1 ADMM

Let $x, y, w \in R^n$ be the location, response, and weight data, respectively. We are trying to solve:

$$\min_\theta \sum_{i=1}^n w_i(y_i - \theta_i)^2$$
$$\text{where } \theta \in \mathcal{K}, \mathcal{K} = \{\theta \in R^n : \exists \text{ convex function } f(\cdot), \text{s.t. } f(x_i) = \theta_i, i = 1, 2, \cdots, n\}$$

This is equivalent to:

$$\begin{aligned}
&\min_\theta \sum_{i=1}^n w_i(y_i - \theta_i)^2 \\
&\text{s.t. } z_i = \frac{\theta_{i+1} - \theta_i}{x_{i+1} - x_i} \text{ for } i = 1, 2, \cdots, n-1 \\
&z_1 \leq z_2 \leq \cdots \leq z_{n-1}
\end{aligned}$$

Let $D^{(x,1)} = \text{diag}\left(\frac{1}{x_2 - x_1}, \cdots, \frac{1}{x_n - x_{n-1}}\right) \cdot D^{(1)}$, where $D^{(1)} \in R^{(n-1)\times n}$ is the discrete different operator of order 1, then the first constraint can be written as $D^{(x,1)}\theta - z = 0$.

We implement the ADMM algorithm as follows. Let $Q = \{z \in R^{n-1} : z_1 \leq z_2 \cdots \leq z_{n-1}\}$, and

$$L(\theta, z, u) = \frac{1}{2}\|W^{\frac{1}{2}}(\theta - y)\|_2^2 + \delta(z \in Q) + \frac{\rho}{2}\|D^{(x,1)}\theta - z + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2,$$

where $W = \text{diag}(w_1, \cdots, w_n)$ and $\delta(\cdot)$ is the convex indicator function. The ADMM algorithm then iterates the steps:

$$\begin{cases}
\theta_{t+1} = (W + \rho(D^{(x,1)})^T D^{(x,1)})^{-1}(Wy + \rho(D^{(x,1)})^T(z_t - u_t)) \\
z_{t+1} = \arg\min_{z \in \mathbb{R}^{n-1}} \frac{\rho}{2}\|D^{(x,1)}\theta_{t+1} - z + u_t\|_2^2 + \delta(z \in Q) \\
u_{t+1} = u_t + (D^{(x,1)}\theta_{t+1} - z_{t+1})
\end{cases}$$

The θ-update step is a banded linear system solve, which can be implemented in time $\mathcal{O}(n)$. The z-update step can be solved using the pool-adjacent-violators algorithm (PAVA), which costs $\mathcal{O}(n)$. Therefore, each iteration is $\mathcal{O}(n)$.

Assuming linear convergence, the total run time to achieve an error of ϵ is $\mathcal{O}\left(nlog\frac{1}{\epsilon}\right)$. Aaditya believes that the convergence rate for problems of time type is indeed linear, as proved by Hong and Luo [1].

2 Duality Gap

We denote f^* as the convex conjugate of f. Given any u, taking inf over (θ, z) with respect to the Lagrangian

$$\frac{1}{2}\|W^{\frac{1}{2}}(y - \theta)\|_2^2 + \delta(z \in Q) + \rho u^T(D^{(x,1)}\theta - z),$$

we obtain the dual

$$\begin{aligned}
D(u) &= \min_{\theta,z}\left\{\frac{1}{2}\|W^{\frac{1}{2}}(y-\theta)\|_2^2 + \delta(z\in Q) + \rho u^T(D^{(x,1)}\theta - z)\right\}\\
&= \min_{\theta}\left\{\frac{1}{2}\|W^{\frac{1}{2}}(y-\theta)\|_2^2 + \rho u^T D^{(x,1)}\theta\right\} + \rho\cdot\min_z\left\{\delta(z\in Q) - u^T z\right\}\\
&= -\sup_{\theta}\left\{-\rho u^T D^{(x,1)}\theta - \frac{1}{2}\|W^{\frac{1}{2}}(y-\theta)\|_2^2\right\} - \rho\cdot\sup_z\left\{u^T z - \delta(z\in Q)\right\}\\
&= \left(\frac{\rho^2}{2}-\rho\right)\cdot\|W^{-\frac{1}{2}}(D^{(x,1)})^T u\|_2^2 + u^T D^{(x,1)}y - \delta^*(u)
\end{aligned}$$

