Comparison between quantile K-NN fused lasso and K-NN fused lasso

August 12, 2021

In [1], we proposed a new estimator, quantile K-NN fused lasso for robust estimation of piecewise constant or piecewise polynomial signals in a multivariate set-up.

Assume that we have n observations, $(x_1, y_1), ..., (x_n, y_n)$, of the pair of random variables (X, Y). The response variable Y is a real-valued vector and X is a multivariate covariate. Quantile K-NN fused lasso estimate is the solution to the following optimization problem

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\arg\min} \sum_{i=1}^{n} \rho_\tau(y_i - \theta_i) + \lambda \|\nabla_G \theta\|_1, \tag{1}$$

where τ is a given quantile, $\rho(t) = (\tau - 1\{t \leq 0\})t$ is the asymmetric absolute deviation function, $\lambda > 0$ is a tuning parameter, and ∇_G is an oriented incidence matrix of the K-NN graph G (See details for the construction of the K-NN graph and the defintion of incidence matrix in [1]).

Our proposed estimator is leveraged from the K-NN fused lasso estimator from [2], of which the corresponding optimization is

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \|\nabla_G \theta\|_1. \tag{2}$$

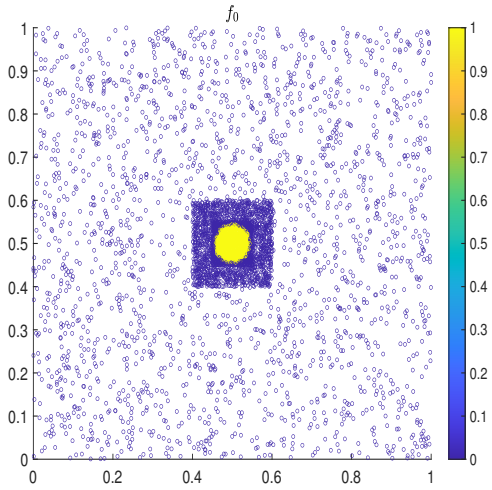
The main advantage of quantile K-NN fused lasso estimator compared to K-NN fused lasso estimator is that our estimator is by construction expected to be more robust to heavy tails and outliers than its precursor.

In this demo, we use a simulation experiment to present the comparison of performance between quantile KNN fused lasso and KNN fused lasso in a multivariate setup with heavy-tailed errors.

We inherit the data generative model of Scenario 2 from the Experiment section in [1]. We generate a 2d data sample with sample size $n = 10000$ and errors are selected to be t-distributed with degrees of freedom 3.

```
n = 10000;
[X1, X2] = genran(n);
f0 = double((X1 - 0.5).^2 + (X2 - 0.5).^2 <= 2/1000);
y = f0 + trnd(3,1,n);
X = [X1; X2];
```

The following plot depicts the true signals f_0 over a 2d grid.

