Projection onto simplex

Andersen Ang

ECS, Uni. Southampton, UK

andersen.ang@soton.ac.uk

Homepage angms.science

Version: April 2, 2023
First draft: June 15, 2020

Content

$$\Delta = \left\{ \boldsymbol{x} \in \mathbb{R}^r \;\middle|\; \langle \boldsymbol{x}, \mathbf{1}_r \rangle = 1,\ \boldsymbol{x} \geq 0 \right\}$$

Projection onto convex sets

Partial Lagrangian.

Thresholding operator.

Root of piecewise linear equation.

(Euclidean) Simplex

- The r-dimensional a-simplex is the set of nonnegative vector with elements sum up to a:

$$\Delta = \Big\{ \boldsymbol{x} \in \mathbb{R}^r \;\Big|\; \langle \boldsymbol{x}, \mathbf{1}_r \rangle = a,\ \boldsymbol{x} \geq 0 \Big\}.$$

- The r-dimensional unit simplex is a simplex with $a = 1$:

$$\Delta = \Big\{ \boldsymbol{x} \in \mathbb{R}^r \;\Big|\; \langle \boldsymbol{x}, \mathbf{1}_r \rangle = 1,\ \boldsymbol{x} \geq 0 \Big\}.$$

- Δ is a closed and convex set: for $\boldsymbol{x}, \boldsymbol{y} \in \Delta$,

$$\langle \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}, \mathbf{1}_r \rangle = \lambda \langle \boldsymbol{x}, \mathbf{1}_r \rangle + (1-\lambda)\langle \boldsymbol{y}, \mathbf{1}_r \rangle = \lambda a + (1-\lambda) a = a.$$

- This PDF: how to project onto simplex

$$\boldsymbol{x} = P_\Delta(\boldsymbol{y}) = \underset{\boldsymbol{x} \in \Delta}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Projections onto convex sets (POCS)

- The simplex set is the intersection of two sets:

$$\Delta = \underbrace{\{\boldsymbol{x} \in \mathbb{R}^r \mid \langle \boldsymbol{x}, \mathbf{1}_r \rangle = 1\}}_{\mathcal{C}_1} \cap \underbrace{\{\boldsymbol{x} \in \mathbb{R}^r \mid x_i \geq 0\}}_{\mathcal{C}_2}$$

  where $\mathcal{C}_1$ is a hyperplane and $\mathcal{C}_2$ is the nonnegative orthant.

- Projection onto C_1 and projection onto C_2 have closed-form sol., thus a way to solve $P_\Delta(\boldsymbol{y})$ is to use POCS

$$P_\Delta(\boldsymbol{y}) = P_{\mathcal{C}_1 \cap \mathcal{C}_2}(\boldsymbol{y}) = P_{\mathcal{C}_2}(P_{\mathcal{C}_1} ...(P_{\mathcal{C}_2}(P_{\mathcal{C}_1}(\boldsymbol{y}))...)).$$

  POCS works if $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \varnothing$.

- Another POCS using averaged projection: starting at $\boldsymbol{x}_0$, iterate the following

$$\boldsymbol{x}_{k+1} = \frac{1}{2}\Big(P_{\mathcal{C}_1}(\boldsymbol{x}_k) + P_{\mathcal{C}_2}(\boldsymbol{x}_k)\Big) \quad \text{until converges.}$$

- POCS approach can be expensive.

The Lagrangian approach

$$(\mathcal{P}) \; : \; \underset{\boldsymbol{x} \geq \mathbf{0}, \langle \boldsymbol{x}, \mathbf{1}_r \rangle = 1}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

- We now consider the Lagrangian of $\mathcal{P}$
- $\mathcal{P}$ has two constraints
  - Inequality constraint $\boldsymbol{x} \geq \mathbf{0}$
  - Equality constraint $\langle \boldsymbol{x}, \mathbf{1}_r \rangle = 1$
- A trick: ignore the nonnegativity constraint, focus on the equality constraint. The "partial" Lagrangian is

$$L(\boldsymbol{x}, \mu) = \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu\Big(\langle \boldsymbol{x}, \mathbf{1}_r \rangle - 1 \Big),$$

  where μ is the Lagrangian multiplier for the equality constraint.
- We call it partial Lagrangian because we only use Lagrangian multiplier for **part** of the constraint.

Lagrangian is a saddle point problem

- To get the solution $(\boldsymbol{x}^*, \mu^*)$, we need to minimize L w.r.t. $\boldsymbol{x}$ and maximize L w.r.t. μ.

$$(\boldsymbol{x}^*, \mu^*) = \underset{\boldsymbol{x} \geq \mathbf{0}}{\operatorname{argmin}} \; \underset{\mu}{\operatorname{argmax}} \; L(\boldsymbol{x}, \mu)$$

- Recall that when we build L we only put the equality constraint into L, so we have to keep with $\boldsymbol{x} \geq 0$ in the minimization of $\boldsymbol{x}$

- How do we solve $\underset{\boldsymbol{x} \geq \mathbf{0}}{\operatorname{argmin}} \; \underset{\mu}{\operatorname{argmax}} \; L(\boldsymbol{x}, \mu)$? There are two ways
  1. We first solve $\underset{\boldsymbol{x} \geq \mathbf{0}}{\operatorname{argmin}} \; L(\boldsymbol{x}, \mu)$ to get $\boldsymbol{x}^*$

     Then we use $\boldsymbol{x}^*$ to get μ^* by solving $\underset{\mu}{\operatorname{argmax}} \; L(\boldsymbol{x}, \mu)\Big|_{\boldsymbol{x}=\boldsymbol{x}^*}$
  2. We first solve $\underset{\mu}{\operatorname{argmax}} \; L(\boldsymbol{x}^*, \mu)$ to get μ^*

     Then we use μ^* to get $\boldsymbol{x}^*$ by solving $\underset{\boldsymbol{x} \geq \mathbf{0}}{\operatorname{argmin}} \; L(\boldsymbol{x}, \mu)\Big|_{\mu=\mu^*}$

- We will see that approach 1 is better.