The duality gap is therefore

$$\frac{1}{2}\|W^{\frac{1}{2}}(y_t-\theta_t)\|_2^2 + \left(\rho - \frac{\rho^2}{2}\right)\cdot\|W^{-\frac{1}{2}}(D^{(x,1)})^T u_t\|_2^2 - u_t^T D^{(x,1)}y_t + \delta(z_t\in Q) + \delta^*(u_t)$$

$$= \begin{cases} \infty \text{ , if } z_t \notin Q \text{ or } u_t \notin \text{ the polar cone of Q} \\ \frac{1}{2}\|W^{\frac{1}{2}}(y_t-\theta_t)\|_2^2 + \left(\rho - \frac{\rho^2}{2}\right)\cdot\|W^{-\frac{1}{2}}(D^{(x,1)})^T u_t\|_2^2 - u_t^T D^{(x,1)}y_t \text{ , o.w.} \end{cases}$$

As Anup suggests, to check if u is in the polar cone of Q, we can solve the linear program

$$\begin{aligned} &\max_z \frac{u^T z}{\|z\|_2} \\ &\text{s.t. } z\in Q \end{aligned}$$

and check if the optimal value is negative.

In the code, I rescale the duality gap by $\left(\frac{x_n-x_1}{n}\right)^2$, before using it as a termination criteria. This is because u is multiplied by $\frac{n}{x_n-x_1}$ and squared in the duality gap. The rescaled duality gap works well in simulations.

3 Experiments

3.1 Simulated data

To create the simulated data, I generate the sequence of x from $n = 1001$ equally spaced points in [-0.5, 0.5], and $y_i = x_i^2 + 0.02z_i$, with z_i drawn from $\mathcal{N}(0,1)$ independently, for $i = 1, 2, \cdots, n$. The weights $w_1 = w_2 \cdots = w_n = 1$. The y, x, w data frame is created in R ("cvx_reg_simulations.R") and outputted in cvs file "cvxReg_input.csv". This is then taken as the input file for C ("CvxReg1d.c"), where convex regression is performed and the outputs are saved into "cvxReg_output_1.csv" (the fitted curve) and "csvReg_output_2.csv" (the time series of duality gap).

For ADMM, I choose $\rho \sim \left(\frac{x_n-x_1}{n}\right)^2$. This is due to the observation that, in the θ-update step of ADMM, ρ multiplies the matrix/vector whose elements are of order $\left(\frac{n}{x_n-x_1}\right)^2$ (notice that u, z are of order $\left(\frac{n}{x_n-x_1}\right)$). Here, I choose $\rho = 10^{-4}, 10^{-5}, 10^{-6}$, and set the termination criteria as duality gap $\leq 10^{-5}$ or reaching 1000 iterations, whichever comes first.

Figure 1 shows time series of the (rescaled) duality gap, the sequence of simulated data, and the fitted curve. As ρ increases, the (rescaled) duality gap converges faster, though the fitted curve when $\rho = 10^{-4}$ appears different from the fitted curves when $\rho = 10^{-5}, 10^{-6}$.

Figure 3 diagnoses whether the fitted curve is convex, by showing the first and second differences for the fitted data (blue), compared with the corresponding reference curves for $y = x^2$ case (red). We can see that, when $\rho = 10^{-5}, 10^{-6}$, the first differences for fitted curves are monotone increasing, and the second differences stay non-negative, meaning that the fitted curve is indeed convex. However, when $\rho = 10^{-4}$, the first difference for the fitted curve breaks monotonicity, and the second difference breaks the non-negativity. This suggests that the convergence of duality gap when $\rho = 10^{-4}$ is merely due to the "rescaling". I then re-run the ADMM for $\rho = 10^{-4}$ with only maxima $= 1000$ as the termination criteria. The fitted curve and diagnostic results stay the same. $\rho = 10^{-4}$ is not a proper choice for a statistically sound result.

I run the same simulation with unit spacing (i.e. 1000 times the spacing in the original x), and use $\rho = 100, 10, 1$. The rescaling of the duality gap here is simply 1. The results are shown in Figure 2 and Figure 4.

