Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application

Weiran Wang Miguel Á. Carreira-Perpiñán
Electrical Engineering and Computer Science, University of California, Merced
{wwang5,mcarreira-perpinan}@ucmerced.edu

September 3, 2013

Abstract

We provide an elementary proof of a simple, efficient algorithm for computing the Euclidean projection of a point onto the probability simplex. We also show an application in Laplacian K-modes clustering.

1 Projection onto the probability simplex

Consider the problem of computing the Euclidean projection of a point $\mathbf{y} = [y_1, \dots, y_D]^\top \in \mathbb{R}^D$ onto the probability simplex, which is defined by the following optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^D} \quad \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 \tag{1a}$$

$$\text{s.t.} \quad \mathbf{x}^\top \mathbf{1} = 1 \tag{1b}$$

$$\mathbf{x} \geq \mathbf{0}. \tag{1c}$$

This is a quadratic program and the objective function is strictly convex, so there is a unique solution which we denote by $\mathbf{x} = [x_1, \dots, x_D]^\top$ with a slight abuse of notation.

2 Algorithm

The following $\mathcal{O}(D \log D)$ algorithm finds the solution $\mathbf{x}$ to the problem:

Algorithm 1 Euclidean projection of a vector onto the probability simplex.

Input: $\mathbf{y} \in \mathbb{R}^D$
Sort $\mathbf{y}$ into $\mathbf{u}$: $u_1 \geq u_2 \geq \cdots \geq u_D$
Find $\rho = \max\{1 \leq j \leq D: \ u_j + \frac{1}{j}(1 - \sum_{i=1}^{j} u_i) > 0\}$
Define $\lambda = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} u_i)$
Output: $\mathbf{x}$ s.t. $x_i = \max\{y_i + \lambda, 0\}$, $i = 1, \dots, D$.

The complexity of the algorithm is dominated by the cost of sorting the components of $\mathbf{y}$. The algorithm is not iterative and identifies the active set exactly after at most D steps. It can be easily implemented (see section 4).

The algorithm has the following geometric interpretation. The solution can be written as $x_i = \max\{y_i + \lambda, 0\}$, $i = 1, \dots, D$, where λ is chosen such that $\sum_{i=1}^{D} x_i = 1$. Place the values $y_1, \dots, y_D$ as points on the X axis. Then the solution is given by a rigid shift of the points such that the points to the right of the Y axis sum to 1.

The pseudocode above appears in Duchi et al. (2008), although earlier papers (Brucker, 1984; Pardalos and Kovoor, 1990) solved the problem in greater generality[1].

[1] http://www.cs.berkeley.edu/~jduchi/projects/DuchiShSiCh08.html

Other algorithms The problem (1) can be solved in many other ways, for example by particularizing QP algorithms such as active-set methods, gradient-projection methods or interior-point methods. It can also be solved by alternating projection onto the two constraints in a finite number of steps (Michelot, 1986). Another way (Boyd and Vandenberghe, 2004, Exercise 4.1, solution available at `http://see.stanford.edu/materials/lsocoee364b/hw4sol.pdf`) is to construct a Lagrangian formulation by dualizing the equality constraint and then solve a 1D nonsmooth optimization over the Lagrange multiplier. Algorithm 1 has the advantage of being very simple, not iterative, and identifying exactly the active set at the solution after at most D steps (each of cost $O(1)$) after sorting.

3 A simple proof

A proof of correctness of Algorithm 1 can be found in Shalev-Shwartz and Singer (2006) and Chen and Ye (2011), but we offer a simpler proof which involves only the KKT theorem.

We apply the standard KKT conditions for the problem (Nocedal and Wright, 2006). The Lagrangian of the problem is

$$\mathcal{L}(\mathbf{x}, \lambda, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 - \lambda(\mathbf{x}^\top \mathbf{1} - 1) - \boldsymbol{\beta}^\top \mathbf{x}$$

where λ and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_D]^\top$ are the Lagrange multipliers for the equality and inequality constraints, respectively. At the optimal solution $\mathbf{x}$ the following KKT conditions hold:

$$x_i - y_i - \lambda - \beta_i = 0, \qquad i = 1, \ldots, D \tag{2a}$$
$$x_i \geq 0, \qquad i = 1, \ldots, D \tag{2b}$$
$$\beta_i \geq 0, \qquad i = 1, \ldots, D \tag{2c}$$
$$x_i \beta_i = 0, \qquad i = 1, \ldots, D \tag{2d}$$
$$\sum_{i=1}^{D} x_i = 1. \tag{2e}$$

