A^\top \phi & (\sigma^2 \mathbb{I}_n - A^\top A) z &= \sigma \psi
\end{aligned}$$

Note how this shows that the off-diagonal entries are obviously regularized least squares problems! However, we really do not want to compute the matrices $A A^\top$ and $A^\top A$ since this leads to numerical stability (squared condition number!) To circumvent this issue, we do a reformulation

$$\begin{aligned}
(\sigma^2 \mathbb{I}_m - AA^\top)y = A\psi &\iff y = \underset{y}{\operatorname{argmin}} \| -A^\top y - \psi \|_2^2 - \sigma^2 \|y\|_2^2 \\
&\iff y = \underset{y}{\operatorname{argmin}} \left\| \begin{bmatrix} A^\top \\ \sigma^2 \mathbb{I}_m \end{bmatrix} y - \begin{bmatrix} -\psi \\ \mathbf{0}_m \end{bmatrix} \right\|_2^2 \\
(\sigma^2 \mathbb{I}_n - A^\top A)w = A^\top \phi &\iff w = \underset{w}{\operatorname{argmin}} \| -Aw - \phi \|_2^2 - \sigma^2 \|w\|_2^2 \\
&\iff w = \underset{w}{\operatorname{argmin}} \left\| \begin{bmatrix} A \\ \sigma^2 \mathbb{I}_n \end{bmatrix} w - \begin{bmatrix} -\phi \\ \mathbf{0}_n \end{bmatrix} \right\|_2^2
\end{aligned}$$

2.1 What happens if ϕ or ψ are zero?

In this case we want to fast track the calculation, meaning skip half of the necessary inversions. Looking at the equations we find that if $\phi = 0$ then $x = 0$ and $w = 0$, and if $\psi = 0$ then $y = 0$ and $z = 0$. This suggests that backward substitution is better than forward substitution, since it allows decoupling of the two gradient contributions.

2.2 Via Forward Substitution

Now, the diagonal entries we have a problem: the RHS lacks the A matrix. Thus, we solve in two steps instead:

$$\begin{aligned}
A\mu = \sigma\phi &\implies x = \underset{x}{\operatorname{argmin}} \left\| \begin{bmatrix} A^\top \\ \sigma^2 \mathbb{I}_m \end{bmatrix} x - \begin{bmatrix} -\mu \\ \mathbf{0}_m \end{bmatrix} \right\|_2^2 \\
A^\top \nu = \sigma\psi &\implies z = \underset{z}{\operatorname{argmin}} \left\| \begin{bmatrix} A \\ \sigma^2 \mathbb{I}_n \end{bmatrix} z - \begin{bmatrix} -\nu \\ \mathbf{0}_n \end{bmatrix} \right\|_2^2
\end{aligned}$$

We can optimize further by performing a simultaneous solve:

$$\begin{aligned}
[x, y] &= \underset{x, y}{\operatorname{argmin}} \left\| \begin{bmatrix} A^\top \\ \sigma^2 \mathbb{I}_m \end{bmatrix} [x, y] - \begin{bmatrix} -\mu & -\psi \\ \mathbf{0}_m & \mathbf{0}_m \end{bmatrix} \right\|_2^2 & \mu &= \underset{\mu}{\operatorname{argmin}} \|A\mu - \sigma\phi\|_2^2 \\
[w, z] &= \underset{w, z}{\operatorname{argmin}} \left\| \begin{bmatrix} A \\ \sigma^2 \mathbb{I}_n \end{bmatrix} [w, z] - \begin{bmatrix} -\phi & -\nu \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \right\|_2^2 & \nu &= \underset{\nu}{\operatorname{argmin}} \|A^\top \nu - \sigma\psi\|_2^2
\end{aligned}$$

2.3 Via Backward Substitution

We need to introduce an additional modification:

If $A\mu = \sigma\phi$ not solveable, we instead can multiply the equation by A^\top to obtain:

$$\begin{aligned}
(\sigma^2 \mathbb{I}_m - AA^\top)x = \sigma\phi &\implies (\sigma^2 \mathbb{I}_n - A^\top A)\mu = \sigma A^\top \phi & A^\top x &= \mu \\
(\sigma^2 \mathbb{I}_n - A^\top A)z = \sigma\psi &\implies (\sigma^2 \mathbb{I}_n - A^\top A)A\nu = \sigma A\psi & Az &= \nu
\end{aligned}$$

