Generalized Additive Model with Large Sample Size

1 Introduction

Although the generalized additive model has a wide range of applications, yet when the data size is too large, there may be cases where the computer cannot run the model at all. Therefore, some scholars proposed a modified version of generalized additive model for the big sample size, and this article tries to introduce this particular method in a language that is easy to understand. The "smoothing term" in the model is represented by a function of spline penalty, once the submatrix of the model matrix itself can be implemented on the computer, the method can get the factor of the model matrix by iteration.

If we assume n and p are the number of rows and columns of the model matrix, respectively, and M is the number of smoothing parameters. The difficulty of the generalized additive model is that the computer storage space required by the GAM method is usually $O(Mnp^2)$, so when the sample size is greatly increased, the requirement for the quality of the computer will also increase significantly.

In the following discussion, we consider the model function like this:

$$g\left\{E\left(y_i\right)\right\} = A_i\theta + \sum_j L_{ij} f_j$$

In which y_i is a univariate response variable of n observations and follows the exponential family distribution (or at least we assume the mean and variance are finite), $i = 1, 2, \ldots, n$; g is the smooth monotone connection function; A is a model

matrix of n rows; θ is a column vector composed by unknown parameters; L_{ij} is the linear function; f_j is an unknown univariate or multivariable smooth function, and the degree of smoothness is unknown . For each f_j, there is a function $J_j(f)$ that measures the deviation from smoothness. In GAM, L_{ij} is an evaluation function.

In particular, when f_j is a medium rank spline function, which acts as a penalty, the algorithm can be very **efficient** by fitting by iterative re-weighted penalty combining with the least square method, and the smooth parameter selection is performed by generalized cross-validation method (GCV), restricted maximum likelihood (RMEL) .

2 Gaussian identity connection

2.1 Fundamental Situation

First, consider that y_i obeys a normal distribution with independent and variance ϕ, and g is an identity function. F_j is generated by a linear basis (such as a B-spline base, or a thin plate regression spline base), and J_j is a quadratic form of the base coefficient. In this case, the model's expectations can be rewritten as:

$$E(y) = X\beta$$

X is the $n \times p$ order model matrix including A and the evaluation basis function, and β includes θ and all basis coefficients. We assume that $p < n$, because this method only makes sense in this case. The estimation of β can minimize the following equation:

$$y - X\beta^2 + \sum_j \lambda_j \beta^T S_j \beta$$

Here, S_j is a known coefficient matrix, such as $\mathrm{J_i}\left(f_j\right) = \beta^T S_j \beta$ (S_j is a $p \times p$-order matrix, but its non-zero matrix block is usually smaller than $p \times p$), and λ_j is a control f_j fit smoothing trade-off smoothing parameters (symbol notation here

is not very strict, because f_j may correspond to multiple smoothing parameters). As long as λ is given, the coefficient estimator $\hat{\beta}_\lambda$ can be obtained by minimizing equation $y - X\beta^2 + \sum_j \lambda_j \beta^T S_j \beta$. However, the estimation of λ is relatively complicated. One of the methods is to minimize the prediction error of the GCV, which is the same to minimize the following equations for smoothing parameters:

$$v_g(\lambda)\frac{n\left\|y - X\hat{\beta}_\lambda\right\|^2}{\{n - \mathrm{tr}\,(F_\lambda)\}^2}$$

Here, $tr(F_\lambda)$ is the effective degree of freedom of the model, $\mathrm{S}_\lambda = \sum_{\mathrm{j}} \lambda_{\mathrm{j}} \mathrm{S}_{\mathrm{j}}$, $\mathrm{F}_\lambda = \left(X^T X + S_\lambda\right)^{-1} \mathrm{X^T X}$. The Newton method is usually used to optimize the $_g$ for $log(\lambda)$, which is obtained for each of the obtained λ by direct minimization $y - X\beta^2 + \sum_j \lambda_j \beta^T S_j \beta$.

Now suppose that the model matrix is first decomposed into $X = QR$ using QR decomposition, where Q is the column orthogonal matrix of $n \times p$ and R is the upper triangular matrix of $p \times p$. At the same time, let $f = Q^T y$, $\|r\|^2 = \|y\|^2 - \|f\|^2$, then equation $y - X\beta^2 + \sum_j \lambda_j \beta^T S_j \beta$ becomes:

$$\|f - R\beta\|^2 + \|r\|^2 + \sum_{\mathrm{j}} \lambda_j \beta^T S_\beta \beta$$

Through ordinary calculation, we can get:

$$v_g(\lambda) = \frac{n\left\|f - R\hat{\beta}_\lambda\right\|^2 + \|r\|^2}{\{n - \mathrm{tr}(F_\lambda)\}^2}$$
$$F_\lambda = \left(R^T R + S_\lambda\right)^{-1} R^T R$$

The problem here is that as long as we have R, f, and $\|r\|^2$, we have all the parameters needed to fit, and the X at this point no longer works. So if we can get the values of these variables without building X, we can estimate the model without spending a lot of computer storage space.