3.2 The choice of ρ

At unit spacing, the fastest convergence when $\rho \in [1, 100]$ appears at $\rho = 25$, which takes 0.53 sec. At 10^{-3} unit spacing, the fastest convergence when $\rho \in [10^{-5}, 10^{-6}]$ appears at $\rho = 10^{-5}$, which takes 2.65 sec. This empirically supports selecting $\rho \sim \left(\frac{x_n - x_1}{n}\right)^2$.

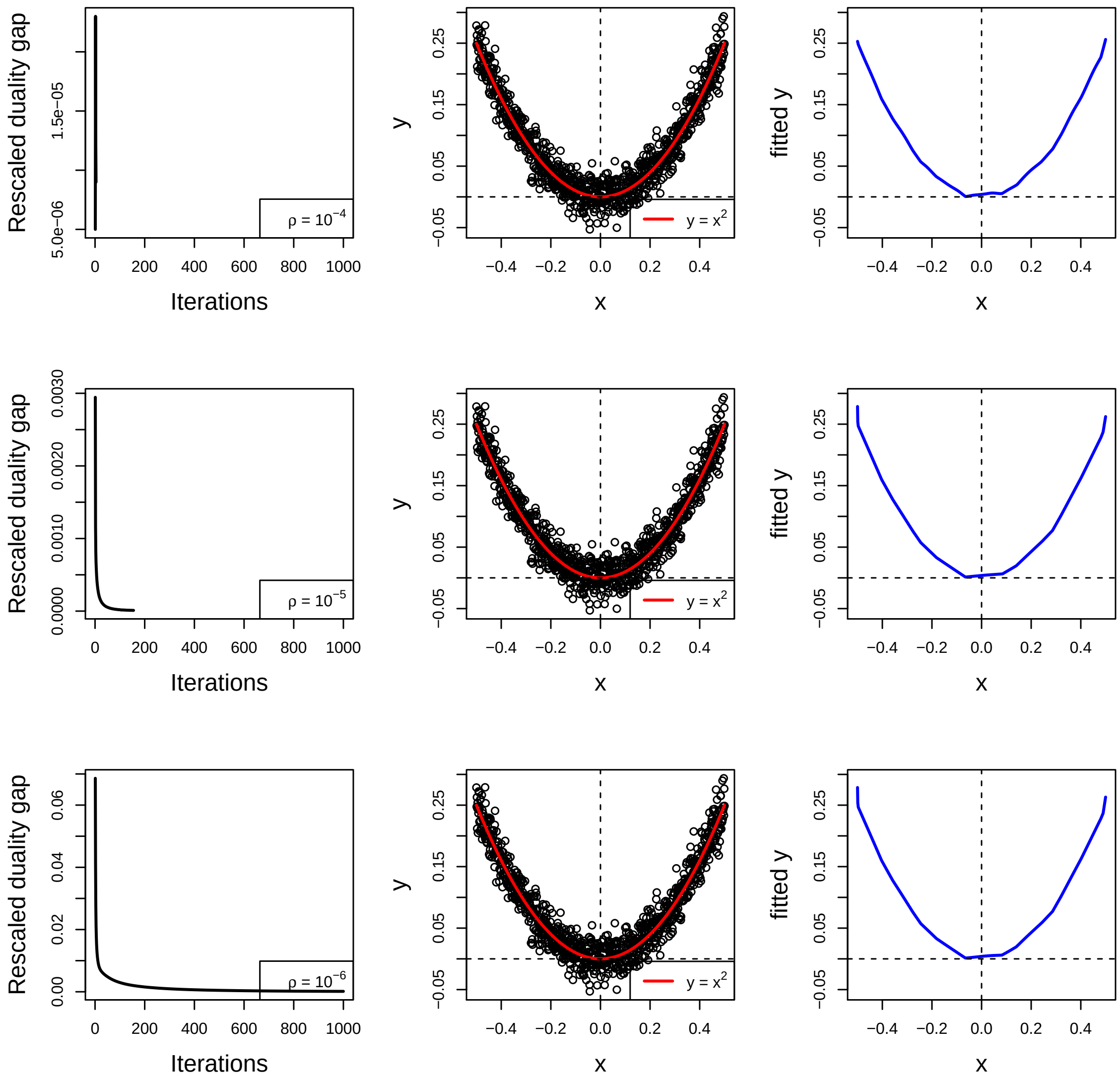

Figure 1: Time series of the (rescaled) duality gap, the sequence of simulated data, and the fitted curve, for $\rho = 10^{-4}, 10^{-5}, 10^{-6}$, respectively. Here the rescaling $= 10^{-6}$.

