Course project

Yunzhang Zhu

1 Background

Conventional penalized spline utilize a ℓ_2 norm to penalize the coefficients of those splines basis. And this penalized spline optimization problem corresponds to choosing $(\boldsymbol{\beta}, \boldsymbol{u})$ to be the minimizer of the following optimization problem

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{u}}) = \underset{(\boldsymbol{\beta},\boldsymbol{u})}{\operatorname{argmin}} \Big\{ \big\| \boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u} \big\|^2 + \lambda \sum_{k=1}^{K} u_k^2 \Big\}, \tag{1}$$

where K is the number of knots in the spline model. The ℓ_2 penalty serves well when we only want to do prediction. In the case where prediction and knot selection are both desired, one may take advantage of some other recently-proposed penalty which can do both penalization and variable selection. An early landmark in this area was the lasso ℓ_1 penalty [5]. For our problem, the lasso corresponds to the penalty $\sum_{k=1}^{K} |u_k|$. And the estimator is defined as

$$(\hat{\boldsymbol{\beta}}^{\ell_1}, \hat{\boldsymbol{u}}^{\ell_1}) = \underset{(\boldsymbol{\beta},\boldsymbol{u})}{\operatorname{argmin}} \Big\{ \big\| \boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u} \big\|^2 + \lambda \sum_{k=1}^{K} |u_k| \Big\}, \tag{2}$$

In the optimization problem (2), this lasso penalty has the effect of forcing individual u_k's to be exactly zero, which is equivalent to doing variable selection (in our case knot selection). Equivalently, we will get a sparse estimate $\hat{\boldsymbol{u}}^{\ell_1}$ with nonzeros components corresponds to the selected knots by this method. Not only can lasso do automatic variable selection, as observed in [5] , it also often delivers much better predictive performance as compared to ℓ_2 penalty when the number of variables is comparable or even larger than the sample size. Penalized splines were used to fit models of the form $y_i = f(x_i) + \text{error}$, where x_i is scalar and $f(\cdot)$ is a smooth function of x. When the input variable $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ip})$ is a vector rather than a scalar, the model would be of the form $y_i = f(\boldsymbol{x}_i) + \text{error}$. Additive models where $f(\boldsymbol{x}_i) = \sum_{j=1}^{p} f_k(x_{ij})$ are the simplest version of this model that still allows a flexible shape for $f(\boldsymbol{x})$. Similarly, we can apply

the ℓ_1 penalty and optimize the following penalized log-likelihood

$$\sum_{j=1}^{p} \Big\| \boldsymbol{y} - \sum_{j=1}^{p} \big(\boldsymbol{X}_j \boldsymbol{\beta}_j + \boldsymbol{Z}_j \boldsymbol{u}_j \big) \Big\|^2 + \lambda \sum_{j=1}^{p} \sum_{k=1}^{K} |u_{jk}|. \tag{3}$$

2 Adaptive selection

One downside of either ℓ_2 or ℓ_1 penalized estimator is that they are both bias estimator of the underlying true function. This phenomenon is ubiquitous for convex penalties. To alleviate this issue, many adaptive procedure (adaptive lasso [7]) or non-convex alternatives (SCAD [3], MCP [6]) have been proposed recently. Here we introduce a ℓ_0 penalty proposed in [4] and its computational surrogate to do simultaneous selection and parameter estimation. It is shown in [4] that under some mild conditions on the design matrix and the true coefficients both variable selection and optimal parameter estimation can be achieved for linear models. For our spline model, we minimize the following objective

$$\big\| \boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u} \big\|^2 + \lambda \sum_{k=1}^{K} J_\tau\big(|u_k|\big), \tag{4}$$

where $J_\tau\big(|x|\big) = \min\Big(|x|/\tau, 1\Big)$ is a computational surrogate of ℓ_0 function $\mathbb{I}\big(|x| \neq 0\big)$ and $\tau > 0$ controls the similarity between these two function in that $\lim_{\tau \to 0} J_\tau\big(|x|\big) \to \mathbb{I}\big(|x| \neq 0\big)$.

By using this ℓ_0 penalty, the estimator recovers the true nonzero regression coefficients without incurring any additional bias. In other word, this results in a nearly unbiased estimator of the true underlying function curve and also it is expected to have much better predictive performance compared to ℓ_1 or ℓ_2 penalized splines which both deliver seriously bias estimates of the true function curve especially for the case when the total number of candidate knots is large compared to the sample size.