From the complementarity condition (2d), it is clear that if $x_i > 0$, we must have $\beta_i = 0$ and $x_i = y_i + \lambda > 0$; if $x_i = 0$, we must have $\beta_i \geq 0$ and $x_i = y_i + \lambda + \beta_i = 0$, whence $y_i + \lambda = -\beta_i \leq 0$. Obviously, the components of the optimal solution $\mathbf{x}$ that are zeros correspond to the smaller components of $\mathbf{y}$. Without loss of generality, we assume the components of $\mathbf{y}$ are sorted and $\mathbf{x}$ uses the same ordering , i.e.,

$$y_1 \geq \cdots \geq y_\rho \geq y_{\rho+1} \geq \cdots \geq y_D,$$
$$x_1 \geq \cdots \geq x_\rho > x_{\rho+1} = \cdots = x_D,$$

and that $x_1 \geq \cdots \geq x_\rho > 0$, $x_{\rho+1} = \cdots = x_D = 0$. In other words, ρ is the number of positive components in the solution $\mathbf{x}$. Now we apply the last condition and have

$$1 = \sum_{i=1}^{D} x_i = \sum_{i=1}^{\rho} x_i = \sum_{i=1}^{\rho} (y_i + \lambda)$$

which gives $\lambda = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} y_i)$. Hence ρ is the key to the solution. Once we know ρ (there are only D possible values of it), we can compute λ, and the optimal solution is obtained by just adding λ to each component of $\mathbf{y}$ and thresholding as in the end of Algorithm 1. (It is easy to check that this solution indeed satisfies all KKT conditions.) In the algorithm, we carry out the tests for $j = 1, \ldots, D$ if $t_j = y_j + \frac{1}{j}(1 - \sum_{i=1}^{j} y_i) > 0$. We now prove that the number of times this test turns out positive is exactly ρ. The following theorem is essentially Lemma 3 of Shalev-Shwartz and Singer (2006).

Theorem 1. Let ρ be the number of positive components in the solution $\mathbf{x}$, then

$$\rho = \max\{1 \leq j \leq D\colon\ y_j + \tfrac{1}{j}(1 - \textstyle\sum_{i=1}^{j} y_i) > 0\}.$$

Proof. Recall from the KKT conditions (2) that $\lambda\rho = 1-\sum_{i=1}^{\rho} y_i$, $y_i+\lambda > 0$ for $i = 1,\ldots,\rho$ and $y_i+\lambda \le 0$ for $i = \rho+1,\ldots,D$. In the sequel, we show that for $j = 1,\ldots,D$, the test will continue to be positive until $j = \rho$ and then stay non-positive afterwards, i.e., $y_j + \frac{1}{j}(1 - \sum_{i=1}^{j} y_i) > 0$ for $j \le \rho$, and $y_j + \frac{1}{j}(1 - \sum_{i=1}^{j} y_i) \le 0$ for $j > \rho$.

(i) For $j = \rho$, we have

$$y_\rho + \frac{1}{\rho}\Big(1 - \sum_{i=1}^{\rho} y_i\Big) = y_\rho + \lambda = x_\rho > 0.$$

(ii) For $j < \rho$, we have

$$y_j + \frac{1}{j}\Big(1 - \sum_{i=1}^{j} y_i\Big) = \frac{1}{j}\Big(jy_j + 1 - \sum_{i=1}^{j} y_i\Big) = \frac{1}{j}\Big(jy_j + \sum_{i=j+1}^{\rho} y_i + 1 - \sum_{i=1}^{\rho} y_i\Big) = \frac{1}{j}\Big(jy_j + \sum_{i=j+1}^{\rho} y_i + \rho\lambda\Big)$$
$$= \frac{1}{j}\Big(j(y_j+\lambda) + \sum_{i=j+1}^{\rho} (y_i+\lambda)\Big).$$

Since $y_i + \lambda > 0$ for $i = j,\ldots,\rho$, we have $y_j + \frac{1}{j}(1 - \sum_{i=1}^{j} y_i) > 0$.

(iii) For $j > \rho$, we have

$$y_j + \frac{1}{j}\Big(1 - \sum_{i=1}^{j} y_i\Big) = \frac{1}{j}\Big(jy_j + 1 - \sum_{i=1}^{j} y_i\Big) = \frac{1}{j}\Big(jy_j + 1 - \sum_{i=1}^{\rho} y_i - \sum_{i=\rho+1}^{j} y_i\Big) = \frac{1}{j}\Big(jy_j + \rho\lambda - \sum_{i=\rho+1}^{j} y_i\Big)$$
$$= \frac{1}{j}\Big(\rho(y_j+\lambda) + \sum_{i=\rho+1}^{j} (y_j - y_i)\Big).$$

Notice $y_j + \lambda \le 0$ for $j > \rho$, and $y_j \le y_i$ for $j \ge i$ since $\mathbf{y}$ is sorted, therefore $y_j + \frac{1}{j}(1 - \sum_{i=1}^{j} y_i) < 0$.