So:

$$\begin{aligned}\mu &= \underset{\mu}{\operatorname{argmin}} \left\| \begin{bmatrix} A \\ \sigma^2 \mathbb{I}_n \end{bmatrix} \mu - \begin{bmatrix} -\sigma\phi \\ \mathbf{0}_n \end{bmatrix} \right\|_2 & A^\top x &= \mu \\ \nu &= \underset{\nu}{\operatorname{argmin}} \left\| \begin{bmatrix} A^\top \\ \sigma^2 \mathbb{I}_m \end{bmatrix} \nu - \begin{bmatrix} -\sigma\psi \\ \mathbf{0}_m \end{bmatrix} \right\|_2 & Az &= \nu\end{aligned}$$

So

$$\begin{aligned} \begin{bmatrix} \mu & w \end{bmatrix} &= \begin{bmatrix} A \\ \sigma^2 \mathbb{I}_n \end{bmatrix} \begin{bmatrix} -\sigma\phi & -\phi \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \\ \begin{bmatrix} y & \nu \end{bmatrix} &= \begin{bmatrix} A^\top \\ \sigma^2 \mathbb{I}_m \end{bmatrix} \begin{bmatrix} -\psi & -\sigma\psi \\ \mathbf{0}_m & \mathbf{0}_m \end{bmatrix} \end{aligned}$$

In principle, one could try to rephrase these as smaller problems, but for now, it's better to just stick to the bigger system. We can use the **push-through identity** to convert these into 4 linear systems:

$$\begin{aligned} Px &= \phi & Py &= \tilde{A}\psi \\ Qz &= \tilde{A}^\top \phi & Qw &= \psi \end{aligned}$$

Then $\tilde{\phi} = x + y$ and $\tilde{\psi} = z + w$, and the VJP are given by the previous equations:

$$\begin{aligned} \xi^\top \frac{\partial \sigma}{\partial A} &= \xi u v^\top \\ \phi^\top \frac{\partial u}{\partial A} &= (\mathbb{I}_m - uu^\top) \tilde{\phi} v^\top = (\tilde{\phi} - (u^\top \tilde{\phi})u) v^\top \\ \psi^\top \frac{\partial v}{\partial A} &= u \tilde{\psi}^\top (\mathbb{I}_n - vv^\top) = u(\tilde{\psi} - (v^\top \tilde{\psi})v)^\top \end{aligned}$$

3 Spectral Normalization

The VJP of spectral normalization can be computed as follows: let $g(A) = \|A\|_2$ and V be the vector in the VJP. then

$$\begin{aligned} \nabla_A \langle V \mid \frac{A}{\|A\|_2} \rangle &= \langle V \mid \frac{A + \Delta A}{g(A + \Delta A)} - \frac{A}{g(A)} \rangle \\ &= \langle V \mid \frac{A + \Delta A}{g(A) + \nabla g(A) \Delta A} - \frac{A}{g(A)} \rangle \\ &= \langle V \mid \frac{(A + \Delta A)(g(A) - \nabla g(A) \Delta A)}{(g(A) + \nabla g(A) \Delta A)(g(A) - \nabla g(A) \Delta A)} - \frac{A}{g(A)} \rangle \\ &= \langle V \mid \frac{\Delta A g(A) - A \nabla g(A) \Delta A}{g(A)^2} \rangle \\ &= \langle \frac{1}{g(A)} V - \frac{\langle V \mid A \rangle}{g(A)} \nabla g(A) \mid \Delta A \rangle \end{aligned}$$

$$g(A) = 1 \implies \nabla_A \langle V \mid \frac{A}{\|A\|_2} \rangle = \langle V - \langle V \mid A \rangle \nabla g(A) \mid \Delta A \rangle$$

4 Projected gradient

When using spectral normalization we want to do the following:

$$\begin{aligned}\text{update: } A' &= A - \nabla_A \mathcal{L}\left(\frac{A}{\|A\|_2}\right) \\ \text{project: } A &= \frac{A'}{\|A'\|_2}\end{aligned}$$

Moreover, we want:

- During forward, compute $\frac{A}{\|A\|_2}$ only once and then reuse this node.
- Compute $\|A\|_2$ effectively between gradient updates.
 - Avoid built-in torch algos, as they make use of full SVD algos.
- After gradient update, perform projection step. (maybe unnecessary)

NOTE: gradients are different if we include normalization!