Algorithm for Robust \((\ell^p)\) Regression

**What we want: general idea.** We want to find out how a quantity \(y\) depends on another quantity \(x\). We know that the dependence can be described as \(y = f(x, p)\), for some parameters \(p\). We need to find the values of these parameters.

To find these values, in several situations we measure the values of both quantities. Thus, we get the values \(x_1, \ldots, x_n\) and \(y_1, \ldots, y_n\). We want to find the values of the parameters \(p\) for which, for each \(k\) from 1 to \(n\), we have \(y_k \approx a \cdot x_k + b\).

**What we want: first example.** The simplest case is when \(y\) does not depend on \(x\) at all: \(f(x, a) = a\). In this case, we simply have \(n\) estimates \(y_1, \ldots, y_n\) for the same quantity \(a\). In this case, we want to find the value \(a\) for which \(y_k \approx a\) for all \(k\) from 1 to \(n\).

**What we want: second example.** The next simplest case is when we have a linear dependence, i.e., \(y = a \cdot x + b\) for some parameters \(a\) and \(b\).

In this case, we are given the values \(x_1, \ldots, x_n\) and \(y_1, \ldots, y_n\), and we want to find the values \(a\) and \(b\) for which, for all \(k\) from 1 to \(n\), we have \(y_k \approx a \cdot x_k + b\).

**Traditional Least Squares approach and its limitations.** The traditional approach to the above problem is to find the value \(p\) for which the following sum is the smallest possible: \(\sum_{k=1}^{n} (y_k - f(x_k, p))^2\).

In particular, for the case of a constant model, we find the value \(a\) that minimizes the sum \(\sum_{k=1}^{n} (y_k - a)^2\). If we differentiate this expression with respect to \(a\) and equate the derivative to 0, we conclude that

\[
a = \frac{y_1 + \ldots + y_n}{n}.
\]
For the case of a linear model, we find the values $a$ and $b$ for which the sum $\sum_{k=1}^{n} (y_k - (a \cdot x_k + b))^2$ is the smallest possible.

The limitation of this approach is that if one of the values $y_k$ is an outlier, the whole estimate will be drastically changed. As the saying goes, when Bill Gates walks into a bar, on average, everyone becomes a millionaire. We need methods which are more robust, i.e., less dependent on outliers.

**$\ell^p$-method: precise formulation of the problem.** Instead of minimizing the sum of the squares of the approximation error $e_k \overset{\text{def}}{=} y_k - f(x_k, p)$, let us select some other function $f(z)$ and minimize the sum $\sum_{k=1}^{n} f(e_k)$.

It is reasonable to require that the method should not depend on which units we use to measure $y$, i.e., that the method be scale-invariant. We already know that scale-invariant functions have the form $f(z) = \text{const} \cdot z^\alpha$ for some $\alpha$. Thus, the new formulation is to minimize the sum $\sum_{k=1}^{n} |y_k - f(x_k, p)|^\alpha$.

In particular, for the constant model, we find the value $a$ for which the sum $\sum_{k=1}^{n} |y_k - a|^\alpha$ is the smallest possible.

For the case of a linear model, we find the values $a$ and $b$ for which the sum $\sum_{k=1}^{n} |y_k - (a \cdot x_k + b)|^\alpha$ is the smallest possible.

When $\alpha = 1$, we get the most robust method. For example, for the case of the constant model, the resulting value $a$ is the median.

In general, the smaller $\alpha$, the more robust is the method. So, $\alpha$ can be viewed as a characteristic of the method’s robustness.

**Resulting algorithm for the linear case: main steps.** First, we apply the usual Least Squares approach and find the values $a$ and $b$ for which the sum $\sum_{k=1}^{n} (y_k - (a \cdot x_k + b))^2$ is the smallest possible. These values will be our first approximations $a^{(1)}$ and $b^{(1)}$.

