Projection onto Balls

Royi Avital

June 29, 2017

Contents

1 Projection onto the L_1 Ball

$$\arg\min_{\|x\|_1 \le r} \left\{ \frac{1}{2} \|x - y\|_2^2 \right\}$$

The Lagrangian of the problem can be written as:

$$
\begin{aligned}
L(x, \lambda) &= \frac{1}{2} \|x - y\|^2 + \lambda(\|x\|_1 - r) \\
&= \sum_{i=1}^{n} \left(\frac{1}{2}(x_i - y_i)^2 + \lambda |x_i| \right) - \lambda r \qquad \text{Component wise form}
\end{aligned}
$$

The Dual Function is given by:

$$g(\lambda) = \inf_x L(x, \lambda)$$

The above can be solved component wise for the term $\left(\frac{1}{2}(x_i - y_i)^2 + \lambda |x_i| \right)$ which is solved by the Soft Thresholding Operator:

$$x_i^* = \operatorname{sign}(y_i)(|y_i| - \lambda)_+$$

Where $(t)_+ = \max(t, 0)$.

Clearly, when $\lambda = 0$ it suggests that $x = y$ and $\|y\|_1 \le r$. For the case $\lambda > 0$, which suggests $\|y\|_1 > 1$, all needed is to find the optimal $\lambda > 0$ which is given by the root of the objective function (Which is the constrain of the KKT System):

$$
\begin{aligned}
h(\lambda) &= \sum_{i=1}^{n} |x_i^*(\lambda)| - r \\
&= \sum_{i=1}^{n} (|y_i| - \lambda)_+ - r
\end{aligned}
$$

Namely, the solution is given by $\lambda^* = \lambda : h(\lambda) = 0$. The above is a Piece Wise linear function of λ and its Derivative given by:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\lambda} h(\lambda) &= \frac{\mathrm{d}}{\mathrm{d}\lambda} \sum_{i=1}^{n} (|y_i| - \lambda)_+ \\
&= \sum_{i=1}^{n} -\mathbf{1}_{\{|y_i| - \lambda > 0\}}
\end{aligned}
$$

Hence it can be solved using Newton Iteration.

Listing 1: MATLAB Code - L_1 Ball Projection

```matlab
function [ vX ] = ProjectL1Ball( vY, ballRadius, stopThr )
% ...
%----------------------------------------------------------------------------------
%
% [ vX ] = ProjectL1Ball( vY, ballRadius, stopThr )
%   Solving the Orthoginal Porjection Problem of the input ...
%   vector onto the
%   L1 Ball using Dual Function and Newtin Iteration.
% Input:
%   - vY            -   Input Vector.
%                       Structure: Vector (Column).
%                       Type: 'Single' / 'Double'.
%                       Range: (-inf, inf).
%   - ballRadius    -   Ball Radius.
%                       Sets the Radiuf of the L1 Ball. For ...
%   Unit L1 Ball
%                       set to 1.
%                       Structure: Scalar.
%                       Type: 'Single' / 'Double'.
%                       Range: (0, inf).
%   - stopThr       -   Stopping Threshold.
%                       Sets the trheold of the Newton ...
%   Iteration. The
%                       absolute value of the Objective ...
%   Function will be
%                       below the threshold.
%                       Structure: Scalar.
%                       Type: 'Single' / 'Double'.
%                       Range: (0, inf).
% Output:
%   - vX            -   Output Vector.
%                       The projection of the Input Vector ...
%   onto the Simplex
%                       Ball.
%                       Structure: Vector (Column).
%                       Type: 'Single' / 'Double'.
%                       Range: (-inf, inf).
% References
%   1.  https://math.stackexchange.com/questions/2327504.
%   2.  https://en.wikipedia.org/wiki/Newton%27s_method.
% Remarks:
%   1.  a
% TODO:
%   1.  U.
% Release Notes:
%   -   1.0.001     29/06/2017  Royi Avital
%       *   Enforcing Lambda to be non negative (Dealing ...
%   with the case 'vY'
%           is obeying || vY ||_1 <= ballRadius).
%   -   1.0.000     27/06/2017  Royi Avital
%       *   First release version.
% ...
%----------------------------------------------------------------------------------
%

FALSE   = 0;
```

```
47        TRUE    = 1;
48
49        OFF     = 0;
50        ON      = 1;
51
52        paramLambda     = 0;
53        % The objective functions which its root (The 'paramLambda' ...
              which makes it
54        % vanish) is the solution
55        objVal          = sum(max(abs(vY) - paramLambda, 0)) - ...
              ballRadius;
56
57        while(abs(objVal) > stopThr)
58        objVal          = sum(max(abs(vY) - paramLambda, 0)) - ...
              ballRadius;
59        df              = sum(-((abs(vY) - paramLambda) > 0)); %<! ...
              Derivative of 'objVal' with respect to Lambda
60        paramLambda     = paramLambda - (objVal / df); %<! Newton ...
              Iteration
61        end
62
63        % Enforcing paramLambda ≥ 0. Otherwise it suggests || vY ...
              ||_1 ≤ ballRadius.
64        % Hence the Optimal vX is given by vX = vY.
65        paramLambda = max(paramLambda, 0);
66
67        vX = sign(vY) .* max(abs(vY) - paramLambda, 0);
68
69
70        end
```