2.2 Correlation error

Modeling auto-correlation residuals usually uses a simple $AR(p)$ correlation structure. First, the Gaussian constant connection model $E(y) = X\beta$ is reduced to $E(y) = X\beta + e$, where the covariance matrix of e is $\phi \sum$, and $\sum$ is the correlation matrix of auto-regressive $AR(p)$. Then the Choleski factor C of $\sum^{-1}$ is belt-like, and $\epsilon = Ce$ is independently and identically distributed as $N(0, \phi)$. In summary, if $\tilde{y} = Cy$, $\tilde{X} = CX$, then:

$$\tilde{y} = \tilde{X}\beta + \varepsilon$$

This formula is a variant of equation $E(y) = X\beta$, so it could be used to estimate β by the above method as well. The only change here is that if we use REML to estimate ρ, we must adjust the logarithm of REML so that it can be converted by C, but considering that C is a triangular matrix, we can easily get the logarithm of its determinant. In computation, a simple one-dimensional search can be conducted to obtain ρ, and for each ρ, the model needs to re-fit. Note that C is strip-shaped and can be obtained without solving $\sum^{-1}$, which makes $AR(p)$ more convenient to implement on a computer: $\tilde{y}$ and $\tilde{X}$ are weighted differences on y and X's adjacent rows, which obviously can save more computer storage and memory.

3 Fitting Generalized Additive Model

Broadly speaking, both the unknown function and their penalty spline can be represented as the simple Gaussian identity connection described above. The only differnece is that the model becomes an over-parameterized generalized linear model:

$$g\left\{E\left(y_i\right)\right\} = X_i\beta$$

At this point, we will use the penalty maximum likelihood estimate instead of

the penalty least square estimate. First define V such that $\operatorname{var}(y_i) = \phi V(\mu_i), \mu_i = E(y_i)$. The specific estimation steps are as follows:

First initialize $\hat{\mu}_{\mathrm{i}} = y_i + \xi_i, \hat{\eta}_i = g(\hat{\mu}_i)$, where ξ_i is a small amount (usually equals to 0), and this term is added to ensure the presence of $g(\hat{\mu}_i)$.

Then iterate the following steps until converge:

1. First, let $\mathrm{z_i} = g'(\hat{\mu}_i)(y_i - \hat{\mu}_i) + \hat{\eta}_i, w_i = V(\hat{\mu}_i)^{-1/2} g'(\hat{\mu}_i)^{-1}$;
2. Second, the diagonal matrix W is composed of w_i, and then minimize the equation $y - X\beta^2 + \sum_j \lambda_j \beta^T S_j \beta$ of the weighted form of β :
$$\|\mathrm{Wz} - \mathrm{WX}\beta\|^2 + \sum_{\mathrm{j}} \lambda_{\mathrm{i}} \beta^{\mathrm{T}} \mathrm{S_j} \beta$$
3. Then, the estimated value $\hat{\beta}$ is obtained, and $\hat{\eta} = \mathrm{X}\hat{\beta}$ and $\hat{\mu}_{\mathrm{i}} = \mathrm{g}^{-1}(\hat{\eta}_i)$ are updated.

For medium-sized data sets, it is relatively easy to iterate to convergence by **PIRLS** for each λ, where λ can be estimated using generalized GCV, Cp, Laplace approximation REML (Wood, 2008, 2011). But for large-size data sets, using these methods requires a large amount of memory storage to multiply the derivatives of the smooth criteria. In order to avoid spending the cost of such a huge storage space, we can use an earlier-designed method to replace it. In this way, we can use GCV, Cp, REML and other methods to select the smoothing parameters of the linear model at each step of the PIRLS algorithm. Using this method usually leads us to a convergent solution, and even if not all the time the convergence could be guaranteed, the disadvantages of convergence problems will still decrease as the sample size n increases.