□

Remarks

1. We denote $\lambda_j = \frac{1}{j}(1-\sum_{i=1}^{j} y_i)$. At the j-th test, λ_j can be considered as a guess of the true λ (indeed, $\lambda_\rho = \lambda$). If we use this guess to compute a tentative solution $\bar{\mathbf{x}}$ where $\bar{x}_i = \max\{y_i + \lambda_j, 0\}$, then it is easy to see that $\bar{x}_i > 0$ for $i = 1,\ldots,j$, and $\sum_{i=1}^{j} \bar{x}_i = 1$. In other words, the first j components of $\bar{\mathbf{x}}$ are positive and sum to 1. If we find $\bar{x}_{j+1} = 0$ (or $y_{j+1} + \lambda_j \le 0$), then we know we have found the optimal solution and $j = \rho$ because $\bar{\mathbf{x}}$ satisfies all KKT conditions.

2. To extend the algorithm to a simplex with a different scale, i.e., $\mathbf{x}^\top\mathbf{1} = a$ for $a > 0$, replace the $1-\sum u_i$ terms with $a - \sum u_i$ in Algorithm 1.

4 Matlab code

The following vectorized Matlab code implements algorithm 1. It projects each row vector in the $N \times D$ matrix $\mathbf{Y}$ onto the probability simplex in D dimensions.

```
function X = SimplexProj(Y)

[N,D] = size(Y);
X = sort(Y,2,'descend');
Xtmp = (cumsum(X,2)-1)*diag(sparse(1./(1:D)));
X = max(bsxfun(@minus,Y,Xtmp(sub2ind([N,D],(1:N)',sum(X>Xtmp,2)))),0);
```

5 An application: Laplacian K-modes clustering

Consider the *Laplacian K-modes clustering algorithm* of Wang and Carreira-Perpiñán (2013) (which is an extension of the K-modes algorithm of Carreira-Perpiñán and Wang, 2013a). Given a dataset $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$ and suitably defined affinity values $w_{nm} \geq 0$ between each pair of points $\mathbf{x}_n$ and $\mathbf{x}_m$ (for example, Gaussian), we define the objective function

$$\min_{\mathbf{Z},\mathbf{C}} \quad \frac{\lambda}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} w_{mn} \left\| \mathbf{z}_m - \mathbf{z}_n \right\|^2 - \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} G\left(\left\| \frac{\mathbf{x}_n - \mathbf{c}_k}{\sigma} \right\|^2 \right) \tag{3a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} z_{nk} = 1, \text{ for } n = 1, \dots, N \tag{3b}$$

$$z_{nk} \geq 0, \text{ for } n = 1, \dots, N, \ k = 1, \dots, K. \tag{3c}$$

$\mathbf{Z}$ is a matrix of $N \times K$, with $\mathbf{Z}^\top = (\mathbf{z}_1, \dots, \mathbf{z}_N)$, where $\mathbf{z}_n = (z_{n1}, \dots, z_{nK})^\top$ are the soft assignments of point $\mathbf{x}_n$ to clusters $1, \dots, K$, and $\mathbf{c}_1, \dots, \mathbf{c}_K \in \mathbb{R}^D$ are modes of the kernel density estimates defined for each cluster (where $G(\cdot^2)$ gives a Gaussian). The problem of projection on the simplex appears in the training problem, i.e., in finding a (local) minimum $(\mathbf{Z}, \mathbf{C})$ of (3), and in the out-of-sample problem, i.e., in assigning a new point to the clusters.

Training The optimization of (3) is done by alternating minimization over $\mathbf{Z}$ and $\mathbf{C}$. For fixed $\mathbf{C}$, the problem over $\mathbf{Z}$ is a quadratic program of NK variables:

$$\min_{\mathbf{Z}} \quad \lambda \operatorname{tr}\left(\mathbf{Z}^\top \mathbf{L} \mathbf{Z}\right) - \operatorname{tr}\left(\mathbf{B}^\top \mathbf{Z}\right) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{Z}\mathbf{1}_K = \mathbf{1}_N \tag{4b}$$

$$\mathbf{Z} \geq \mathbf{0}, \tag{4c}$$

where $b_{nk} = G(\|(\mathbf{x}_n - \mathbf{c}_k)/\sigma\|^2)$, $n = 1, \dots, N$, $k = 1, \dots, K$, and $\mathbf{L}$ is the graph Laplacian computed from the pairwise affinities w_{mn}. The problem is convex since $\mathbf{L}$ is positive semidefinite. One simple and quite efficient way to solve it is to use an (accelerated) gradient projection algorithm. The basic gradient projection algorithm iteratively takes a step in the negative gradient from the current point and projects the new point onto the constraint set. In our case, this projection is simple: it separates over each row of $\mathbf{Z}$ (i.e., the soft assignment of each data point, $n = 1, \dots, N$), and corresponds to a projection on the probability simplex of a K-dimensional row vector, i.e., the problem (1).