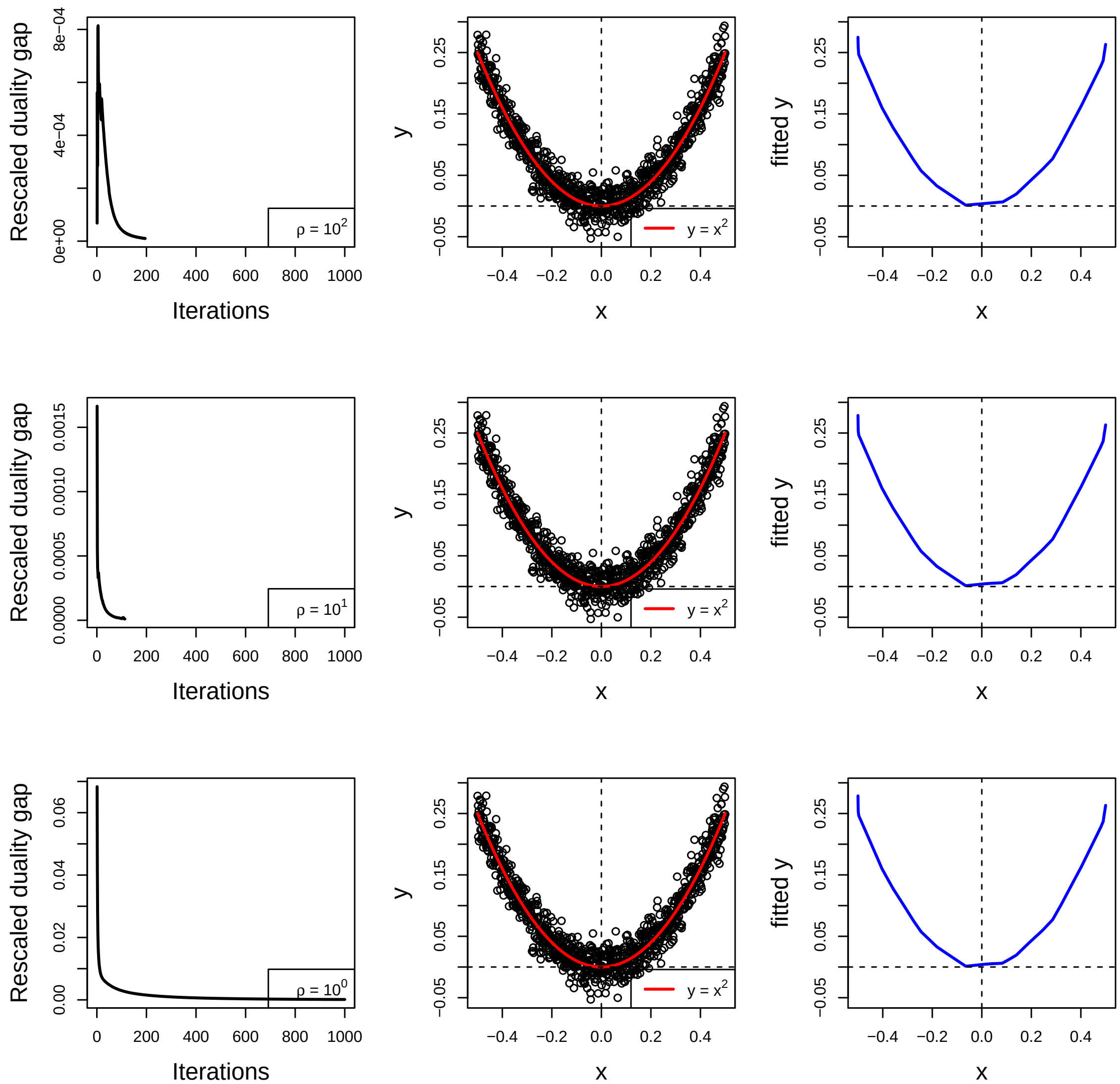

Figure 2: Time series of the (rescaled) duality gap, the sequence of simulated data, and the fitted curve, for $\rho = 100, 10, 1$, respectively. Here the rescaling $= 1$.