After that, we perform the following iterative process. For each $s = 2, 3, \ldots$:

- Once we know the values $a^{(s-1)}$ and $b^{(s-1)}$, we compute, for all $k$ from 1 to $n$, the values
  \[ w_k \overset{\text{def}}{=} |y_k - (a^{(s-1)} \cdot x_k + b^{(s-1)})|^{-(2-\alpha)}. \]
Then, we find the values \(a\) and \(b\) for which the sum

\[
\sum_{k=1}^{n} w_k \cdot (y_k - (a \cdot x_k + b))^2
\]

is the smallest possible (see section titled Auxiliary algorithm on how to do it). These new values are the next approximation \(a^{(s)}\) and \(b^{(s)}\) to the desired values \(a\) and \(b\).

We then apply the same two-step procedure to find \(a^{(s+1)}\) and \(b^{(s+1)}\) based on \(a^{(s)}\) and \(b^{(s)}\), etc.

We stop when, for some pre-determined value \(\varepsilon > 0\) – that determines how accurately we compute the parameters – we have \(|a^{(s)} - a^{(s-1)}| \leq \varepsilon\) and \(|b^{(s)} - b^{(s-1)}| \leq \varepsilon\).

Auxiliary algorithm: derivation and the resulting formulas. Differentiating the expression \(\sum_{k=1}^{n} w_k \cdot (y_k - (a \cdot x_k + b))^2\) by \(a\) and by \(b\) and equating the derivatives to 0, we conclude that

\[
a \cdot x + b \cdot \bar{y} = \bar{y}; \quad (1)
\]

\[
a \cdot x^2 + b \cdot \bar{x} = \bar{x} \cdot \bar{y}; \quad (2)
\]

where for each quantity \(q\), the expression \(\bar{q}\) means \(\sum_{k=1}^{n} w_k \cdot q_k\). So:

\[
\bar{I} \overset{\text{def}}{=} \sum_{k=1}^{n} w_k, \quad \bar{x} \overset{\text{def}}{=} \sum_{k=1}^{n} w_k \cdot x_k,
\]

\[
\bar{y} \overset{\text{def}}{=} \sum_{k=1}^{n} w_k \cdot y_k, \quad \bar{x^2} \overset{\text{def}}{=} \sum_{k=1}^{n} w_k \cdot x_k^2,
\]

\[
\bar{x} \cdot \bar{y} \overset{\text{def}}{=} \sum_{k=1}^{n} w_k \cdot x_k \cdot y_k.
\]

If we multiply both sides of (1) by \(\bar{x}\) and (2) by \(\bar{I}\), we get

\[
a \cdot (\bar{x})^2 + b \cdot \bar{I} \cdot \bar{x} = \bar{x} \cdot \bar{y};
\]

\[
a \cdot \bar{I} \cdot \bar{x^2} + b \cdot \bar{I} \cdot \bar{x} = \bar{I} \cdot \bar{x} \cdot \bar{y}.
\]
Subtracting the first equation from the second one, we get

\[ a \cdot (1 \cdot x^2 - (\bar{x})^2) = \bar{1} \cdot \bar{xy} - \bar{x} \cdot \bar{y}, \]

hence

\[ a = \frac{\bar{1} \cdot \bar{xy} - \bar{x} \cdot \bar{y}}{1 \cdot x^2 - (\bar{x})^2}. \quad (3) \]

Now, \( b \) can be found from the equation (2), as

\[ b = \frac{\bar{y} - a \cdot \bar{x}}{1}. \quad (4) \]

**Algorithm: summary.** First, we take \( w_k = 1 \), and use the the expression (3) and (4) to find estimates for \( a \) and \( b \).

Then, on each iteration, we use the values \( a \) and \( b \) from the previous iteration to compute the new values for the weights \( w_k \):

\[ w_k \overset{\text{def}}{=} |y_k - (a \cdot x_k + b)|^{-(2-\alpha)}. \quad (5) \]

Then, we apply the formulas (3) and (4) with the new weights \( w_k \) to compute the next approximation to \( a \) and \( b \), etc. – until the process converges, i.e., until the values \( a \) and \( b \) on the two consecutive iterations are sufficiently close (i.e., differing by no more than some pre-determined constant \( \varepsilon > 0 \)).