3 Computations

For ℓ_2-spline, the optimization is straight-forward which is equivalent to solve a linear equation. But for ℓ_1 and ℓ_0 penalty, special optimization routines should be employed to overcome the non smoothness of the objective functions in (2) and (4). For ℓ_1 penalized methods, there has been a huge literature recently on how to get a efficient algorithm for super-large problems. Here I use a relatively fast methods called

Alternating Directional Methods of Multipliers (ADMM) [2] which has been applied successfully be many non-smooth high dimension problems in statistics and machine learning. See [2] for an recent overview. For ℓ_0 penalty, we need an additional step before using ADMM because ℓ_0 penalty is not a convex function which results in a non-convex objective function. Our approach is to use Difference of Convex programming (DC programming) approach to convert the non-convex optimization problem into a sequence of convex relaxations, c.f. [1]. Then each relaxed convex problems are solved by ADMM. The detailed formulation of ADMM and DC programming is omitted here and can be found in [2] and [4].

4 Global mean surface temperature data

In this section, we compare results using ℓ_2, ℓ_1(Lasso) and ℓ_0 (TLP) methods to the global mean surface temperature data using quadratic and cubic truncated polynomial basis.

4.1 Design matrix specifications

In this section, we detail how the design matrix $\boldsymbol{X}$, $\boldsymbol{Z}$ in (1), (2) or (4) are generated from the raw data. Here we assume the raw data is $(\boldsymbol{x}, \boldsymbol{y})$ with $\boldsymbol{x}$ being the predictor and $\boldsymbol{y}$ the response vector. If we consider the general Mth order truncated polynomial basis for K knots, then the "raw random effect" before scaling is generated through

$$\boldsymbol{Z}_1^{\text{raw}} = \big(\boldsymbol{x} - \kappa_1\big)_+^M, \boldsymbol{Z}_2^{\text{raw}} = \big(\boldsymbol{x} - \kappa_2\big)_+^M, \cdots, \boldsymbol{Z}_K^{\text{raw}} = \big(\boldsymbol{x} - \kappa_K\big)_+^M,$$

where the knots $\kappa_1, \cdots, \kappa_K$ are equally spaced points on the interval $[\min(\boldsymbol{x}) + 5, \max(\boldsymbol{x}) - 5]$. Also the "raw fixed effects" $\boldsymbol{X}^{\text{raw}}$ is $\big(\boldsymbol{x}, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^M\big)$. To do penalized likelihood estimation, we must do some scaling to these raw fixed effects and random effects in order to penalize equally among random effects and also to ensure well-conditioned objective function for computational concerns. Here we scale all the effects: $\boldsymbol{X}^{\text{raw}}$ and $\boldsymbol{Z}^{\text{raw}}$ to get scaled design matrices: $\boldsymbol{X}$, $\boldsymbol{Z}$ where the columns of these two matrices has mean zero and ℓ_2 norm one. And also we center the response vector $\boldsymbol{y}$ to be $\boldsymbol{y}^c$. The final estimate $\big(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{u}}, \hat{\beta}_0\big)$ is obtained through first do the following optimization problem

$$\big(\hat{\boldsymbol{\beta}}^{\text{scaled}}(\lambda), \hat{\boldsymbol{u}}^{\text{scaled}}(\lambda)\big) \quad = \quad \underset{\boldsymbol{\beta} \in \mathbb{R}^M, \boldsymbol{u} \in \mathbb{R}^K}{\operatorname{argmin}} \Big(\big\| \boldsymbol{y}^c - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u} \big\|^2 + \lambda \boldsymbol{p}(\boldsymbol{u}) \Big),$$

where $\boldsymbol{p}(\boldsymbol{u})$ is the penalty function and λ is the penalty level controlling the complexity of the fit. After solving this, $\big(\hat{\boldsymbol{\beta}}^{\text{scaled}}(\lambda), \hat{\boldsymbol{u}}^{\text{scaled}}(\lambda)\big)$ are scaled back (divided by the normalizing constant of the corresponding column in the design matrix) to obtain $\big(\hat{\boldsymbol{\beta}}(\lambda), \hat{\boldsymbol{u}}(\lambda)\big)$. Finally, $\hat{\beta}_0(\lambda)$ is set to be the mean of $\boldsymbol{y} - \boldsymbol{X}^{\text{raw}}\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{Z}^{\text{raw}}\hat{\boldsymbol{u}}(\lambda)$.