3.1 Performance-oriented iteration for large data sets

Performance-oriented iterations for large data sets can be implemented using the QR-update method for the WX matrix at each step of the PIRLS algorithm. This

means that the calculation of the model matrix (submatrix) needs to be repeated at each step, but the storage space required for these calculations is $O(np)$ instead of the $O(np^2)$ of the QR-decomposition method, therefore, for those For smoothing, the computational cost is no longer important. The basic algorithm is as follows:

1. Initialization

   Let x_i be the covariate corresponding to the response variable y_i, $i = 1, ..., n$. The integers 1 to n are decomposed into M subsets of approximately the same dissimilar size $\gamma_1, ...\gamma_M$, $\bigcup_i y_i = \{1, \ldots, n\}$ and $\mathrm{y_j} \cap \mathrm{y_i} = \phi, \quad \forall \mathrm{i} \neq \mathrm{j}$. The above setting of M is to ensure that the calculation is within the storage space of the computer. Let $\bar{\eta}_\mathrm{i} = \mathrm{g}\left(y_i + \xi_i\right)$. Let the initial value of the PIRLS iteration be $q = 0$, $D = 0$ (in fact, we can make it to be any arbitrary constant). Perform any necessary initialization steps to set the base of the smooth item.

2. Iteration

   (1) Let $D_{old} = D$, R be a p-order zero matrix, and f is a zero vector, $D = 0$, $r = 0$.

   (2) Repeat the following steps (a) to (f), $k = 1, ...M$.

   (a) Let $f_0 = f$ and $R_0 = R$;

   (b) Construct a submatrix X_k of the model for the covariate set $\{x_i : i\gamma)k\}$

   (c) If $q > 0$, let $\hat{\eta} = \mathrm{X_k}\hat{\beta}$; otherwise, let $\hat{\eta} = \bar{\eta}_{\mathrm{yk}}$.

   (d) Let $\hat{\mu}_\mathrm{i} = \mathrm{g}^{-1}\left(\hat{\eta}_i\right)$, $z_i = g'\left(\hat{\mu}_i\right)\left(y_i - \hat{\mu}_i\right) + \hat{\eta}_i$, $w_i = V\left(\hat{\mu}_i\right)^{-1/2} g'\left(\hat{\mu}_i\right)^{-1}$, $\forall i \in y_k$, Denote z as a vector of z_i values and W be a diagonal matrix with diagonals corresponding to w_i values.

   (e) Let $r \leftarrow r + \|Wz\|^2$, calculate the outlier residuals of the current subdataset, and put the sum of squares into D.

   (f) Construct $\mathrm{QR} = \begin{pmatrix} \mathrm{R_0} \\ \mathrm{WX_k} \end{pmatrix}$ and $\mathrm{f} = \mathrm{Q^T}\begin{pmatrix} f_0 \\ W_z \end{pmatrix}$, delete D.

   (3) Let $\|r\|^2 = r - \|f\|^2$.

(4) If $q > 0$, check for convergence by comparing the current exception D with the previous exception D_{old}. If convergence is reached, the step is stopped (if the value of q has exceeded the preset threshold, this process should also be stopped).

(5) According to the introduction in the second part, $q > 0$ is estimated by optimizing $U_{r/g}$ or C_p. We can also get the value of $\hat{\beta}_\lambda$ through this step.

(6) Let $q \leftarrow q + 1$, the convergence results $\hat{\beta}_\lambda$ and λ are the estimated value of the coefficients and smoothing parameters. Further inference is based on Bayesian approximation

$$\beta \sim \mathrm{N}\left\{\hat{\beta}_\lambda, \left(R^T R + S_\lambda\right)^{-1} \phi\right\}$$

ϕ can be estimated through the optimization process of REML, or using the following equation:

$$\hat{\phi} = \frac{\|\mathrm{f} - \mathrm{R}\beta\|^2 + \|\mathrm{r}\|^2}{n - \mathrm{tr}\,(F_\lambda)}$$

3. Subsampling of initial values

Using the full data set to run the first few steps of the PIRLS algorithm above is obviously computationally wasteful. It is tantamount to a lot of work to fit a model that is wrong. Therefore, in general, we could always first randomly extract 5% to 10% of data from the original data set to form a **subsample**, and then the estimated β and λ in this subsample are used as initial values to do iteration and fit the entire data set. In practice, this procedure generally saves the processing time of first to second steps required by the PIRLS algorithm to fit the entire data set.

$$v_r^* = \frac{f - R\hat{\beta}_\lambda^2 + \hat{\beta}_\lambda^r S_\lambda \hat{\beta}_\lambda}{2\phi} + \frac{p - M_p}{2} \log(2\pi\phi)$$