2   Projection onto the L_2 Ball

$$\arg\min_{\|x\|_2 \leq r} \left\{ \frac{1}{2}\|x - y\|_2^2 \right\}$$

The Lagrangian is given by:

$$L(x, \lambda) = \frac{1}{2}\|x - y\|_2^2 + \lambda\left(x^T x - r\right)$$

The problem above is convex and Slater's condition holds by choosing $x = \mathbf{0}$.
The KKT Conditions are given by:

$$\nabla_x L(x, \lambda) = x - y + 2\lambda x = 0 \qquad \text{(1) Stationary}$$
$$\lambda\left(x^T x - r\right) = 0 \qquad \text{(2) Slackness}$$
$$x^T x - r \leq 0 \qquad \text{(3) Primal Feasibility}$$
$$\lambda \geq 0 \qquad \text{(4) Dual Feasibility}$$

From (1) one could see that if $\lambda = 0$ then $x = y$ and from (3) it means $x^T x = y^T y \leq r$. If $\lambda > 0$ then from (2) it means $x^T x = r$. Taking (1) and multiply it by x^T yields (Remember $x^T x = r$):

$$x^T x - x^T y + 2\lambda x^T x = 0 \Rightarrow \lambda = \frac{x^T y - r}{2r}$$

Plugging the result back into (1) yields $x = \left(1 + \frac{x^T y - r}{r}\right)^{-1} y = \frac{r}{x^T y} y$. Since $x^T x = r$ and x is a scaled version of y (Namely has the same direction as y) which results in:

$$x = \begin{cases} y & \text{if } \|y\|_2 \leq r \\ r \frac{y}{\|y\|_2} & \text{if } \|y\|_2 > r \end{cases}$$

Listing 2: MATLAB Code - L_2 Ball Projection

```matlab
function [ vX ] = ProjectL2Ball( vY, ballRadius )
% ...
  % ------------------------------------------------------------------
  %
% [ vX ] = ProjectL2Ball( vY, ballRadius, stopThr )
%    Solving the Orthoginal Porjection Problem of the input ...
     vector onto the
%    L1 Ball.
% Input:
%    - vY              -    Input Vector.
%                          Structure: Vector (Column).
%                          Type: 'Single' / 'Double'.
%                          Range: (-inf, inf).
%    - ballRadius      -    Ball Radius.
%                          Sets the Radius of the L2 Ball. For ...
     Unit L2 Ball
%                          set to 1.
%                          Structure: Scalar.
%                          Type: 'Single' / 'Double'.
%                          Range: (0, inf).
% Output:
%    - vX              -    Output Vector.
%                          The projection of the Input Vector ...
     onto the L2
%                          Ball.
%                          Structure: Vector (Column).
%                          Type: 'Single' / 'Double'.
%                          Range: (-inf, inf).
% References
%    1.  h
% Remarks:
%    1.  a
% TODO:
%    1.  U.
% Release Notes:
%    -    1.0.000      29/06/2017  Royi Avital
%         *   First release version.
```

```
33      % ...
        %
34
35      FALSE    = 0;
36      TRUE     = 1;
37
38      OFF      = 0;
39      ON       = 1;
40
41      vX = min((ballRadius / norm(vY, 2)), 1) * vY;
42
43
44      end
```