4.2 Discussions

The solution paths are displayed in Figure 1-6. Also we display a comparison figure comparing Lasso solutions and TLP solutions on 9 different penalties in Figure 7. Here we discuss some key differences between these methods.

- (Knot selection) Both ℓ_1 and ℓ_0 can do automatic knot selection since they are designed to do simultaneous penalization and variable selection. We plot the selected knots with vertical lines in Figure 3-6.
- Compared to the non-convex ℓ_0 penalty, ℓ_1 and ℓ_2, being convex functions, will incur bias in the process of penalization. This is because ℓ_0 does knot pursuit of sparseness without incurring any bias the estimates. So it is expected that the solution path generated by approach using ℓ_0 penalty has bigger magnitude of fluctuation. So we plot estimators with same values of λ in Figure 7 to compare ℓ_0 and ℓ_1 methods. The fitted curve using ℓ_0 penalization is marked different from that of using ℓ_1 penalty.

Also notice that the fitted line using quadratic splines is similar to that of cubic splines. And also checked that the fitted curve is not sensitive to how many knots one uses as candidate knots. This is due to the selection feature of the ℓ_1 and ℓ_0 penalization methods.

References

[1] L.T.H. An and P.D. Tao. The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1):23–46, 2005.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Now Publishers, 2011.

[3] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[4] X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, To appear.

[5] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[6] C.H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[7] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