3.2 Choosing the degree of smoothness

In every step of the PIRLS iteration: The assumptions required for these criteria are also the assumptions to the model, so no special reason is needed when applying GCV and C_p to the model. There are certain difficulties for REML (or ML), which is mainly because the data z_i may be quite different from the normal assumption v_r. But for large data sets, in which $n >> p$ is satisified, according to the central limit theorem, we can derived that $f = Q^T z$ (where z contains all the data) will tend to follow the $N(R\beta, I\phi)$ distribution. The REML score based on the f's density function is:

$$v_r^* = \frac{f - R\hat{\beta}_\lambda^2 + \hat{\beta}_\lambda^T S_\lambda \hat{\beta}_\lambda}{2\phi} + \frac{p - M_p}{2}\log(2\pi\phi) + \frac{\log\left|R^T R + S_\lambda\right| - \log|S_\lambda|_+}{2}$$

$|S_\lambda|_+$ is the product of all positive eigenvalues of S_λ, and S_λ has M_p numbers of eigenvalues. Thus v_r^* has the same form as v_r, but its $\|r\|^2 = 0$ and n is set to p. For an arbitrary ϕ, it is obvious that v_r and v_r^* are obtained by minimizing the same λ value. However, if ϕ is unknown, it should be estimated through some method. Obviously, optimizing v_r^* is not a good choice because of the missing term $\|r\|^2$ carries information on ϕ. At this point, we can use the following formula to estimate:

$$\hat{\phi} = \frac{\left\|f - R\hat{\beta}_\lambda\right\|^2 + \|r\|^2 + \hat{\beta}_\lambda^T S_\lambda \hat{\beta}_\lambda}{n - M_p}$$

This estimate can be thought of as an estimate similar to RMEL, or as a simple moment estimator (usually the denominator of the estimator is the sum of the Pearson statistic and the smoothed penalty). It is easy to see that $\hat{\phi}$ is the minimum value of v_r , and v_r and v_r^* are obtained by minimizing the same λ. Therefore, when $\hat{\phi}$ is used as the estimator of ϕ, we can obtain $\hat{\lambda}$ and $\hat{\phi}$ by minimizing v_r (where R, f, $\|r\|^2$ could be obtained by the PIRLS algorithm in the third part).

Reissue and Ogden (2009) have already shown that v_r is unlikely to have multiple local minima (or to say it's closely related to C_p) comparing to v_g. In

the performance-oriented iteration context, this means that REML facilitates the convergence of iterations because it reduces the loop of the iteration among multiple optimal values.

3.3 The assumption of p < n

Using GAM to process large data sets, the computational feasibility depends on the ability to use p-level smoothing to make p much smaller than n. But a problem cannot be neglected is that whether it is reasonable to assume the growth rate of p is much slower than the growth rate of n. Obviously, if we use the smoothing term as a random part to remove the residual auto-correlation, or to smooth time when new data is added to extend the time axis, the assumption that "p grows much faster than n" is no longer reasonable.

Otherwise, all the theoretical basis will be changed to that the smoothing base only grow slowly as the sample size grows: For example, consider a case that a cubic regression spline with balanced interval distance. The average of the squared deviation of the cubic splines is $O(h^8) = O(k^{-8})$, where h is the interval spacing and k is the number of nodes.

From the basic regression theory, the mean value of the spline variance is $O(k/n)$, where n is the sample size. In order to avoid the interaction between the skewness and the variance, and considering the sub-optimal mean square error when $n \to \text{inf}$, we need to select the k which has the same order as the square of the skewness and the variance, such as $k \propto n^{1/9}$. Considering that in any finite sample set, we choose to reduce the penalty for mean square error (as opposed to pure regression). Therefore, the above ratio can also be used in the regression of the penalty, that is, $k \propto n^{1/9}$ (of course, other ratios are also appliable). In fact, Gu and Kim (2002) have shown that the basic dimension with penalty should be an $n^{2/9}$ scale, that is, comparing with the the data set with 1000 observations, the

number of coefficients needed in the sample of 1 million data points is only about 5 times larger.

4 Conclusion

The advantage of this method is that it's a relatively straightforward extension of existing methods, but it's improved in terms of processing in larger sample size of the data set.

Nowadays, more and more large size of data sets are produced in the fields of remote sensing, genetic technology, finance and information technology, etc. For some researches, dealing with these data sets requires a new analytical approach; yet there are also some problems which using the existing methods are still the key to achieve effectiveness. The method proposed in this paper is based on the existing solutions, and hope it could be widely used.

References

[1] 许亦频, 倪苹. 适用于大数据集的广义可加模型 [J]. 统计研究, 2016, 33(04) : 104-112.

[2] Hongmei Lin,Heng Lian,Hua Liang. Rank reduction for high-dimensional generalized additive models[J]. *Journal of Multivariate Analysis*, 2019, 173.

[3] Giampiero Marra,Simon N. Wood. Practical variable selection for generalized additive models[J]. *Computational Statistics and Data Analysis*, 2011, 55(7).