3 Projection onto the L_∞ Ball

$$\arg\min_{\|x\|_\infty \leq r} \left\{ \frac{1}{2} \|x - y\|_2^2 \right\}$$

Since $\|x\|_\infty = \max_i |x_i|,\ i = 1, 2, \ldots, n$ the above can be written as:

$$\begin{aligned} \arg\min_x \quad & \frac{1}{2} \|x - y\|_2^2 \\ \text{subject to} \quad & x_i \leq r \quad i = 1, 2, \ldots, n \\ & -x_i \leq r \quad i = 1, 2, \ldots, n \end{aligned}$$

The Lagrangian is given by:

$$L(x, \lambda_1, \lambda_2) = \frac{1}{2} \|x - y\|_2^2 + \lambda_1^T (x - r) + \lambda_2^T (-x - r)$$

The problem above is convex and Slater's condition holds by choosing $x = \mathbf{0}$. The KKT Conditions are given by:

$$\begin{aligned} \nabla_x L(x, \lambda_1, \lambda_2) = x - y + \lambda_1 x - \lambda_2 x = 0 && \text{(1) Stationary} \\ \lambda_{1,i}(x_i - r) = 0 && \text{(2) Slackness} \\ \lambda_{2,i}(-x_i - r) = 0 && \text{(3) Slackness} \\ x_i - r \leq 0 && \text{(4) Primal Feasibility} \\ -x_i - r \leq 0 && \text{(5) Primal Feasibility} \\ \lambda_{1,i}, \lambda_{2,i} \geq 0 && \text{(6) Dual Feasibility} \end{aligned}$$

As can be seen from above, the problem can be solved component wise. Hence the sub script i for the Lagrange Multiplier will be be neglected. From

6

(2) and (3) it can be shown that either $\lambda_1 > 0$ or $\lambda_2 > 0$ but not both. Hence if $\lambda_1 = 0$ then $\lambda_2 = 0$ which means $x_i = y_i$ and $|x_i| \leq 1$. Moreover, if $\lambda_1 > 0$ then $\lambda_2 = 0$ hence $x_i = r$ and $y_i > 1$. The same goes the other way around which yields:

$$x_i = \operatorname{sign}(y_i) \min\{r, |y_i|\}, \ i = 1, 2, \ldots, n$$

Listing 3: MATLAB Code - L_∞ Ball Projection

```
 1      function [ vX ] = ProjectLInfBall( vY, ballRadius )
 2      % ...
        ------------------------------------------------------------------------
        %
 3      % [ vX ] = ProjectL2Ball( vY, ballRadius, stopThr )
 4      %    Solving the Orthoginal Porjection Problem of the input ...
               vector onto the
 5      %    L Inf Ball.
 6      % Input:
 7      %    - vY              -    Input Vector.
 8      %                          Structure: Vector (Column).
 9      %                          Type: 'Single' / 'Double'.
10      %                          Range: (-inf, inf).
11      %    - ballRadius      -    Ball Radius.
12      %                          Sets the Radius of the L Inf Ball. ...
               For Unit L Inf
13      %                          Ball set to 1.
14      %                          Structure: Scalar.
15      %                          Type: 'Single' / 'Double'.
16      %                          Range: (0, inf).
17      % Output:
18      %    - vX              -    Output Vector.
19      %                          The projection of the Input Vector ...
               onto the L Inf
20      %                          Ball.
21      %                          Structure: Vector (Column).
22      %                          Type: 'Single' / 'Double'.
23      %                          Range: (-inf, inf).
24      % References
25      %    1.  h
26      % Remarks:
27      %    1.  a
28      % TODO:
29      %    1.  U.
30      % Release Notes:
31      %    -    1.0.000     29/06/2017   Royi Avital
32      %         *   First release version.
33      % ...
        ------------------------------------------------------------------------
        %
34
35      FALSE    = 0;
36      TRUE     = 1;
37
38      OFF      = 0;
39      ON       = 1;
```

```
40
41        vX = sign(vY) .* min(abs(vY), ballRadius);
42
43
44     end
```

4   Projection onto the Simplex

$$\arg\min_{x \succeq 0, \mathbf{1}^T x = r} \left\{ \frac{1}{2} \|x - y\|_2^2 \right\}$$