Solving $\operatorname*{argmin}_{\boldsymbol{x} \geq \mathbf{0}} L(\boldsymbol{x}, \mu)$

$$\min_{\boldsymbol{x} \geq 0} L(\boldsymbol{x}, \mu) = \min_{\boldsymbol{x} \geq 0} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu \Big(\langle \boldsymbol{x}, \mathbf{1}_r \rangle - 1 \Big).$$

- The problem is on $\boldsymbol{x}$, remove unrelated terms gives

$$\min_{\boldsymbol{x} \geq 0} L(\boldsymbol{x}, \mu) \;=\; \min_{\boldsymbol{x} \geq 0} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \langle \boldsymbol{x}, \mu \mathbf{1}_r \rangle \;=\; \min_{\boldsymbol{x} \geq 0} \sum_{i=1}^{r} \underbrace{\frac{1}{2} (x_i - y_i)^2 + \mu x_i}_{= l_i(x_i)}$$

- L is minimized if all l_i are minimized.
- The minimization of l_i are scalar problems in the form

$$\min_{x \geq 0} \underbrace{\frac{1}{2} (x - y)^2 + \mu x}_{= l(x)} .$$

The solution is a thresholding operator

- By 1st-order optimality condition / Fermat's rule, the minimizer of $l(x)$ is $y-\mu$.
  - If $y-\mu \geq 0$, then $x = y-\mu$ is the feasible optimal sol..
  - If $y-\mu < 0$, then $x = 0$ is the feasible optimal sol..

  Hence, the solution of the minimization of l_i is

$$x_i = [y_i - \mu]_+ := \max\{0, y_i - \mu\}.$$

- The (feasible) minimizer of L is thus

$$\boldsymbol{x}^* = [y - \mu \mathbf{1}_r]_+.$$

- Therefore, the solution of the projection $P_\Delta(\boldsymbol{y})$ is just a thresholding operator: for the vector $\boldsymbol{y}$, subtract each elements by μ. If the result is negative, replace it by zero.

- Question: how to find μ?
  Answer: using the constraint

How to find μ: using the constraint

- Recall primal constraint:

$$\langle x, \mathbf{1}_r \rangle = 1, \boldsymbol{x} \geq 0.$$

  Using summation sign

$$\sum_i x_i = a, \boldsymbol{x} \geq 0.$$

- Put the solution $\boldsymbol{x} = [y - \mu \mathbf{1}_r]_+$ into the primal feasibility, and note that

$$\sum_i [y_i - \mu]_+ = a, \quad \underbrace{[y - \mu \mathbf{1}_r]_+ \geq 0}_{\text{always true}}.$$

- Therefore, to find μ, the key is to solve the piecewise linear equation

$$\sum_i \max\{0, y_i - \mu\} = a.$$

Solving the piecewise linear equation

$$\sum_i \max\{0, y_i - \mu\} = a.$$

- Recall a property of max: if $y_i \leq \mu$, the max term becomes zero. Then the nonlinear equation can be expressed as

  $$\sum_{j : y_j > \mu} (y_j - \mu) = a.$$

  That is, only the components of $\boldsymbol{y}$ that is large enough will contribute to the formation of μ. In other words, small components in $\boldsymbol{y}$ can be ignored.

- As the size of components in $\boldsymbol{y}$ matters, we can solve μ by a sorting method with general complexity $\mathcal{O}(n \log n)$. Details here.

Solving the piecewise linear equation by sorting

- Assume elements in $\boldsymbol{y}$ are sorted: $y_1 \geq y_2 \geq ... \geq y_n$. Define the set

$$\mathcal{J} = \{j \mid y_j > \mu\},\ |\mathcal{J}| = K.$$

Then the piecewise linear equation becomes

$$\sum_{j\,:\,y_j > \mu} (y_j - \mu) \overset{\mathcal{J}}{=} \sum_{j \in \mathcal{J}} y_j - K\mu = a \quad \Longrightarrow \quad \mu = \frac{\displaystyle\sum_{j \in \mathcal{J}} y_j - a}{K}.$$

- The key now is how to find K. It is the largest j such that

$$y_K - \mu > 0.$$

As $y_K - \mu > 0$ is equivalent to $\max\{0, y_K - \mu\} \neq 0$, meaning y_K (the K-th largest component in $\boldsymbol{y}$) contributes to the formation of μ.

Summary of solving the piecewise linear equation by sorting

$$\sum_i \max\{0, y_i - \mu\} = a.$$

- Step-1. Sort $\boldsymbol{y}$.
- Step-2. Find K, which is the largest integer in $\{1, 2, \ldots, n\}$ that $y_K - \frac{\sum_{j \in \mathcal{J}} y_j - a}{K} > 0.$
- Step-3. Output $\mu = \frac{\sum_{j \in \mathcal{J}} y_j - a}{K}.$
- The most expensive part here is step-1, using quick sort, the worst computational complexity is $\mathcal{O}(n \log n)$.
- This method is first proposed in: Held, M., Wolfe, P., Crowder, H.: Validation of subgradient optimization. Math. Program. (1974)

Last page – summary

- The simplex and the projection onto simplex.
- Solving projection onto simplex by POCS.
- Driving the solution of projection onto simplex by partial Lagrangian.
- The solution of projection onto simplex is a thresholding operator.
- The threshold is the root of a piecewise linear equation.
- The threshold can be solved by sorting with complexity $\mathcal{O}(n \log n)$.

End of document.