8 Generalized Additive models

A generalized linear model (GLM) is a flexible extension of the ordinary linear regression model and can represent a variety of distinct regression models. The configuration of a Generalized linear model is the following :

- A linear predictor $\eta(x) = \beta_0 + X\beta$. i.e. covariates are defined as a linear model, upon which is build the GLM.
- A random component is defined as following some distribution from the exponential distribution family [69]. For example, the ordinary linear regression model defines the random component $\epsilon \sim N(0, \sigma^2)$, and as a result $y|x \sim N(X\beta, \sigma^2)$
- A link function that links between the random $E[Y|X]$ and the covariates. The link function is a bijection that transforms $E[Y|X]$ to the linear predictor $\eta(x)$. For instance in a linear regression $\mu(x) = \beta_0 + X\beta$, the link function is the identity function.

In summary, the Generalized Linear Model is a way to express different regression models based on some linear predictor assuming some random component and given some link function

I will focus in this paper, on a more generalized version of the Generalized linear models, notably the Generalized additive model (GAMs) . Generalized additive models are defined as conditional expectation regressions linked to the sum of arbitrary smooth functions (one for each variable) by a link function. Formally, it is defined as:

$$g(\mathrm{E}(Y|X)) = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$$

where $g(.)$ is the link function; $f_j(.)$ are unspecified smooth functions , and m is the number of factors. $f_j(.)$ are arbitrary functions that can change from one predictor to another, provided that it is smooth [70] ; and what makes this model special is that it is flexible: $f_j(.)$ can , for example, take the form of some fully-parametric functions (polynomial regressions, linear regressions etc...) , expansions of basis functions (Natural K-splines, Sigmoid basis expansions etc...) , or fully non-parametric smoothing functions (Nadaraya-Watson Kernel regressions, K-NN etc...) ... the list is expansive. Generalized additive models also assume $\mathbf{E}[Y] = \beta_0$ and $\mathbf{E}[f_j(X_j)] = 0$ in order to make the problem identifiable. Essentially, if we do not assume the following, we end up with "Concurvity" - The generalization of collinearity in an additive model framework- that is, there are infinitely many parameters that gives

[69] Do not confound with exponential distribution. An exponential distribution family is a set of probability distribution function expressed as $f_X(x \mid \theta) = h(x)\exp[\eta(\theta) \cdot T(x) - A(\theta)]$

[70] The smoothness of the functions refers to their continuity in their first and second order derivatives. Hence $f_j()$ can be represented by any C^2 function

us the same regression function. $g(.)$ the link function, is the same as the one defined for Generalized linear models; however, instead of linking to a linear predictor model it links to an additive model. In addition; because, under GAMs, $\eta(.)$ is no more a linear function, the estimation process changes[71]: GAMs use instead a "Back-fitting" algorithm to fit $f_j()$s . Hastie et al. define the back fitting algorithm as such: First Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i, \hat{f}_j \equiv 0, \forall i, j$. Then for: $j = 1, 2, \ldots, p, \ldots, 1, 2, \ldots, p, \ldots,$

$$\hat{f}_j \leftarrow \mathcal{S}_j \left[\left\{ y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k \left(x_{ik} \right) \right\}_1^N \right],$$
$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N} \sum_{i=1}^N \hat{f}_j \left(x_{ij} \right).$$

And iterate until convergence criterion is attained [72] (See Backfitting Algorithm section in Appendix for an explanation of the underlying logic of this algorithm) For S_j some smoothing operator which we choose according to the $f_j(.)$. In other words, the backfitting algorithm sequentially fits each factor while keeping others fixed. Updating a function involves applying the fitting method to a partial residual. For instance, if $f_1(.)$ and $f_2(.)$ are known, we can fit $f_3(.)$ by treating the partial residual as a response in some smooth regression on x_3. (Refer to appendix for detailed explanation)

We can clearly see that GLMs are special cases of GAMs, for $f_j(.)$ being linear in x, they only differ in their estimation, in their speed and in their biasedness [73]. I introduce both Generalized Additive Models and Generalized linear models, because in this paper I present a penalized form of GAM which, if penalized enough, may reduce to a GLM model. Moreover, I chose Generalized Additive models to represent non-linear models because it is the best suited to my problem; it is a good compromise between fully parametric non linear models and unstructured non-parametric smoothing methods .