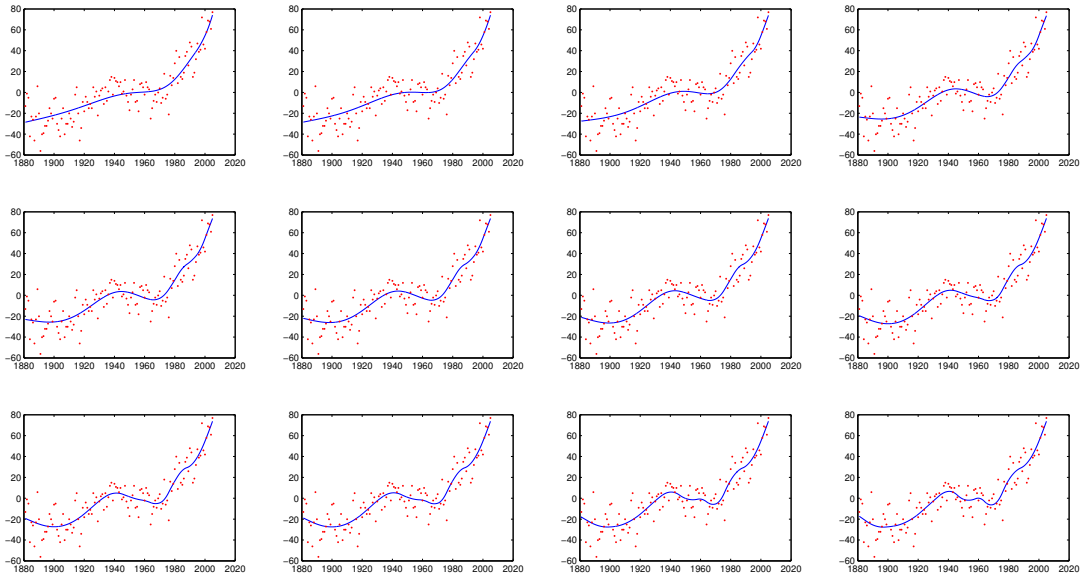

Figure 1: Solution path of ℓ_2(Ridge) using quadratic spline with 20 knots

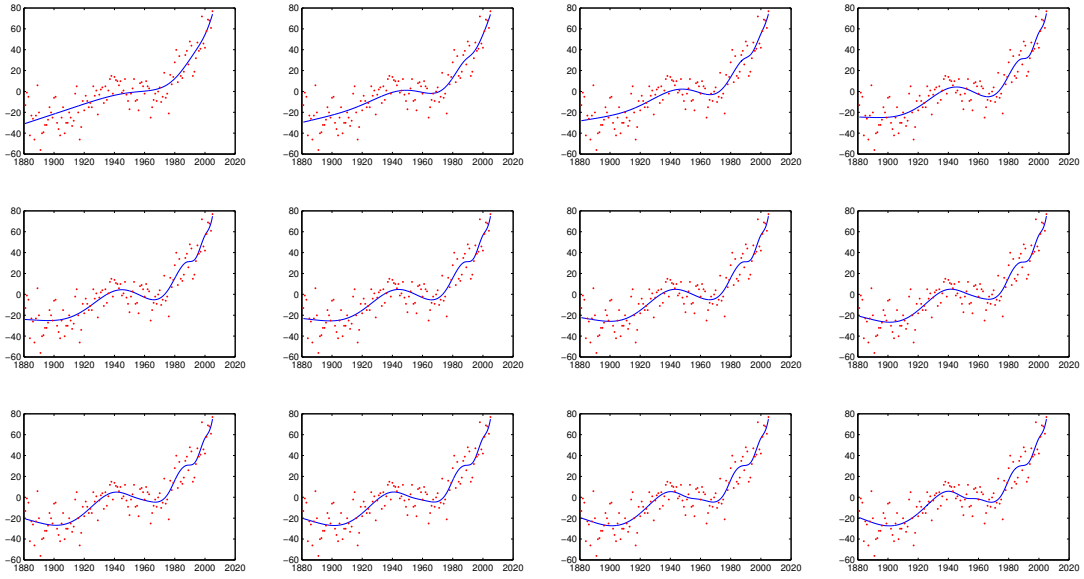

Figure 2: Solution path of ℓ_2(Ridge) using cubic spline with 20 knots

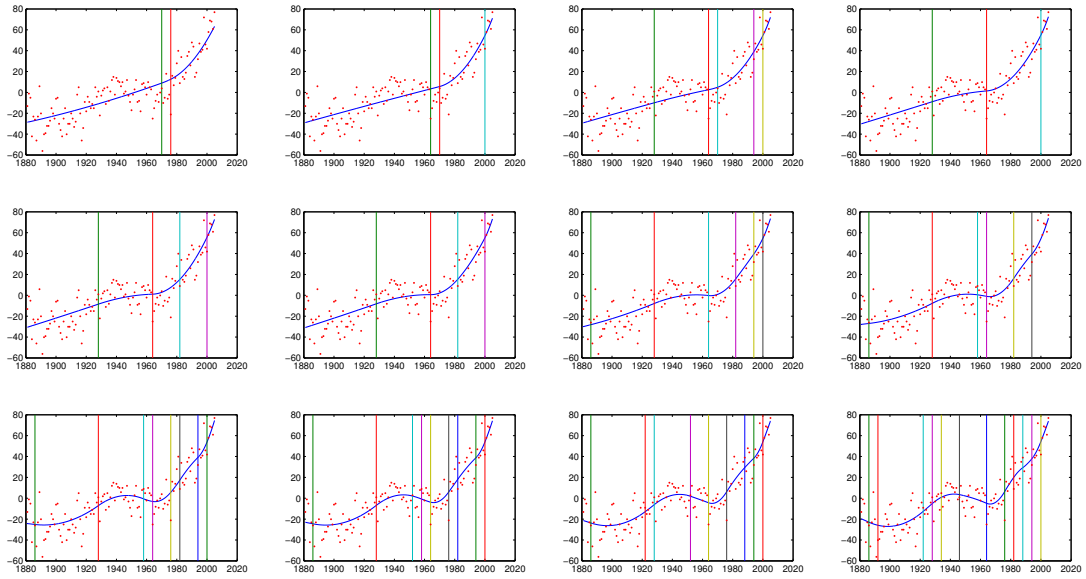