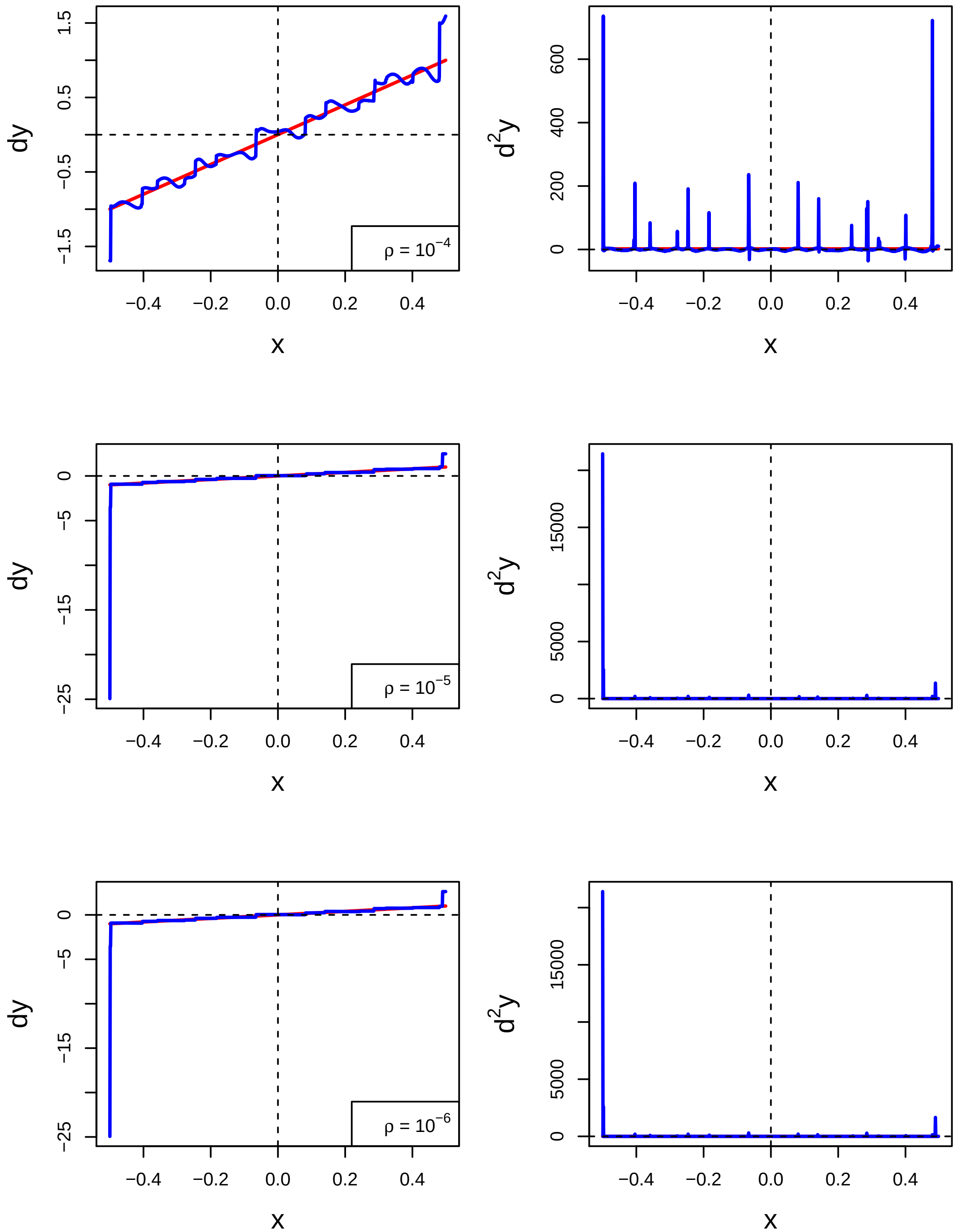

Figure 3: The first and second differences for the fitted data (blue) for $\rho = 10^{-4}, 10^{-5}, 10^{-6}$, compared with the corresponding reference curves for $y = x^2$ case (red).