In fact, on the one hand, even though fully-parametric models have been traditionally used in factor modeling and despite the fact that their estimation error converges quickly as the number of data increase[74]. The main problem is that it will always result in an approximation error if the underlying conditional expectation is not exactly matching the model.

On the other hand, unstructured non-parametric smoothing methods (Those are regressions that impose no assumptions on the the shape of the regression function) [75] can asymptotically capture any true underlying conditional expectation function as their fitting approach is data-dependent, free

[71] We cannot use linear regression - as we did for GLM - for non-parametric $f_j(.)$s - It does not make sense

[72] One can either specify a tolerance level, or some fixed maximum number of iterations

[73] GLM converge faster while GAM are less bias

[74] For instance, mean squared error 's convergence of linear models is $MSE_{\text{linear}} = \underbrace{\sigma}_{\text{intrinsic error}} + \underbrace{a_{\text{linear}}}_{approximateerror} + \underbrace{O\left(n^{-1}\right)}_{estimationerror}$ (Shalizi,2021), this is derivable by using the Law of iterated expectations from MSE, essentially, there will always be some approximation error, even if we infinitely increase the sample size. These MSE convergence property is generalizable to any parametric model (Shalizi, 2021[23])

[75] They are generally defined as: $\mathbb{E}(Y|X) = \sum_{i=1}^n y_i w\left(x, x_i, h\right)$ with $w(.)$ some fully non-parametric function of some tuning parameter h. Kernel regressions or K-NN are notable instances of unstructured non-parametric models

of any model restrictions. However , the main issue with these methods is that their estimation error is dependent on p (the independent variables), and fitting these models may fall under the curse of high dimensionality: Intuitively, for some sample of observations, fitting the model just by looking at the data becomes increasingly difficult as the number of dimensions increase. Wassermann (2006)[30], derives the Mean squared error asymptotics of unstructured non parametric methods as :$MSE_{\text{nonpara}} - \underbrace{\sigma^2}_{intrinsic error} = \underbrace{O\left(n^{-4/(p+4)}\right)}_{rate of convergence of estimation error}$, as having no approximation error but with an estimation-error rate of convergence (to zero) dependent on the number of features i.e. this is a formalized representation of the curse for dimensionality for unstructured non parametric methods.

Generalized additive models emerge as a perfect compromise between fitting well the data and not falling in the high dimensionality curse trap: In fact, it is a structured non-parametric method that uses non parametric smoothing functions $f_j(X_j)$ on each of the predictors ; the regression is no more dependent on p parameters. Rather; what we have with GAMs is P non-parametric functions each dependent on a single parameter. We thus fit a non parametric function - that minimizes its specific approximation error - without suffering from a large estimation error due to high dimensionality - as GAMs smooths P times on 1 dimension. For instance, for a simple GAM on p features, with $f_1(X_1), ..., f_p(X_p)$ all being smoothing splines, Shalizi (2021) derives $MSE_{\text{additive}} - \sigma^2 = a_{\text{additive}} + O\left(n^{-4/5}\right)$[76]: i.e. there still is some approximation error $a_{additive}$ as the approximate error combining all dimensions together has not been tackled by GAM, however, approximation error is better than what we get for a linear model $a_{additive} \leq a_{linear}$ [77] and we do not fall into any dimensionality problem - the estimation error convergence solely depends on N not P (Note:$O(n^{-4/5})$ in the formula) Another, yet weaker, advantage of choosing GAMs is that they are interpretable models: By posing the problem as an additive one, one clearly see the parts constituting the overall model, and thus conjecture the dynamics of the model.

