

# Weighted Projection Quantiles Algorithm

October 7, 2014

**Algorithm to calculate weighted projection quantile** along the vector  $\mathbf{u} \in \mathcal{B}_p$ , given a set of observations  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$ :

1. **Compute  $Q_{proj}(\mathbf{u})$ , the projection quantile along  $\mathbf{u}$**

- Project each  $\mathbf{X}_i$  along  $\mathbf{u}$  to obtain  $X_{\mathbf{u}i} = \frac{\langle \mathbf{X}_i, \mathbf{u} \rangle}{\|\mathbf{u}\|}$ , for  $i = 1, 2, \dots, n$ .
- Find  $\alpha = \frac{1+\|\mathbf{u}\|}{2}$ -th quantile of  $X_{\mathbf{u}1}, \dots, X_{\mathbf{u}n}$ , say  $q_{\mathbf{u}}$ .
- $Q_{proj}(\mathbf{u}) = q_{\mathbf{u}}\mathbf{e}_{\mathbf{u}}$ ,  $\mathbf{e}_{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|$  being the unit vector along  $\mathbf{u}$ .

2. **Compute *Weights corresponding to this projection quantile*  $Q_{proj}(\mathbf{u})$**

- Compute global weights for the direction vector  $\mathbf{u}$  by  $k$ -mean distance:
  - Compute  $k$ -mean distance corresponding to  $Q_{proj}(\mathbf{u})$  using  $\bar{d}_k = \frac{1}{n} \sum_{i=1}^n d_i \mathbb{I}_{\{d_i < d_{(k)}\}}$ , where  $d_i$  is the euclidean distance of  $\mathbf{X}_i$  from  $Q_{proj}(\mathbf{u})$  given by  $\|\mathbf{X}_i - Q_{proj}(\mathbf{u})\|$ .  $k$  is a tuning parameter.
  - Compute the weights corresponding to  $\mathbf{u}$ :

$$w_{\mathbf{u}} = \exp(-a.d_k)$$

where  $a$  is a tuning parameter.

- Compute weights for each sample point  $\mathbf{X}_i; i = 1, 2, \dots, n$ :
  - Compute the orthogonal Norms by  $\|\mathbf{X}_{\mathbf{u}\perp i}\| = \|\mathbf{X}_i - X_{\mathbf{u}i}\mathbf{e}_{\mathbf{u}}\|$ .
  - Compute weight of  $i^{th}$  sample:

$$w_{2i} = \begin{cases} \exp \left[ -b \frac{\|\mathbf{X}_{\mathbf{u}\perp i}\|}{\|\mathbf{X}_i\|} \right] & \text{if } \|\mathbf{X}_i\| \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

$b, \epsilon$  being tuning parameters.

3. **Compute the weighted projection quantile**

- Define  $\tilde{X}_{\mathbf{u}i} = w_{\mathbf{u}}w_{2i}X_{\mathbf{u}i}$  for  $i = 1, \dots, n$ .
- Find  $\alpha = \frac{1+\|\mathbf{u}\|}{2}$ -th quantile of  $\tilde{X}_{\mathbf{u}1}, \dots, \tilde{X}_{\mathbf{u}n}$ . Let it be  $\tilde{q}_{\mathbf{u}}$ .
- Find the weighted projection quantile as  $\tilde{Q}_{proj}(\mathbf{u}) = \tilde{q}_{\mathbf{u}}\mathbf{e}_{\mathbf{u}}$ .

**Definition** Given a random vector  $\mathbf{X} \in \mathbb{R}^p$  that follows a multivariate distribution  $F$ , and a point  $\mathbf{p} \in \mathbb{R}^p$ , find  $\alpha_{\mathbf{p}}$  such that  $\|\mathbf{p}\|$  is the  $\alpha_{\mathbf{p}}$ -th quantile for the projection of  $\mathbf{X}$  on  $\mathbf{p}$ , say  $X_{\mathbf{p}}$ . Then the **Projection Quantile Depth** (PQD) at  $\mathbf{p}$  with respect to  $F$  is defined as

$$D(\mathbf{p}, F) = \exp(-\alpha_{\mathbf{p}})$$

Given data  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$ , the PQD at a given  $\mathbf{p}$  can be estimated by finding the two nearest points on either side of  $\|\mathbf{p}\|$  along  $\mathbf{p}$ , say  $\mathbf{p}_1, \mathbf{p}_2$ , obtain their corresponding quantiles, say  $\alpha_1, \alpha_2$  respectively, then estimate  $\alpha_{\mathbf{p}}$  by a linear approximation:

$$\hat{\alpha}_{\mathbf{p}} = \frac{(\alpha_1 - \alpha_2)(\|\mathbf{p}\| - \|\mathbf{p}_1\|)}{\|\mathbf{p}_1\| - \|\mathbf{p}_2\|} + \alpha_1$$

and plugging it in the above definition.

---

**Algorithm 1** Algorithm for PQD-based classification

---

- 1: **procedure** PQDClassifier(training data  $\mathbf{X}_i \in \mathbb{R}^{n_i \times p}$  with class labels  $i$ ;  $i = 1, 2, \dots, k$ , new data  $\mathbf{x}_{new} \in \mathbb{R}^p$ )
- 2:   Set  $i = 1$ .
- 3:   *top*:
- 4:   Estimate from the sample the PQD of  $\mathbf{p}$  with respect to the  $i^{th}$  population, say  $D(\mathbf{x}_{new}, \mathbf{X}_i)$ .
- 5:   **if**  $i = k$  **then Stop**
- 6:   **else**
- 7:     Set  $i \leftarrow i + 1$ , **goto top**
- 8:
- 9:   Find  $c$  that maximizes the PQD of  $\mathbf{x}_{new}$  w.r.t. all possible classes:

$$D(\mathbf{x}_{new}, \mathbf{X}_c) = \max\{D(\mathbf{x}_{new}, \mathbf{X}_i) : i = 1, 2, \dots, k\}$$

- 10:   Assign class  $c$  to new data  $\mathbf{x}_{new}$ .
- 

**Note** One can define a weighted version of PQD by replacing  $X_{\mathbf{p}}$  by their weighted version  $\tilde{X}_{\mathbf{p}}$ . A weighted classification scheme follows similarly.