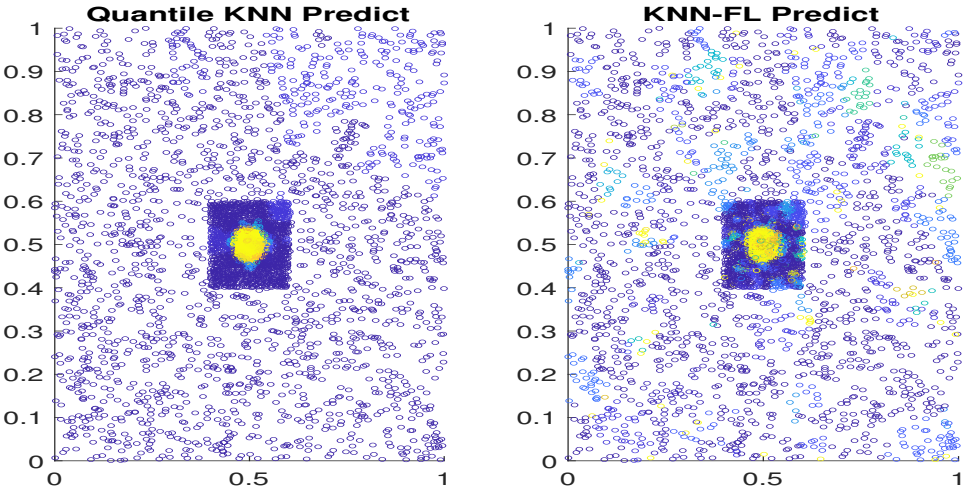


We compute the estimates from both methods, with a fixed tuning parameter $\lambda = 0.5$ and 1. We then calculate the mean squared errors for both estimates, respectively.

```
lambda = 0.5;
theta_qt = qt_knn_admm(X, y, 5, lambda, 0.5, 50);
mean((f0 - theta_qt).^2)
theta_fl = knnfl(X, y, 5, lambda);
mean((f0 - theta_fl).^2)

ans =

    0.0206


ans =

    0.8580


lambda = 1;
theta_qt = qt_knn_admm(X, y, 5, lambda, 0.5, 50);
mean((f0 - theta_qt).^2)
```

```
theta_fl = knnfl(X, y, 5, lambda);
mean((f0 - theta_fl).^2)

ans =

    0.0247


ans =

    0.4368
```

The results show that quantile K-NN fused estimates achieve smaller MSE than K-NN fused lasso estimates under both tuning parameter selections. These verify the robustness of our proposed estimator than its competitor in a heavy-tailed error setting.



If we take a careful look at the plots of comparison between the estimates from two estimators, we find that our method provides much closer estimates to the true function, while estimates from K-NN fused lasso is noisier.

We now repeat the above procedure with 100 Monte Carlo simulations and the tuning parameters are selected to minimize the loss for each sample. We compare the averaged mean square errors from both methods over these 100 Monte Carlo samples at the end. The results also validate our expectation on robustness of our estimator.

```
N = 100;
lambda = logspace(-2,2,50);
```

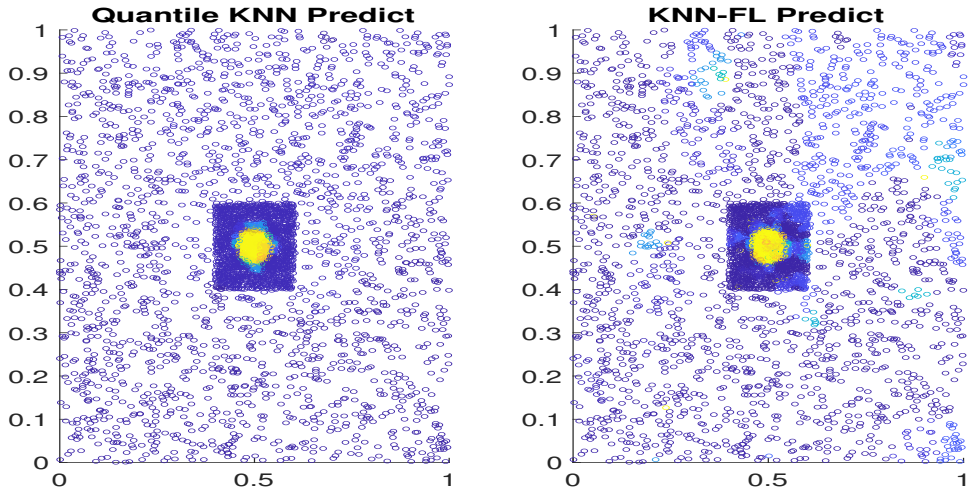


Figure 1: The comparison between estimates from quantile K-NN fused lasso and K-NN fused lasso. The top row is for estimates with $\lambda = 0.5$, and the bottom is for estimates with $\lambda = 1$.

```
mse_qt = zeros(1,N);
mse_fl = zeros(1,N);
for i = 1:N
    [X1, X2] = genran(n);
    f0 = double((X1 - 0.5).^2 + (X2 - 0.5).^2 <= 2/1000);
    y = f0 + trnd(3,1,n);
    X = [X1; X2];

    for j = 1:length(lambda)
        theta_qt = qt_knn_admm(X, y, 5, lambda(j), 0.5, 50);
        theta_fl = knnfl(X, y, 5, lambda(j));
        mse_1(j) = mean((f0 - theta_qt).^2);
        mse_2(j) = mean((f0 - theta_fl).^2);
    end
    mse_qt(i) = min(mse_1);
    mse_fl(i) = min(mse_2);
end
mean(mse_qt)
mean(mse_fl)

ans =

    0.0262
```

```
ans =

    0.0718
```

References

[1] Ye, S. and Padilla, O. H. M. (2021). Non-parametric quantile regression via the K-NN fused lasso. *Journal of Machine Learning Research.* **22**(111) 1–38.

[2] Padilla, O. H. M., Sharpnack, J., Chen, Y., and Witten, D. (2020). Adaptive non-parametric regression with the K-NN fused lasso. *Biometrika.* **107** (2) 293–310.