For these reasons I chose GAMs to modelize non-linearly the returns with respect to high dimensional factors. and choose to model the arbitrary functions with respect to second order splines with k knots (i.e $k+1$ intervals). The number of knots is a hyper parameter; which I tune using cross validation. Formally: Each $f_j(X_j)$ is modeled as $\theta_j p(x)$ with $p(x)$ a second order spline defined as $\beta_0 + \beta_1 x_j + \beta_2 x_j^2 + \beta_3 (x_j - k)_+^2$ with $(x_j - k)_+^2$ a truncated power basis defined as $(x - k_i)_+^2 = \begin{cases} (x-k_1)^2 & \text{if } x \in [k_1, k_2) \\ \cdots & \\ (x-k_K)^2 & \text{if } x > k_K \\ 0 & \text{otherwise} \end{cases}$ Second order splines are chosen in this framework simply

[76]This derivation requires Taylor approximating the MSE and using Oracle assumptions
[77]Since Additive models $\subset$ Linear models - this has been discussed in the statistical learning theory Chapter

because they represent standard flexible C^2 functions [78] .They are fitted by least squares regression, hence the smoothing operator S_j in the backfitting algorithm is the squared loss. Concretely, the Generalized additive model will look like this:

$$g(\mathrm{E}(Y|X)) = \theta_0 + \sum_{j=1}^{P} p\left(X_j\right)' \theta_j$$

with $g(.) = \mathbb{I}$ the Identity function , P the total number of factors, and I assume that $p(z_1), ..., p(z_j)$ are all second order splines with K knots (Notice that there are no feature index to the spline functions as all splines in this model are all the same for the different features; features differ in their coefficient θ_j). While generalized additive models mitigates the curse of high dimensionality, the model can become highly parameterized, particularly with an increased number of knots, as the number of parameters, k.(Order of the spline + 1), scales linearly, increasing by a constant factor of 3 for each additional knot. Given the increased parametrization, which complicates model interpretation, and considering my earlier discussion on regularization's role in enhancing generalization, I employ the Grouped Lasso Regularization method (Yuan, Lin 2005). Grouped lasso is, like Ridge and Lasso, a Tikhonov Regularization, on an l_2 normed (Non-squared) penalty. It thus follows the same rationale discussed in the regularization section. Grouped lasso has however two distinctive features: First, its penalty norm is l_2 normed thereby inducing sparsity; this is clearly shown in the Lasso Chapter [79]. Secondly, and importantly, it penalizes coefficients in batches rather than individually. In this paper, I utilize this regularization method to nullify all the k-splines associated with each feature if needed. Specifically, groups are formed by the k coefficients (β_k) of each basis function in each predictor. And formally, the smoothing operator in the back fitting algorithm is now defined as such: $\min_\beta \left(\left\| \mathbf{y} - \theta_0 - \sum_{j=1}^{P} p\left(X_j\right)' \theta_j \right\|_2^2 + \lambda \sum_{j=1}^{P} \sqrt{N^j} \left\| \beta_j \right\| \right)$, With $\beta_j = (\beta_1, ..., \beta_K)$ and N^j the number of elements the coefficients of the basis function of the K spline associated to each predictor. [80]

[78] It is common to choose cubic spline (As far as I know) as they are able to represent complex curvatures smoothly. Second order splines, are able to represent non linear functions too, however, they can be less efficient in representing complex curvatures (e.g. sharp wiggly behaviours) , they are nonetheless used in this paper for computational purposes. In fact, since I perform GAM on 920 features; cubic splines gives me $3680k$ parameters per smoothing function while the second order result in $2760k$.

[79] Note on terminology: Even though ridge regression is commonly referred to as the "l_2" normed regularization; Ridge is effectively a "Squared l_2" normed regularization. The squared L_2 ridge penalty does not result in sparse regularization whereas the l_2, used in Group Lasso does induce sparsity.

[80] Notice that; like for the ridge and lasso regularization methods, the intercept is not penalized. In addition, standardization too is required in Grouped Lasso; for the same reasons discussed in the $l1$ and $l2$ regularization chapters