The Lagrangian of the problem can be written as (Leaving the non negativity constrain implicit):

$$L(x, \mu) = \frac{1}{2} \|x - y\|^2 + \mu \left(\mathbf{1}^T x - r \right)$$

The Dual Function is given by (Includes the Non Negativity Constrain):

$$g(\mu) = \inf_{x \succeq 0} L(x, \mu)$$

$$= \inf_{x \succeq 0} \sum_{i=1}^{n} \left(\frac{1}{2} (x_i - y_i)^2 + \mu x_i \right) - \mu r \qquad \text{Component wise form}$$

taking advantage of the Component Wise form the solution is given:

$$x_i^* = (y_i - \mu)_+$$

Where the solution takes into account the non negativity constrain by projecting onto $\mathbb{R}_+$

The solution is given by finding the μ which holds the constrain. In the L_1 case above the constrain was in inequality form hence λ had to be non negative Yet the above is equality constrain hence μ can have any value and it is not limited to non negativity as λ in the L_1 case.

The objective function (From the KKT) is given by:

$$h(\mu) = \sum_{i=1}^{n} x_i^* - r = \sum_{i=1}^{n} (y_i - \mu)_+ - r$$

Namely, the solution is given by $\mu^* = \mu : h(\mu) = 0$. The above is a Piece Wise linear function of μ and its Derivative given by:

$$\frac{\mathrm{d}}{\mathrm{d}\mu} h\left(\mu\right) = \frac{\mathrm{d}}{\mathrm{d}\mu} \sum_{i=1}^{n} \left(y_i - \mu\right)_+$$

$$= \sum_{i=1}^{n} -\mathbf{1}_{\{y_i - \mu > 0\}}$$

Hence it can be solved using Newton Iteration.

Listing 4: MATLAB Code - Simplex Projection

```matlab
function [ vX ] = ProjectSimplex( vY, ballRadius, stopThr )
% ...
%          --------------------------------------------------------------------------------------------------------
%
% [ vX ] = ProjectSimplex( vY, ballRadius, stopThr )
%   Solving the Orthoginal Porjection Problem of the input ...
    vector onto the
%   Simplex Ball using Dual Function and Newton Iteration.
% Input:
%   - vY            -   Input Vector.
%                       Structure: Vector (Column).
%                       Type: 'Single' / 'Double'.
%                       Range: (-inf, inf).
%   - ballRadius    -   Ball Radius.
%                       Sets the Radius of the Simplex Ball. ...
    For Unit
%                       Simplex set to 1.
%                       Structure: Scalar.
%                       Type: 'Single' / 'Double'.
%                       Range: (0, inf).
%   - stopThr       -   Stopping Threshold.
%                       Sets the trheolds of the Newton ...
    Iteration. The
%                       absolute value of the Objective ...
    Function will be
%                       below the threshold.
%                       Structure: Scalar.
%                       Type: 'Single' / 'Double'.
%                       Range: (0, inf).
% Output:
%   - vX            -   Output Vector.
%                       The projection of the Input Vector ...
    onto the Simplex
%                       Ball.
%                       Structure: Vector (Column).
%                       Type: 'Single' / 'Double'.
%                       Range: (-inf, inf).
% References
%   1.  https://math.stackexchange.com/questions/2327504.
%   2.  https://en.wikipedia.org/wiki/Newton%27s_method.
% Remarks:
%   1.  a
% TODO:
```

```matlab
37      %    1.  U.
38      % Release Notes:
39      %    -    1.0.001      09/05/2017  Royi Avital
40      %          *    Renaming 'paramLambda' -> 'paramMu' to match ...
            derivation.
41      %    -    1.0.000      09/05/2017  Royi Avital
42      %          *    First release version.
43      % ...
            ------------------------------------------------------------------------
            %
44
45      FALSE   = 0;
46      TRUE    = 1;
47
48      OFF     = 0;
49      ON      = 1;
50
51      paramMu = min(vY) - ballRadius;
52      % The objective functions which its root (The 'paramLambda' ...
            which makes it
53      % vanish) is the solution
54      objFun      = sum( max(vY - paramMu, 0) ) - ballRadius;
55
56      while(abs(objFun) > stopThr)
57      objFun      = sum( max(vY - paramMu, 0) ) - ballRadius;
58      df          = sum(-((vY - paramMu) > 0)); %<! Derivative of ...
            'objVal' with respect to Mu
59      paramMu = paramMu - (objFun / df); %<! Newton Iteration
60      end
61
62      vX = max(vY - paramMu, 0);
63
64
65      end
```