Figure 3: Solution path of ℓ_1(Lasso) using quadratic spline with 20 knots

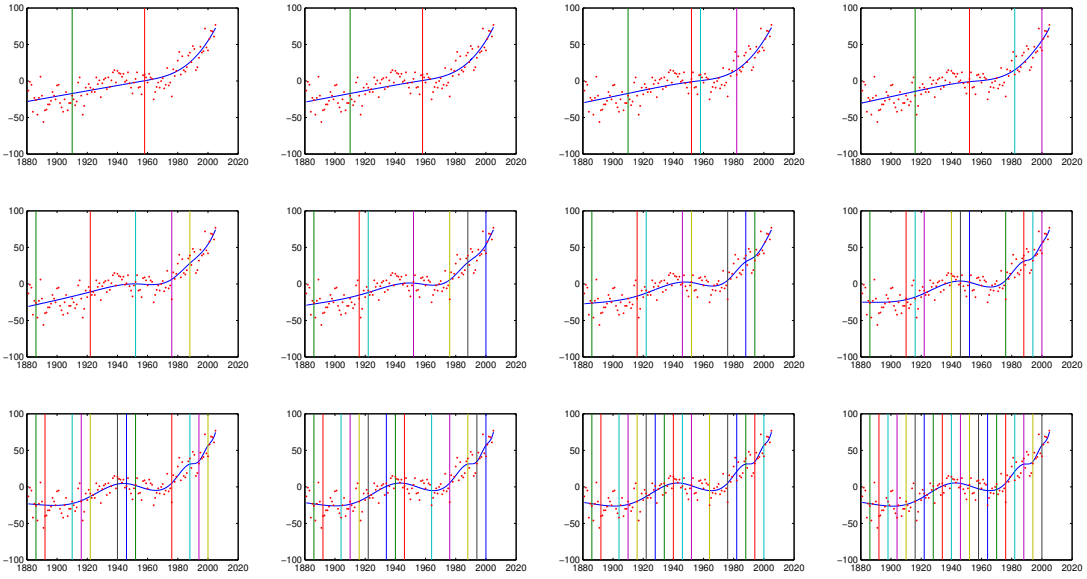

Figure 4: Solution path of ℓ_1(Lasso) using cubic spline with 20 knots

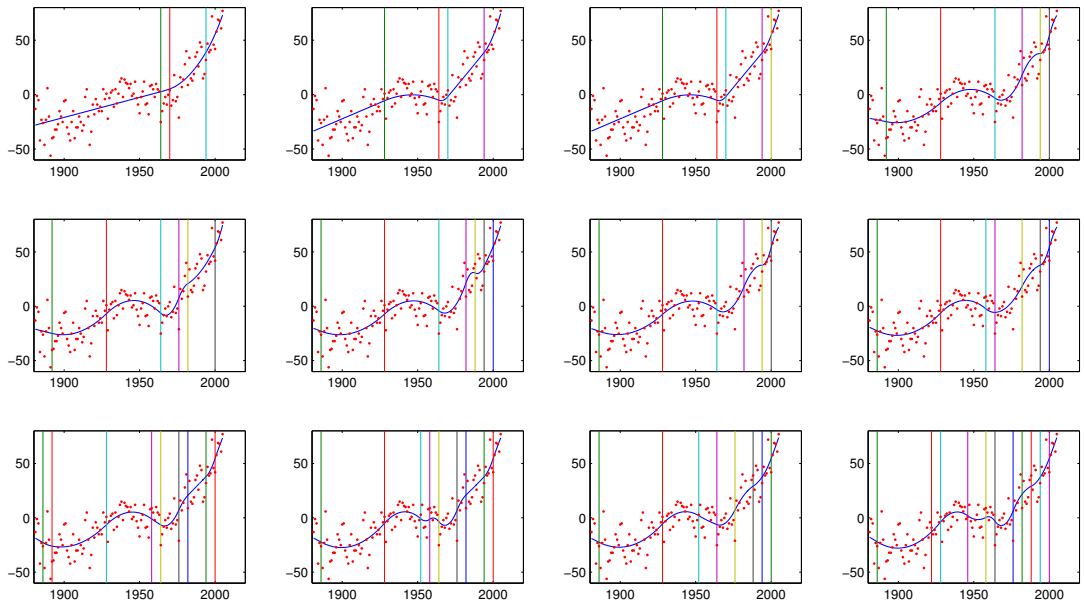

Figure 5: Solution path of ℓ_0(TLP) using quadratic spline with 20 knots