Out-of-sample mapping Given a new, test point $\mathbf{x} \in \mathbb{R}^D$, we wish to assign it to the clusters found during training. We are given the affinity vectors $\mathbf{w} = (w_n)$ and $\mathbf{g} = (g_k)$, where w_n is the affinity between $\mathbf{x}$ and $\mathbf{x}_n$, $n = 1, \dots, N$, and $g_k = G(\|(\mathbf{x} - \mathbf{c}_k)/\sigma\|^2)$, $k = 1, \dots, K$, respectively. A natural and efficient way to define an out-of-sample mapping $\mathbf{z}(\mathbf{x})$ is to solve a problem of the form (3) with a dataset consisting of the original training set augmented with $\mathbf{x}$, but keeping $\mathbf{Z}$ and $\mathbf{C}$ fixed to the values obtained during training (this avoids having to solve for all points again). Hence, the only free parameter is the assignment vector $\mathbf{z}$ for the new point $\mathbf{x}$. After dropping constant terms, the optimization problem (3) reduces to the following quadratic program over K variables:

$$\min_{\mathbf{z}} \quad \frac{1}{2} \left\| \mathbf{z} - (\bar{\mathbf{z}} + \gamma \mathbf{g}) \right\|^2 \tag{5a}$$

$$\text{s.t.} \quad \mathbf{z}^\top \mathbf{1}_K = 1 \tag{5b}$$

$$\mathbf{z} \geq \mathbf{0} \tag{5c}$$

where $\gamma = 1/2\lambda \sum_{n=1}^{N} w_n$ and

$$\bar{\mathbf{z}} = \frac{\mathbf{Z}^\top \mathbf{w}}{\mathbf{w}^\top \mathbf{1}} = \sum_{n=1}^{N} \frac{w_n}{\sum_{n'=1}^{N} w_{n'}} \mathbf{z}_n$$

is a weighted average of the training points' assignments, and so $\bar{\mathbf{z}} + \gamma\mathbf{g}$ is itself an average between this and the point-to-centroid affinities. Thus, the solution is the projection of the K-dimensional vector $\bar{\mathbf{z}} + \gamma\mathbf{g}$ onto the probability simplex, i.e., the problem (1) again.

LASS In general, the optimization problem of eq. (4) is called *Laplacian assignment model (LASS)* by Carreira-Perpiñán and Wang (2013b). Here, we consider N items and K categories and want to learn a soft assignment z_{nk} of each item to each category, given an item-item graph Laplacian matrix $\mathbf{L}$ of $N \times N$ and an item-category similarity matrix $\mathbf{B}$ of $N \times K$. The training is as in eq. (4) and the out-of-sample mapping as in eq. (5), and the problem of projection on the simplex appears in both cases.

References

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 2004.

P. Brucker. An $O(n)$ algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, Aug. 1984.

M. Á. Carreira-Perpiñán and W. Wang. The K-modes algorithm for clustering. Unpublished manuscript, arXiv:1304.6478, Apr. 23 2013a.

M. Á. Carreira-Perpiñán and W. Wang. A simple assignment model with Laplacian smoothing. Unpublished manuscript, 2013b.

Y. Chen and X. Ye. Projection onto a simplex. Unpublished manuscript, arXiv:1101.6081, Feb. 10 2011.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In A. McCallum and S. Roweis, editors, *Proc. of the 25th Int. Conf. Machine Learning (ICML'08)*, pages 272–279, Helsinki, Finland, July 5–9 2008.

C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *J. Optimization Theory and Applications*, 50(1):195–200, July 1986.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, second edition, 2006.

P. M. Pardalos and N. Kovoor. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Math. Prog.*, 46(1–3):321–328, Jan. 1990.

S. Shalev-Shwartz and Y. Singer. Efficient learning of label ranking by soft projections onto polyhedra. *J. Machine Learning Research*, 7:1567–1599, 2006.

W. Wang and M. Á. Carreira-Perpiñán. Laplacian K-modes clustering. Unpublished manuscript, 2013.