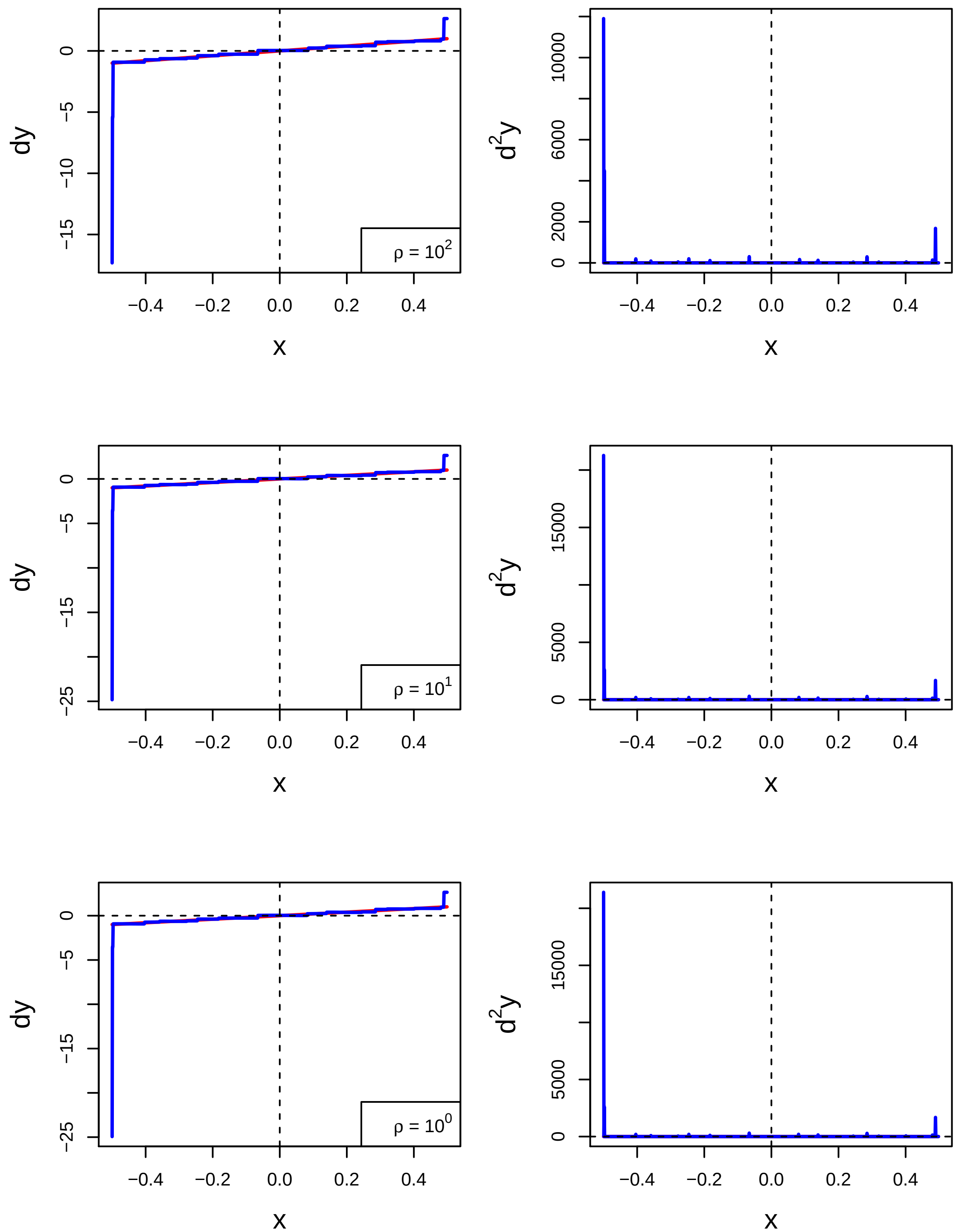

Figure 4: The first and second differences for the fitted data (blue) for $\rho = 100, 10, 1$, compared with the corresponding reference curves for $y = x^2$ case (red).

References

[1] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.