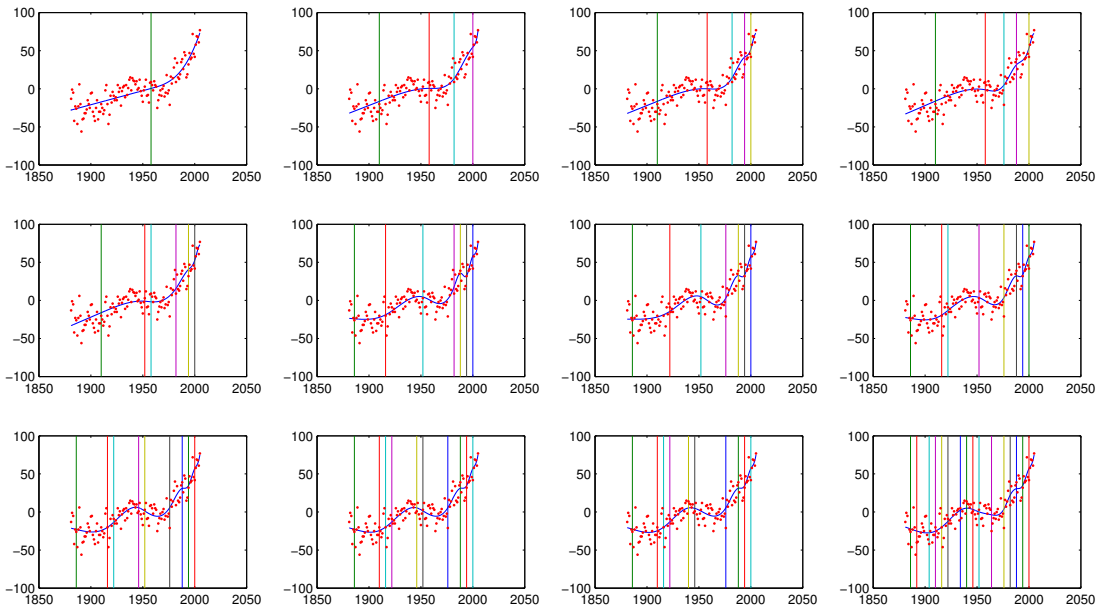

Figure 6: Solution path of ℓ_0(TLP) using cubic spline with 20 knots

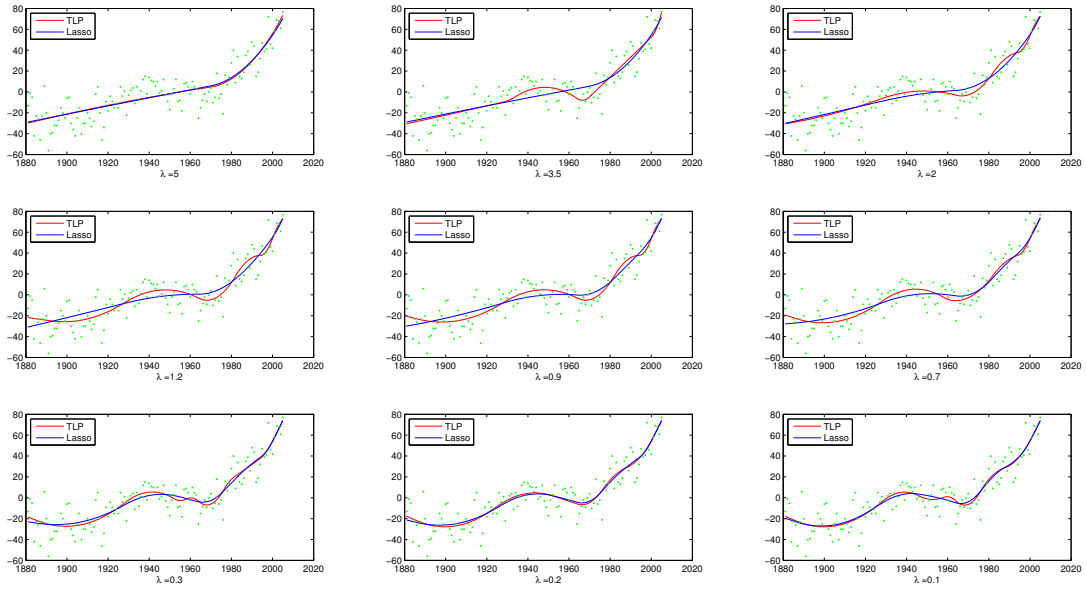

Figure 7: Comparison of ℓ_1(Lasso) and ℓ_0(TLP) using quadratic spline with 20 knots